Informing Determiner and Preposition Error Correction with Word Clusters

Adriane Boyd **Marion Zepf** **Detmar Meurers**
Seminar für Sprachwissenschaft
Universität Tübingen
{adriane,mzepf,dm}@sfs.uni-tuebingen.de

Abstract

We extend our n-gram-based data-driven prediction approach from the Helping Our Own (HOO) 2011 Shared Task (Boyd and Meurers, 2011) to identify determiner and preposition errors in non-native English essays from the Cambridge Learner Corpus FCE Dataset (Yannakoudakis et al., 2011) as part of the HOO 2012 Shared Task. Our system focuses on three error categories: missing determiner, incorrect determiner, and incorrect preposition. Approximately two-thirds of the errors annotated in HOO 2012 training and test data fall into these three categories. To improve our approach, we developed a missing determiner detector and incorporated word clustering (Brown et al., 1992) into the n-gram prediction approach.

1 Introduction

We extend our n-gram-based prediction approach (Boyd and Meurers, 2011) from the HOO 2011 Shared Task (Dale and Kilgarriff, 2011) for the HOO 2012 Shared Task. This approach is an extension of the preposition prediction approach presented in Elghafari, Meurers and Wunsch (2010), which uses a surface-based approach to predict prepositions in English using frequency information from web searches to choose the most likely preposition in a given context. For each preposition in the text, the prediction algorithm considers up to three words of context on each side of the preposition, building a 7-gram with a preposition slot in the middle:

```
rather a question __ the scales falling
```

For each prediction task, a *cohort* of queries is constructed with each of the candidate prepositions in the slot to be predicted:

```
1. rather a question of the scales falling
2. rather a question to the scales falling
3. rather a question in the scales falling
...
9. rather a question on the scales falling
```

In Elghafari, Meurers and Wunsch (2010), the queries are submitted to the Yahoo search engine and in Boyd and Meurers (2011), the search engine is replaced with the ACL Anthology Reference Corpus (ARC, Bird et al., 2008), which contains texts of the same genre as the HOO 2011 data. If no hits are found for any of the 7-gram queries, shorter overlapping n-grams are used to approximate the 7-gram query. For instance, a 7-gram may be approximated by two overlapping 6-grams:

```
[rather a question of the scales falling]
                    ⇓
[rather a question of the scales]
       [a question of the scales falling]
```

If there are still no hits, the overlap backoff will continue reducing the n-gram length until it reaches 3-grams with one word of context on each side of the candidate correction. If no hits are found at the 3-gram level, the Boyd and Meurers (2011) approach predicts the original token, effectively making no modifications to the original text. The approach from Elghafari, Meurers and Wunsch (2010), addressing a prediction task rather than a correction task (i.e., the original token is masked), predicted the most frequent preposition *of* if no hits were found.

The 7th Workshop on the Innovative Use of NLP for Building Educational Applications, pages 208–215,
Montréal, Canada, June 3-8, 2012.

Elghafari, Meurers and Wunsch (2010) showed this surface-based approach to be competitive with published state-of-the-art machine learning approaches using complex feature sets (Gamon et al., 2008; De Felice, 2008; Tetreault and Chodorow, 2008; Bergsma et al., 2009). For a set of nine frequent prepositions (*of, to, in, for, on, with, at, by, from*), they accurately predicted 76.5% on native data from section J of the British National Corpus. For these nine prepositions, De Felice (2008) identified a baseline of 27% for the task of choosing a preposition in a slot (choose *of*) and her system achieved 70.1% accuracy. Humans performing the same task agree 89% of the time (De Felice, 2008).

For the academic texts in the HOO 2011 Shared Task, Boyd and Meurers (2011) detected 67% of determiner and preposition substitution errors (equivalent to detection recall in the current task) and provided the appropriate correction for approximately half of the detected cases. We achieved a detection F-score of approximately 80% and a correction F-score of 44% for the four function word prediction tasks we considered (determiners, prepositions, conjunctions, and quantifiers).

