NUS at the HOO 2011 Pilot Shared Task

Daniel Dahlmeier[1], **Hwee Tou Ng**[1,2], and **Thanh Phu Tran**[2]
[1]NUS Graduate School for Integrative Sciences and Engineering
[2]Department of Computer Science, National University of Singapore
{danielhe,nght,thanhphu}@comp.nus.edu.sg

Abstract

This paper describes the submission of the National University of Singapore (NUS) to the Helping Our Own (HOO) Pilot Shared Task. Our system targets spelling, article, and preposition errors in a sequential processing pipeline.

1 Introduction

Helping Our Own (HOO) (Dale and Kilgarriff, 2010) is a new shared task for automatic grammatical error correction, a task which has attracted increasing attention recently. Instead of correcting errors in a general domain, e.g., essays written by second language learners of English, HOO focuses on papers written by non-native authors of English within the natural language processing community. In this paper, we describe the participating system from the National University of Singapore (NUS). The system targets spelling, article, and preposition errors. The core of our system is built on linear classification models and a large language model filter. We present experimental results on the HOO development and test data.

The next section describes the system in more detail. Section 3 describes the data sets used. Section 4 reports experimental results on the HOO development and test data.

2 System Architecture

The NUS system consists of a sequential pipeline of three processing steps:

1. Spelling correction
2. Article correction
3. Preposition correction

Sentence segmentation and tokenization are carried out on the HOO input files in a pre-processing step. Sentence segmentation uses the gold standard sentence boundaries. Each subsequent step takes a one-sentence-per-line plain text as input and outputs a one-sentence-per-line plain text in return. A post-processing step detokenizes the text and extracts the edit structures that encode the corrections.

2.1 Spelling Correction

We use the open-source spell checker Aspell[1] to correct spelling errors. Words are excluded from spelling correction if they are shorter than a threshold, or if they include hyphens or upper case characters inside the word. We use an in-domain Aspell dictionary constructed from all words that appear at least ten times in the ACL-ANTHOLOGY data set described in Section 3. Finally, we filter the corrections using a language model. The system only keeps corrections that strictly increase the normalized language model score of the sentence, defined as $\frac{1}{n} \log P$, where n is the length of the sentence, and P the language model probability.

2.2 Article Errors

Article error correction is treated as a multi-class classification problem. The possible classes are the articles *a*, *the*, and the empty article. The article *an* is normalized as *a* and restored later using a rule-based heuristic.

[1]http://aspell.net

Each input sentence is tagged with part-of-speech (POS) tags and syntactic chunks. We use OpenNLP[2] for POS tagging and YamCha (Kudo and Matsumoto, 2003) for chunking. For each noun phrase (NP), the system extracts a feature vector representation. We use the features proposed in (Han et al., 2006) which include the words before, in, and after the NP, the head word, POS tags, etc. A multi-class classifier then predicts the most likely article for the NP. We employ a linear classifier trained with empirical risk minimization on NP instances from well-edited text (Dahlmeier and Ng, 2011). The features are only extracted from the surrounding context of the article and do not include the article itself, which would be fully predictive of the class.

During testing, a correction is proposed if the predicted article is not the same as the observed article used by the writer, and the difference between the confidence score for the predicted article and the confidence score for the observed article is larger than a threshold. Finally, we filter the corrections using a large language model and only keep corrections that strictly increase the normalized language model score of the sentence.

2.3 Preposition Errors

Preposition error correction follows the same strategy of multi-class classification and language model filtering. The system only corrects preposition substitution errors, not preposition insertion or deletion errors. The possible classes are the prepositions *about, among, at, by, for, in, into, of, on, to*, and *with*. For each prepositional phrase (PP) which is headed by one of these prepositions, a linear classifier predicts the most likely preposition from the above list. We use the features proposed by (Tetreault and Chodorow, 2008). Again, we apply a threshold to bias the classifier towards the observed preposition and filter corrections with a large language model.

3 Data Sets

We randomly split the files in the HOO development data into a tuning set HOO-TUNE (9 files) and a held-out test set HOO-HELDOUT (10 files). The official HOO test data HOO-TEST is completely unobserved during development. We create two training data sets from the ACL Anthology[3]: ACL-ANTHOLOGY includes all non-OCR documents from the anthology except the 2010 ACL conference and workshop proceedings as these overlap with the HOO data[4]. CL-JOURNAL contains all non-OCR documents from the *Computational Linguistics* journal. In both cases, we filter out section headings, references, tables, etc. The WEB 1T 5-GRAM CORPUS (Brants and Franz, 2006) is used for language modeling. Table 1 gives an overview of the data sets.

Data Set	Sentences	Tokens
HOO-TUNE	477	12,115
HOO-HELDOUT	462	10,691
HOO-TEST	722	18,789
ACL-ANTHOLOGY	708,129	18,020,431
CL-JOURNAL	22,934	611,334

Table 1: Overview of the data sets.

