Detecting Missing Hyphens in Learner Text

Aoife Cahill[*], Martin Chodorow[†], Susanne Wolff[*] and Nitin Madnani[*]

[*] Educational Testing Service, 660 Rosedale Road, Princeton, NJ 08541, USA
{acahill, swolff, nmadnani}@ets.org
[†] Hunter College and the Graduate Center, City University of New York, NY 10065, USA
martin.chodorow@hunter.cuny.edu

Abstract

We present a method for automatically detecting missing hyphens in English text. Our method goes beyond a purely dictionary-based approach and also takes context into account. We evaluate our model on artificially generated data as well as naturally occurring learner text. Our best-performing model achieves high precision and reasonable recall, making it suitable for inclusion in a system that gives feedback to language learners.

1 Introduction

While errors of punctuation are not as frequent, nor often as serious, as some of the other typical mistakes that learners make, they are nevertheless an important consideration for students aiming to improve the overall quality of their writing. In this paper we focus on the error of missing hyphens. The following example is a typical mistake made by a student writer:

(1) Schools may have more after school sports.

In this case the tokens *after* and *school* should be hyphenated as they modify the noun *sports*. However, in Example (2) a hyphen between *after* and *school* would be incorrect, since in this instance *after* functions as as the head of a prepositional phrase modifying *went*.

(2) I went to the dentist after school today.

These examples illustrate that purely dictionary-based approaches to detecting missing hyphens are not likely to be sophisticated enough to differentiate the contexts in which a hyphen is required. In addition, learner text frequently contains other grammatical and spelling errors, further complicating automatic error detection. Example (3) contains an error *father like* instead of *father likes to*. This causes difficulty for automated hyphenation systems because *like* is a frequent suffix of hyphenated words and *play* can function as a noun.

(3) My father like play basketball with me.

In this paper, we propose a classifier-based approach to automatically detecting missing hyphen errors. The goal of our system is to detect missing hyphen errors and provide feedback to language learners. Therefore, we place more importance on the precision of the system than recall. We train our model on features that take the context of a pair of words into account, as well as other discriminative features. We present a number of evaluations on both artificially generated errors and naturally occurring learner errors and show that our classifiers achieve high precision and reasonable recall.

2 Related Work

The task of detecting missing hyphens is related to previous work on detecting punctuation errors. One of the classes of errors in the Helping Our Own (HOO) 2011 shared task (Dale and Kilgarriff, 2011) was punctuation. Comma errors are the most frequent kind of punctuation error made by learners. Israel et al. (2012) present a model for detecting these kinds of errors in learner texts. They train CRF models on sentences from unedited essays written by high-level college students and show that they performs well on detecting errors in learner text. As

Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 300–305,
Atlanta, Georgia, June 13 2013. ©2013 Association for Computational Linguistics

far as we are aware, the HOO 2011 system description of Rozovskaya et al. (2011) is the only work to specifically reference hyphen errors. They use rules derived from frequencies in the training corpus to determine whether a hyphen was required between two words separated by white space.

The task of detecting missing hyphens is related to the task of inserting punctuation into the output of unpunctuated text (for example, the output of speech recognition, automatic generation, machine translation, etc.). Systems that are built on the output of speech recognition can obviously take features like prosody into account. In our case, we are dealing only with written text. Gravano et al. (2009) present an *n*-gram-based model for automatically adding punctuation and capitalization to the output of an ASR system, *without* taking any of the speech signal information into account. They conclude that more training data, rather than wider *n*-gram contexts leads to a greater improvement in accuracy.

3 Baselines

We implement three baseline systems which we will later compare to our classification approach. The first baseline is a naïve heuristic that predicts a missing hyphen between bigrams that appear hyphenated in the Collins Dictionary.[1] As a somewhat less-naïve baseline, we implement a heuristic that predicts a missing hyphen between bigrams that occur hyphenated more than 1,000 times in Wikipedia. A third baseline is a heuristic that predicts a missing hyphen between bigrams where the probability of the hyphenated form as estimated from Wikipedia is greater than 0.66, meaning that the hyphenated bigram is twice as likely as the non-hyphenated bigram. This baseline is similar to the approach taken by Rozovskaya et al. (2011), except that the probabilities are estimated from a much larger corpus.

4 System Description

Using the features in Table 1, we build a logistic regression model which assigns a probability to the likelihood of a hyphen occurring between two words, w_i and w_{i+1}. As we are primarily interested in using this system for giving feedback to language learners, we require very high precision. Therefore, we only predict a missing hyphen error when the probability of the prediction is >0.99.