2 Our Approach

For the 2012 shared task corpus, we do not have the advantage of access to a genre-specific reference corpus such as the ARC used for the first challenge, so we instead use the Google Web 1T 5-gram Corpus (Web1T5, Brants and Franz, 2006), which contains 1-gram to 5-gram counts for a web corpus with approximately 1 trillion tokens and 95 billion sentences. Compared to our earlier approach, using the Web1T5 corpus reduces the size of available context by going from 7-grams to 5-grams, but we are intentionally keeping the corpus resources and algorithm simple. We are particularly interested in exploring the space between surface forms and abstractions by incorporating information from word clustering, an issue which is independent from the choice of a more sophisticated learning algorithm.

Rozovskaya and Roth (2011) compared a range of learning algorithms for the task of correcting errors made by non-native writers, including an averaged perceptron algorithm (Rizzolo and Roth, 2007) and an n-gram count-based approach (Bergsma et al., 2009), which is similar to our approach. They found that the count-based approach performs nearly as well as the averaged perceptron approach when trained with ten times as much data. Without access to a large multi-genre corpus even a tenth the size of the Web1T5 corpus, we chose to use Web1T5. Our longest queries thus are 5-grams with at least one word of context on each side of the candidate function word and the shortest are 3-grams with one word of context on each side. A large multi-genre corpus would improve the results by supporting access to longer n-grams, and it would also make deeper linguistic analysis such as part-of-speech tagging feasible.

Table 1 shows the sets of determiners and prepositions for each of the three categories addressed by our system: missing determiner (MD), incorrect determiner (RD), and incorrect preposition (RT). The function word lists are compiled from all single-word corrections of these types in the training data. The counts show the frequency of the error types in the test data, along with the total frequency of function word candidates.

The following sections describe the main extensions to our system for the 2012 shared task: a simple correction probability model, a missing determiner detector, and the addition of hierarchical word clustering to the prediction approach.

2.1 Correction Probability Model

To adapt the system for the CLC FCE learner data, we added a simple correction probability model to the n-gram predictor that multiplies the counts for each n-gram by the probability of a particular replacement in the training data. The model includes both correct and incorrect occurrences of each candidate, ignoring any corrections that make up less than 0.5% of the corrections for a particular token. For instance, the word *among* has the following correction probabilities: *among* 0.7895, *from* 0.1053, *between* 0.0526. Even such a simplistic probability model has a noticeable effect on the system performance, improving the overall correction F-score by approximately 3%. The preposition substitution error detection F-score alone improves by 9%.

Prior to creating the probability model, we experimented with the addition of a bias toward the original token, which we hoped would reduce the number

Category	# Errors		Candidate Corrections	# Occurrences
	Original	Revised		
MD	125	131	a, an, another, any, her, his, its, my, our, that, the, their, these, this, those, which, your	-
RD	39	37	a, an, another, any, her, his, its, my, our, that, the, their, these, this, those, which, your	1924
RT	136	148	about, after, against, along, among, around, as, at, before, behind, below, between, by, concerning, considering, during, for, from, in, into, like, near, of, off, on, onto, out, outside, over, regarding, since, through, throughout, till, to, toward, towards, under, until, via, with, within, without	2202

Table 1: Single-Word Prepositions and Determiners with Error and Overall Frequency in Test Data

of overcorrections generated by our system. Without the probability model, a bias toward the original token improves the results, however, with the probability model, the bias is no longer useful.

2.2 Word Clustering

In the 2011 shared task, we observed that data sparsity issues are magnified in non-native texts because the n-gram context may contain additional errors or other infrequent or unusual n-gram sequences. We found that abstracting to part-of-speech tags and lemmas in certain contexts leads to small improvements in system performance. For the 2012 shared task, we explore the effects of abstracting to word clusters derived from co-occurrence information (Brown et al., 1992), another type of abstraction relevant to our n-gram prediction approach. We hypothesize that replacing tokens in the n-gram context in our prediction tasks with clusters will reduce the data sparsity for non-native text.