4 Experiments and Results

This section reports experimental results of our system on the HOO-HELDOUT and the HOO-TEST data set. The parameters of the system are as follows. The minimum length for spelling correction is four characters. The language model filter for article and preposition correction uses a 5-gram language model built from the complete WEB 1T 5-GRAM CORPUS using RandLM (Talbot and Osborne, 2007). For spelling correction, the language model filter is built from the ACL-ANTHOLOGY data set. The linear classifiers for article and preposition correction are trained on the CL-JOURNAL data set. Threshold parameters are tuned on HOO-TUNE when testing on HOO-HELDOUT, and on the complete HOO development data when testing on HOO-TEST.

4.1 Evaluation

We report micro-averaged detection, recognition, and correction F_1 scores as defined in the HOO overview paper. The scores are computed over the entire test collection.

For individual error categories, the HOO overview paper only reports the "percentage of

[2] http://opennlp.sourceforge.net

[3] http://www.aclweb.org/anthology-new

[4] Although the use of the HOO source documents was permitted, we believe that excluding them is more realistic.

Step	Detection		Recognition		Correction	
	wb	w/o b	wb	w/o b	wb	w/o b
PRE	.2152	.0000	.2152	.0000	.2152	.0000
+SPEL	.2219	.0095	.2190	.0063	.2162	.0031
+ART	.2681	.1093	.2520	.0917	.2455	.0846
+PREP	.2973	.1354	.2763	.1123	.2657	.1008

Table 2: Overall F_1 scores with (wb) and without bonus (w/o b) on the HOO-HELDOUT data after pre-processing (PRE), spelling (SPEL), article (ART), and preposition correction (PREP).

Step	Detection		Recognition		Correction	
	wb	w/o b	wb	w/o b	wb	w/o b
PRE	.1553	.0000	.1553	.0000	.1553	.0000
+SPEL	.1663	.0093	.1629	.0093	.1611	.0075
+ART	.2718	.1552	.2545	.1373	.2209	.1014
+PREP	.2840	.1774	.2686	.1615	.2274	.1177

Table 3: Overall F_1 scores with (wb) and without bonus (w/o b) on the HOO-TEST data.

instances in each category that were detected, recognized and corrected", but not precision or F_1 scores. Computing precision and F_1 is complicated by the fact that the HOO submission format does not require a system to "label" each proposed correction with the intended error category. As we know which correction was produced by which processing step for our own system, we know which error category a correction belongs to. Therefore, we can calculate micro-averaged precision, recall, and F_1 scores for spelling, article, and preposition errors individually by restricting the set of proposed edits and the set of gold corrections to a particular category.

4.2 Results

Tables 2 and 3 show the overall detection, recognition, and correction F_1 scores after each processing step on the HOO-HELDOUT and HOO-TEST set, respectively. Each processing step builds on the output of the previous step. The single biggest improvement in the score comes from the article correction step. The gap between the scores with and without bonus shows the large number of optional corrections in the HOO data. Tables 4 and 5 show the detection, recognition, and correction F_1 scores for individual error categories on the HOO-HELDOUT and HOO-TEST set, respectively.

Step	Detection		Recognition		Correction	
	wb	w/o b	wb	w/o b	wb	w/o b
SPEL	.2667	.2667	.2667	.2667	.2667	.2667
ART	.3455	.3011	.3455	.3011	.3246	.2796
PREP	.2692	.2353	.2308	.1961	.1731	.1373

Table 4: Individual F_1 scores for each error category with (wb) and without bonus (w/o b) on the HOO-HELDOUT data.

Step	Detection		Recognition		Correction	
	wb	w/o b	wb	w/o b	wb	w/o b
SPEL	.4706	.4706	.4706	.4706	.4706	.4706
ART	.3591	.3404	.3466	.3277	.2630	.2426
PREP	.3409	.2000	.3409	.2000	.2614	.1200

Table 5: Individual F_1 scores for each error category with (wb) and without bonus (w/o b) on the HOO-TEST data.

Acknowledgments

This research was done for CSIDM Project No. CSIDM-200804 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore.

References

T. Brants and A. Franz. 2006. Web 1T 5-gram corpus version 1.1. Technical report, Google Research.

D. Dahlmeier and H.T. Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proceedings of ACL*.

R. Dale and A. Kilgarriff. 2010. Helping Our Own: Text massaging for computational linguistics as a new shared task. In *Proceedings of INLG*.

N.-R. Han, M. Chodorow, and C. Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2).

T. Kudo and Y. Matsumoto. 2003. Fast methods for kernel-based text analysis. In *Proceedings of ACL*.

D. Talbot and M. Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of ACL*.

J. R. Tetreault and M. Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of COLING*.