Tokens	w_{i-1}, w_i, w_{i+1}, w_{i+2}
Stems	s_{i-1}, s_i, s_{i+1}, s_{i+2}
Tags	t_{i-1}, t_i, t_{i+1}, t_{i+2}
Bigrams	w_i–w_{i+1}, s_i–s_{i+1}, t_i–t_{i+1}
Dict	Does the hyphenated form appear in the Collins dictionary?
Prob	What is the probability of the word bigram appearing hyphenated in Wikipedia?
Distance	Distance to following and preceding verb, noun
Verb/Noun	Is there a verb/noun preceding/following this bigram

Table 1: Features used in all models. Positive instances are those where there was a hyphen between w_i and w_{i+1} in the data. Stems are generated using NLTK's implementation of the Lancaster Stemmer, and tags are obtained from the Stanford Parser.

We experiment with two different sources of training data, in addition to their combination. We first train on well-edited text, using almost 1.8 million sentences from the San Jose Mercury News corpus.[2] For training, hyphenated words are automatically split (i.e. *well-known* becomes *well known*). The positive examples for the classifier are all bigrams where a hyphen was removed. Negative examples consist of bigrams where there was no hyphen in the training data. Since this is over 99% of the data, we randomly sample 3% of the negative examples for training. We also restrict the negative examples to only the most likely contexts, where a context is defined as a part-of-speech bigram. A list of possible contexts in which hyphens occur is extracted from the entire training set. Only contexts that occur more than 20 times are selected during training. All contexts are evaluated during testing. Table 2 lists some of the most frequent contexts with examples of when they should be hyphenated and when they should remain unhyphenated.

The second data source for training the model comes from pairs of revisions from Wikipedia articles. Following Cahill et al. (2013), we automatically extract a corpus of error annotations for miss-

[1] LDC catalog number LDC93T1

[2] LDC catalog number LDC93T3A.

Context	Hyphenated	Unhyphenated
NN NN	terrific *truck-stop* waitress	a *quake insurance* surcharge
CD CD	*Twenty-two* thousand	the *126 million* Americans
JJ NN	an *early-morning* blaze	an *entire practice* session
CD NN	a *two-year* contract	about *600 tank* cars
NN VBN	a *court-ordered* program	a *letter delivered* today

Table 2: Some frequent likely POS contexts for hyphenation, with examples from the Brown corpus.

	TP	P	R	F
Baseline				
Collins dict	397	40.5	19.2	26.0
Wiki Counts-1000	359	39.1	17.3	24.0
Wiki Probs-0.66	811	**85.5**	39.1	53.7
Classifier				
SJM-trained	1097	82.0	52.9	**64.3**
Wiki-revision-trained	1061	72.8	51.2	60.1
Combined	1106	80.9	53.4	**64.3**

Table 3: Results of evaluating on the Brown Corpus with hyphens removed

ing hyphens. This is done by extracting the plain text from every revision to every article and comparing adjacent pairs of revisions. For each article, chains of errors are detected, using the surrounding text to identify them. When a chain begins and ends with the same form, it is ignored. Only the first and last points in an error chain are retained for training. An example chain is the following: It has been an ancient {*focal point* → *location* → *focal point* → *focal-point*} of trade and migration., where we would extract the correction *focal point* → *focal-point*. In total, we extract a corpus of 390,298 sentences containing missing hyphen error annotations.

Finally, we combine both data sources.