Clusters derived from co-occurrence frequencies offer an attractive type of abstraction that occupy a middle ground between relatively coarse-grained morphosyntactic abstractions such as part-of-speech tags and fine-grained abstractions such as lemmas. For determiner and preposition prediction, part-of-speech tags clearly retain too few distinctions. For example, the choice of *a/an* before a noun phrase depends on the onset of the first word in the phrase, information which is not preserved by part-of-speech tagging. Likewise, preposition selection may be dependent on lexical specifications (e.g., phrasal verbs such as *depend on*) or on semantic or world knowledge (cf. Wechsler, 1994).

Brown et al. (1992) present a hierarchical word clustering algorithm that can handle a large number of classes and a large vocabulary. The algorithm clusters a vocabulary into C clusters given a corpus to estimate the parameters of an n-gram language model. Summarized briefly, the algorithm first creates C clusters for the C most frequent words in the corpus. Then, a cluster is added containing the next most frequent word. After the new cluster is added, the pair of clusters is merged for which the loss in average mutual information is smallest, returning the number of clusters to C. The remaining words in the vocabulary are added one by one and pairs of clusters are merged in the same fashion until all words have been divided into C clusters.

Using the implementation from Liang (2005),[1] we generate word clusters for the most frequent 100,000 tokens in the ukWaC corpus (Baroni et al., 2009). We convert all tokens to lower case, replace all lower frequency words with a single unique token, and omit from the clustering the candidate corrections from Table 1 along with the low frequency tokens. Our corpus is the first 18 million sentences from ukWaC.[2] After converting all tokens to lowercase and omitting the candidate function words, a total of 75,333 tokens are clustered.

We create three sets of clusters with sizes 500, 1000, and 2000. Due to time constraints, we did not yet explore larger sizes. Brown et al. (1992) report that the words in a cluster appear to share syntactic or semantic features. The clusters we obtained appear to be overwhelmingly semantic in nature.

[1] Available at http://cs.stanford.edu/∼pliang/software

[2] Those sentences in the file ukwac_dep_parsed_01.

	Cluster ID	Selected Cluster Members
(1)	00100	was..., woz, wasn't, was, wasnt
(2)	0111110111101	definetly, definatly, assuredly, definately, undoubtedly, certainly, definitely
(3)	1001110100	extremely, very, incredibly, inordinately, exceedingly, awfully
(4)	1110010001	john, richard, peter, michael, andrew, david, stephen
(5)	11101001001	12.30pm, 7am, 2.00pm, 4.00pm, weekday, tuesdays

Table 2: Sample Clusters from ukWaC with 2000 Clusters

Table 2 shows examples from the set of 2000 clusters. Examples (1) and (2) show how tokens with errors in tokenization or misspellings are clustered with tokens with standard spelling and standard tokenization. Such clusters may be useful for the shared task by allowing the system to abstract away from spelling errors in the learner essays. Examples (3)–(5) show semantically similar clusters.

An excerpt of the hierarchical cluster tree for the cluster ID from example (3) is shown in Figure 1. The tree shows a subset of the clusters for cluster IDs beginning with the sequence 1001110. Each binary branch appends a 0 or 1 to the cluster ID as shown in the edge labels. The cluster 1001110100 (*extremely, very*) is found in the left-most leaf of the right branch. A few of the most frequent cluster members are shown for each leaf of the tree.

In our submissions to the shared task, we included five different cluster settings: 1) using the original word-based approach with no clusters, 2) using only 2000 clusters, 3) using the word-based approach initially and backing off to 2000 clusters if no hits are found, 4) backing off to 1000 clusters, and 5) backing off to 500 clusters. The detailed results will be presented in section 3.

2.3 Missing Determiner Detector

We newly developed a missing determiner detector to identify those places in the learner text where a determiner is missing. Since determiners mostly occur in noun phrases, we extract all noun phrases from the text and put them through a two-stage classifier. For a single-stage classifier, always predicting 'no error' leads to a very high baseline accuracy of 98%. Therefore, we first filter out those noun phrases which already contain a determiner, a possessive pronoun, another possessive token (e.g., *'s*), or an existential *there*, or whose head is a pronoun. This prefiltering reduces the baseline accuracy to 93.6%, but also filters out 10% of learner errors (*false negatives*), which thus cannot be detected in stage two.

In the second stage, a decision tree classifier decides for every remaining noun phrase whether a determiner is missing. From the 203 features we originally extracted to inform the classification, the chi squared algorithm selected 30. Almost all of the selected features capture properties of either the head of the noun phrase, its first word, or the token immediately preceding the noun phrase. We follow Minnen et al. (2000) in defining the head of a noun phrase as the rightmost noun, or if there is no noun, the rightmost token. As suggested by Han et al. (2004), the classifier considers the parts of speech of these three words, while the features that record the respective literal word were discarded.

We also experimented with using the entire noun phrase and its part-of-speech tag sequence as features (Han et al., 2004), which proved not to be helpful due to the limited size of the training data. We replaced the part-of-speech tag sequence with a number of boolean features that each indicate equivalence with a particular sequence. Of these features only the one that checks whether the whole noun phrase consists of a single common noun in the singular was included in the final feature set. Additionally, the selected features include countability information from noun countability lists generated by Baldwin and Bond (2003), which assign nouns to one or more countability classes: *countable*, *uncountable/mass noun*, *bipartite*, or *plural only*.

The majority of the 30 selected features refer to the position of one of the three tokens (head, first word, and preceding token) in the cluster hierarchy described in section 2.2. The set of 500 clusters proved not to be fine-grained enough, so we used