5 Evaluating on Artificial Data

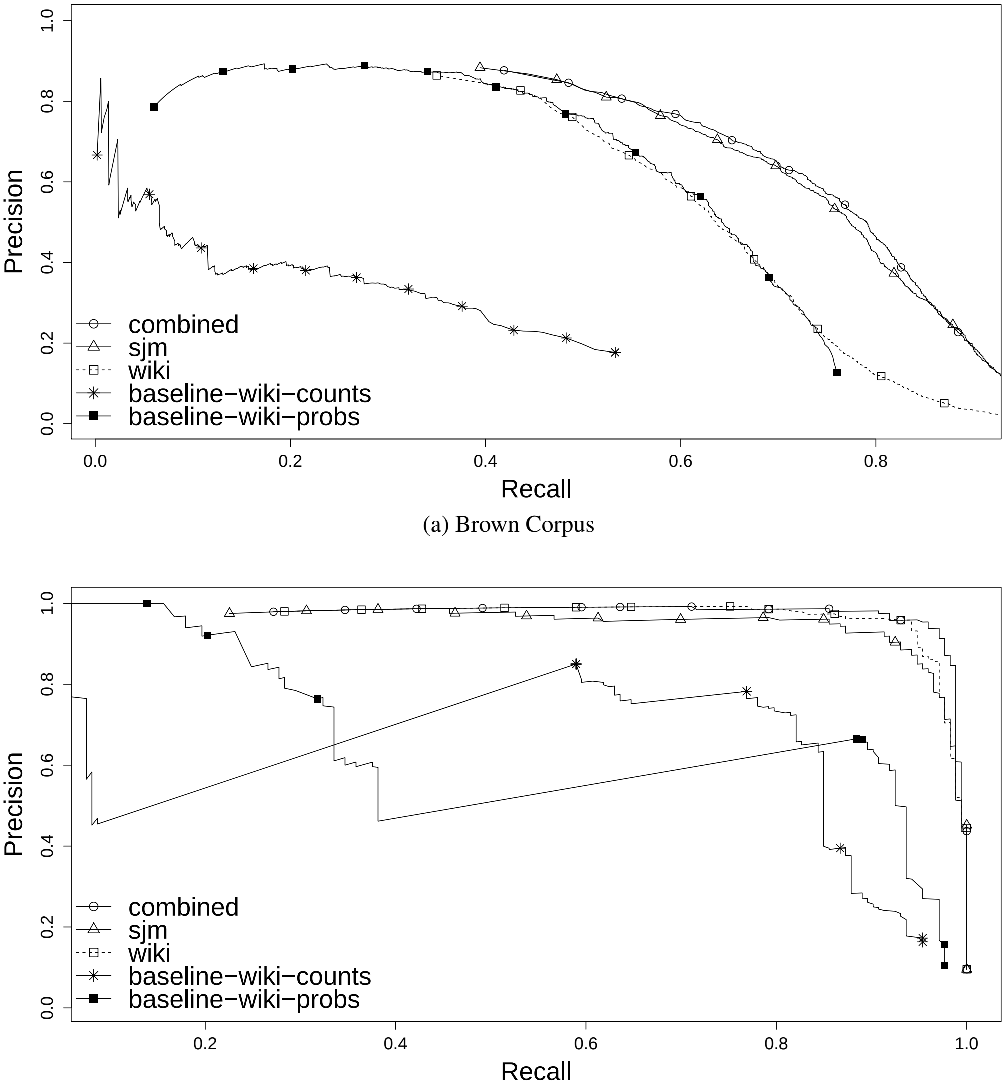
Since there are large corpora of well-edited text readily available, it is easy to evaluate on artificial data. For testing, we take 24,243 sentences from the Brown corpus and automatically remove hyphens from the 2,072 hyphenated words (but not free-standing dashes). Each system makes a prediction for all bigrams about whether a hyphen should appear between the pair of words. We measure the performance of each system in terms of precision, P, (how many of the missing hyphen errors predicted by the system were true errors), recall, R, (how many of the artificially removed hyphens the system detected as errors) and f-score, F, (the harmonic mean of precision and recall). The results are given in Table 3, and also include the raw number of true positives, TP, detected by each system. The results show that the baseline using Wikipedia probabilities obtains the highest precision, however with low recall. The classifiers trained on newswire text and the combined news and Wikipedia revision text achieve the highest overall f-score. Figure (1a) shows the Precision Recall curves for the Wikipedia baselines and the three classifiers. The curves mirror the results in the table, showing that the classifier trained on the newswire text, and the classifier trained on the combined data perform best. The Wikipedia counts baseline performs worst.

6 Evaluating on Learner Text

We carry out two evaluations of our system on learner text. We first evaluate on the missing hyphen errors contained in the CLC-FCE (Yannakoudakis et al., 2011). This corpus contains 1,244 exam scripts written by learners of English as part of the Cambridge ESOL First Certificate in English. In total, there are 173 instances of missing hyphen errors. The results are given in Table 4, and the precision recall curves are displayed in Figure (1b).

The results show that the classifiers consistently achieve high precision on this data set. This is as expected, given the high threshold set. Looking at the curves, it seems that a slightly lower threshold in this case may lead to better results. The curves show that the combined classifier is performing slightly better than the other two classifiers. The baselines are clearly not performing as well on this dataset.

While the overall size of the CLC-FCE data set is quite large, the low frequency of this kind of error means that the evaluation was carried out on a relatively small number of examples. For this reason, the reliability of the results may be called into question. There is, for instance, a striking difference between the f-scores for the Collins Dictionary base-



(a) Brown Corpus

(b) CLC-FCE Corpus

Figure 1: Precision Recall curves for the Wikipedia baselines and the three classifiers.