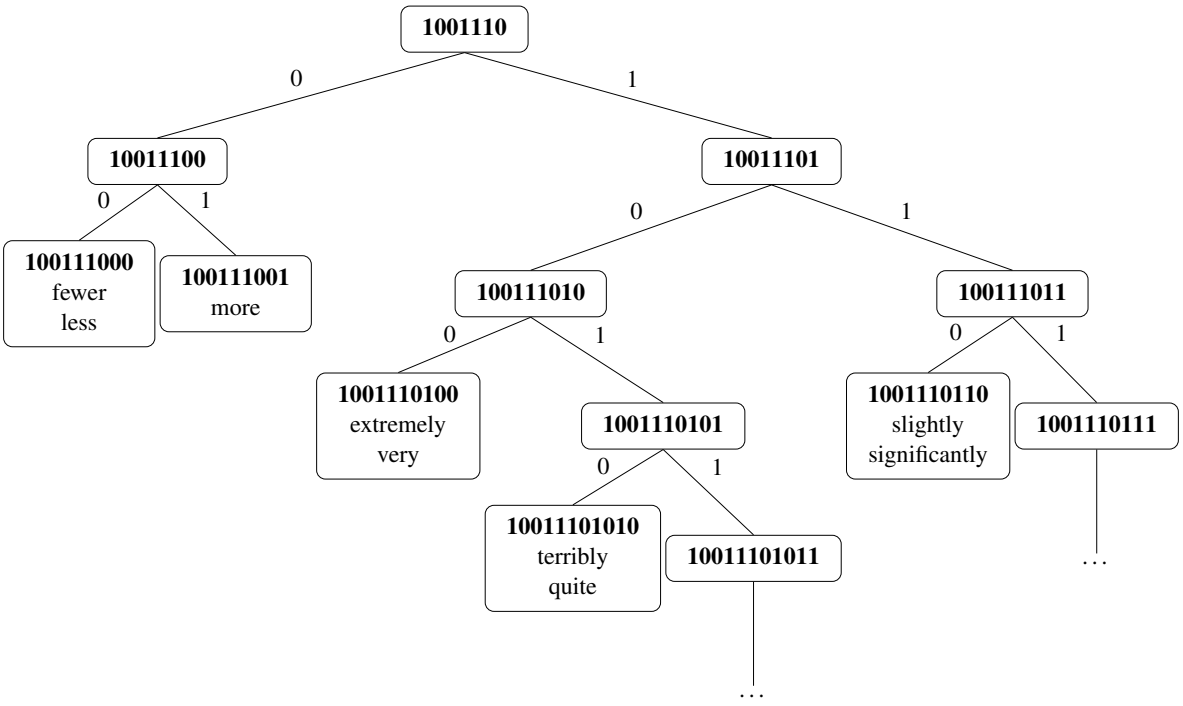

Figure 1: Hierarchical Clustering Subtree for Cluster Prefix 1001110

the set of 1000 clusters. To take full advantage of the hierarchical nature of the cluster IDs, we extract prefixes of all possible lengths (1–18 characters) from the cluster ID of the respective token. For the head and the first word, prefixes of length 3–14 were selected by the attribute selector, in addition to a prefix of length 6 for the preceding token's cluster ID.

Among the discarded features are many extracted from the context surrounding the noun phrase, including the parts of speech and cluster membership of three words to the left and right of the noun phrase, excluding the immediately preceding token. Features referring to possible sister conjuncts of the noun phrase, the next 3rd person pronoun in a following sentence, or previous occurrences of the head in the text also turned out not to be useful. The performance of the classifier was only marginally affected by the reduction in the number of features. We conclude from this that missing determiner detection is sufficiently informed by local features.

In order to increase the robustness of the classifier, we generated additional data from the written portion of the BNC by removing a determiner in 20% of all sentences. The resulting rate of errors is roughly equal to the rate of errors in the learner texts and the addition of the BNC data increases the amount of training data by a factor of 13. We trained a classifier on both datasets (referred to as *HOO-BNC* below). It achieves an F-score of 46.7% when evaluated on 30% of the shared task training data, which was held out from the classifier training data. On the revised test data, it reaches an F-score of 44.5%.

3 Results

The following two sections discuss our overall results for the shared task and our performance on the three error types targeted by our system.

3.1 Overall

Figure 2 shows the overall recognition and correction F-score for the cluster settings described in section 2.2. With the missing determiner detector *HOO-BNC* described in section 2.3, these correspond to runs #5–9 submitted to the shared task. For the unrevised data, Run #6 (2000 clusters only) gives our best result for overall detection F-score (30.26%) and Run #7 (2000 cluster backoff) for correction F-score (18.44%). For the revised data, Run

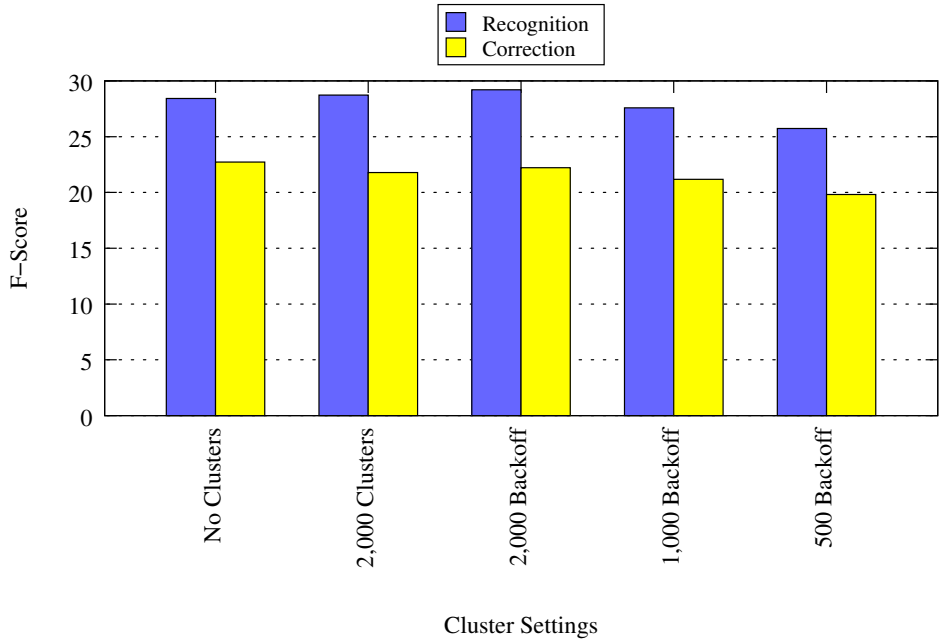

Figure 2: Recognition and Correction F-Score with Clustering

#7 (2000 cluster backoff) has our best overall detection F-score (32.21%) and Run #5 (no clusters) has our best overall correction F-score (22.46%).

Runs using clusters give the best results in two other metrics reported in the shared task results for the revised data. Run #6 (2000 clusters only) gives the best results for determiner correction F-score and Run #2 (2000 cluster backoff), which differs only from Run #7 in the choice of missing determiner detector, gives the best results for preposition detection and recognition F-scores.

The detailed results for Runs #5–9 with the revised data are shown in Figure 2. This graph shows that the differences between the systems with and without clusters are very small. The recognition F-score is best with 2000 cluster backoff and the correction F-score is best with no clusters. In both cases, the difference between the top two results is less than 0.01. There is, however, a noticeable increase in performance as the number of clusters increases, which indicates that a larger number of clusters may improve results further. The set of 2000 clusters may still retain too few distinctions for this task.

3.2 Targeted Error Types

Our system handles three of the six error types in the shared task: missing determiner (MD), incorrect determiner (RD), and incorrect preposition (RT). The recognition and correction F-scores for our best-forming run for each type are shown in Figure 3.

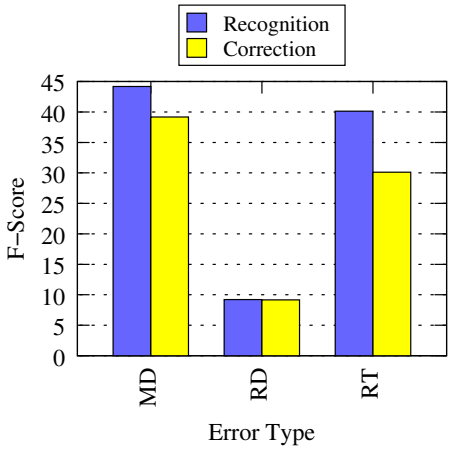

Figure 3: Recognition and Correction F-Score for the Targeted Error Types

In a comparison of performance on individual error types in the shared task, our system does best on the task for which it was originally developed,

preposition prediction. We place 4th in recognition and 3rd in correction F-score for this error type. For missing determiner (MD) and incorrect determiner (RD) errors, our system is ranked similarly as in our overall performance (4th–6th).

For the sake of replicability, as the HOO 2012 test data is not publicly available, we include our results on the HOO training data for the preposition and determiner substitution errors in Table 3.

Error Type	No Clusters			
	Recognition		Correction	
	Prec	Rec	Prec	Rec
RT	32.69	29.94	24.85	22.77
RD	10.63	18.56	8.37	14.61

Error Type	2000 Backoff			
	Recognition		Correction	
	Prec	Rec	Prec	Rec
RT	25.87	35.60	18.26	25.13
RD	9.71	23.65	7.48	18.23

Table 3: Results for HOO 2012 Training Data

Results are reported for the no cluster and 2000 cluster backoff settings, which show that incorporating the cluster backoff improves recall at the expense of precision. Missing determiner errors are not reported directly as the missing determiner detector was trained on the training data, but see the evaluation at the end of section 2.3.

4 Discussion and Conclusion

The n-gram prediction approach with the new missing determiner detector performed well in the HOO 2012 Shared Task, placing 6th in terms of detection and 5th in terms of correction out of fourteen teams participating in the shared task. In our best submissions evaluated using the revised test data, we achieved a detection F-score of 32.71%, a recognition F-score of 29.21% and a correction F-score of 22.73%. For the three error types addressed by our approach, our correction F-scores are 39.17% for missing determiners, 9.23% for incorrect determiners, and 30.12% for incorrect prepositions. Information from hierarchical word clustering (Brown et al., 1992) extended the types of abstractions available to our n-gram prediction approach and improved the performance of the missing determiner detector.