	TP	P	R	F
Baseline				
Collins dict	131	64.5	75.7	69.7
Wiki Counts-1000	141	73.1	**81.5**	**77.0**
Wiki Probs-0.66	36	92.3	20.8	34.0
Classifier				
SJM-trained	60	84.5	34.7	49.2
Wiki-revision-trained	71	**98.6**	41.0	58.0
Combined	66	98.5	38.2	55.0

Table 4: Results of evaluating on the CLC-FCE dataset

line on the Brown corpus (26.0) and on the learner data (69.7). Inspection of the 131 true positives for the learner data reveal that 87 of these are cases of a single type, the word "make-up", which students often wrote without a hyphen in response to a prompt about a fashion and leisure show. Since the hyphenated form was in the Collins Dictionary, the baseline system was credited with detection of this error. However, when the 87 occurrences of "make up" are removed from the data set, the values of precision, recall and f-score for the Collins Dictionary baseline fall to 37.9, 51.2, and 42.9, respectively. This points to a problem for system evaluation that is more general than the low frequency of an error type, such as missing hyphens. The more general problem is that of non-independence among errors, which occurs when an individual writer contributes multiple times to an error count or when a particular prompt gives rise to many occurrences of the same error, as in the current case of "make-up".

Despite the problem of non-independent errors, a more accurate picture of system performance may nonetheless emerge with more evidence. Therefore, we evaluate system precision on a data set of 1,000 student GRE and TOEFL essays written by both native and nonnative speakers, across a wide range of proficiency levels and prompts. The essays, drawn from 295 prompts, ranged in length from 1 to 50 sentences, with an average of 378 words per essay.

We manually inspect a random sample of 100 instances where each system detected a missing hyphen. Two native-English speakers judged the correctness of the predictions using the Chicago Manual of Style as a guide.[3] Inter-annotator agreement on the binary classification task for 600 items was 0.79κ, showing high agreement. The results are given in Table 5.

	Total Predictions	Judge-1 Precision	Judge 2 Precision
Baseline			
Collins dict	416	11	8
Wiki Counts	2185	20	21
Wiki Probs	224	54	52
Classifier			
SJM-trained	421	62	69
Wiki-revision	577	43	41
Combined	450	60	62

Table 5: Precision results on 1000 student responses, estimated by randomly sampling 100 hyphen predictions of each system and manually evaluating them.

The results show that the first two baseline systems do not perform well on this essay data. This is mainly because they do not take context into account. Many of the errors made by these systems involved verb + preposition bigrams, as in Examples (4) and (5). Restricting the detection by probability clearly improves precision, but at the cost of recall (only 224 total instances of missing hyphen errors detected, the lowest of all 6 systems). In the manual evaluation, the system trained on the SJM corpus achieves the highest precision, though all precision figures are lower than the previous evaluations. Example (6) is a typical example of the kinds of false positives made by the classifier models.

(4) If these men were required to step-down after a limited number of years, the damage would be contained.

(5) These families may even choose to eat at-home than outside.

(6) The wellness program will save money in the long-term.

Future work will explore additional features that may help improve performance. A more thorough study will also be carried out to fully understand the differences in performance of the classifiers across corpora. Another direction to explore in future work is the related task of identifying extraneous hyphens in learner text. These are even less frequent than missing hyphens (87 annotated cases in the CLC-FCE corpus), but we believe a similar classification approach could be successful.

7 Conclusion

In this paper we presented a model for automatically detecting missing hyphen errors in learner text. We experimented with two kinds of training data, one well-edited text, and the other an automatically extracted corpus of error annotations. When evaluating on artificially generated errors in otherwise well-edited text, the classifiers generally performed better than the baseline systems. When evaluating on the small number of missing hyphen errors in the CLC-FCE corpus, the word-based models did well, though the classifiers also achieved consistently high precision. A precision-only evaluation on a sample of learner essays resulted in overall lower scores, but the classifier trained on well-edited text performed best. In general, the classifiers outperform the baseline, especially in terms of precision, showing that taking context into account when detecting these kinds of errors is important.

[3] http://www.chicagomanualofstyle.org

References

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust Systems for Preposition Error Correction Using Wikipedia Revisions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, GA.

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

Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4741–4744. IEEE.

Ross Israel, Joel Tetreault, and Martin Chodorow. 2012. Correcting Comma Errors in Learner Essays, and Restoring Commas in Newswire Text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 284–294, Montréal, Canada, June. Association for Computational Linguistics.

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

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA, June. Association for Computational Linguistics.