For the n-gram prediction approach, word clusters IDs from the hierarchical word clustering replace tokens in the surrounding context in order to improve recall for learner texts which may contain errors or infrequent token sequences. The use of cluster-based contexts with 2000 clusters as a backoff from the word-based approach leads to a very small improvement in the overall recognition F-score for the HOO 2012 Shared Task, but our best overall correction F-score was obtained using our original word-based approach. The differences between the word-based and cluster-based approaches are quite small, so we did not see as much improvement from the word cluster abstractions as we had hoped. We experimented with sets of clusters of several sizes (500, 1000, 2000) and found that as the number of clusters becomes smaller, the performance decreases, suggesting that a larger number of clusters may lead to more improvement for this task.

Information from the word cluster hierarchy was also integrated into our new missing determiner detector, which uses a decision tree classifier to decide whether a determiner should be inserted in front of a determiner-less NP. Lexical information from the extracted noun phrases and surrounding context are not as useful for the classifier as information about the position of the tokens in the word cluster hierarchy. In particular, cluster information appears to help compensate for lexical sparsity given a relatively small amount of training data.

In future work, we plan to explore additional clustering approaches and to determine when the use of word cluster abstractions is helpful for the task of predicting determiners, prepositions, and other function words. An approach that refers to word clusters in certain contexts or in a customized fashion for each candidate correction may lead to improved performance for the task of detecting and correcting such errors in texts by non-native writers.

References

Timothy Baldwin and Francis Bond, 2003. Learning the countability of English nouns from corpus data. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*. pp. 463–470.

M. Baroni, S. Bernardini, A. Ferraresi and E. Zanchetta, 2009. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

Shane Bergsma, Dekang Lin and Randy Goebel, 2009. Web-scale N-gram models for lexical disambiguation. In *Proceedings of the 21st international jont conference on Artifical intelligence (IJCAI'09)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Steven Bird, Robert Dale et al., 2008. The ACL Anthology Reference Corpus. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*. Marrakesh, Morocco.

Adriane Boyd and Detmar Meurers, 2011. Data-Driven Correction of Function Words in Non-Native English. In *Proceedings of the 13th European Workshop on Natural Language Generation – Helping Our Own (HOO) Challenge*. Association for Computational Linguistics, Nancy, France.

Thorsten Brants and Alex Franz, 2006. Web 1T 5-gram Version 1. Linguistic Data Consortium. Philadelphia.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, T. J. Watson, Vincent J. Della Pietra and Jenifer C. Lai, 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

Robert Dale and Adam Kilgarriff, 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation*. Nancy, France.

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

Anas Elghafari, Detmar Meurers and Holger Wunsch, 2010. Exploring the Data-Driven Prediction of Prepositions in English. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*. Beijing.

Michael Gamon, Jianfeng Gao et al., 2008. Using Contextual Speller Techniques and Language Modeling for ESL Error Correction. In *Proceedings of the Third International Joint Conference on Natural Language Processing*. Hyderabad.

Na-Rae Han, Martin Chodorow and Claudia Leacock, 2004. Detecting Errors in English Article Usage with a Maximum Entropy Classifier Trained on a Large, Diverse Corpus. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*. Lisbon.

Percy Liang, 2005. Semi-Supervised Learning for Natural Language. Master's thesis, Massachusetts Institute of Technology.

Guido Minnen, Francis Bond and Ann Copestake, 2000. Memory-based learning for article generation. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning*. volume 7, pp. 43–48.

Nick Rizzolo and Dan Roth, 2007. Modeling Discriminative Global Inference. In *Proceedings of the First International Conference on Semantic Computing (ICSC)*. IEEE, Irvine, California, pp. 597–604.

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 (ACL-HLT)*. Portland, Oregon.

Joel Tetreault and Martin Chodorow, 2008. Native Judgments of Non-Native Usage: Experiments in Preposition Error Detection. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*. Manchester.

Stephen Wechsler, 1994. Preposition Selection Outside the Lexicon. In Raul Aranovich, William Byrne, Susanne Preuss and Martha Senturia (eds.), *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics*. CSLI Publications, Stanford, California, pp. 416–431.

H. Yannakoudakis, T. Briscoe and B. 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 (ACL-HLT)*.