

# Learning a Copyeditor from Wikipedia

**Elif Yamangil**

School of Engineering and Applied Sciences  
Harvard University  
Cambridge, MA 02138, USA  
elif@eecs.harvard.edu

## Abstract

Copyediting is the editorial work that an editor does to make spelling, grammar, and style changes and improvements to a manuscript. In this paper we aim at building an automatic copyeditor. Given the complexity of the task, we take a machine learning approach. To obtain training data, we use 1.4 million articles from Wikipedia with their entire revision histories. We mine the historical data to extract a dataset of 7.7 million training examples. We use the dataset to train a Hidden Markov Model of the generative process of writing, and perform Viterbi decoding to search for the most likely correction given a test sentence. We evaluate our method against context-sensitive spelling correction, and not only obtain state-of-the-art levels of accuracy but also redefine the task by eliminating some of its severe limitations on scalability and coverage. We demonstrate that our knowledge-lean, purely data-driven method is capable of corrections that extend well beyond any form of spelling correction, such as grammar correction and stylistic correction.

## 1 Introduction

Text is often fraught with errors in spelling, grammar, and style. Although modern word processors provide support for text correction, including context-sensitive spelling correction and grammar checking, even the most sophisticated tools still fall short of catching all errors.<sup>1</sup>

<sup>1</sup>See <http://faculty.washington.edu/sandeep/check/> for a critique of a popular commercial text editor's correction capabilities.

Given a collection of typical errors and their corrections, supervised learning approaches can help to automatically correct text, possibly more accurately and with broader coverage than the unsupervised approaches that the natural language processing community so far has had to offer. For instance, for context-sensitive spelling correction, which is the task of disambiguating between words that are likely to be confused with each other such as *peace-piece*, Carlson et al. (2001) presented a Winnow-based algorithm achieving accuracy levels in the 99% range. Despite the high success rate, this approach uses only 265 such tuples of words, and therefore it suffers from being limited to these predefined tuples called “confusion sets”. In practice, this brings the drawbacks of requiring manual knowledge engineering in the form of inducing the confusion sets, separate training and feature selection per confusion set, and the more crucial drawback of not having a broad coverage over all possible errors. As it was pointed out by Carlson et al. themselves, a more ideal model would have the power to replace any word with any other word. In effect, it would have one very large confusion set that is “English”, which is exactly what our model achieves.

Nelken and Yamangil (2008) suggest that typical errors and their corrections can be automatically harvested from Wikipedia article revisions, which they validate by extracting corrections of a specific form of lexical error, known as “Eggcorns”. Since Wikipedia articles are iteratively edited by many Web users, by comparing adjacent revisions of the same article, it is possible to collect Wikipedians’ editorial decisions.

In this paper we push the state of the art in automatic text correction in two directions. We follow the same method of mining Wikipedia to obtain large-scale parallel corpora for supervised text correction, the first one ever. We limit context to a sentence, and frame the text correction task as a word replacement task, ignoring the possibility of correcting a sentence by reordering or replacing words and phrases. We get rid of the limiting notion of a confusion set, and propose a knowledge-lean generative model of writing, which allows us to estimate the joint probability of a correct and a potentially incorrect sequence of words, coupled with a Viterbi decoding that efficiently searches for the optimal correction given a test sentence. We demonstrate that such a pure and data-driven method, which has no idea what a confusion set is, can achieve the state-of-the-art levels of accuracy on the task of disambiguating between confusion set members, as well as provide examples of the more general corrections that this method is capable of.

In section 2 of the following, we describe how we make use of Wikipedia to obtain training data. We discuss our method in sections 3 to 5 with details about the probabilistic structure of the generative model, smoothing the model parameters, and decoding. Section 6 introduces our evaluation results and some insights that can be taken away from our experiment, followed by section 7, where we conclude with a brief discussion about the future of this work.

## 2 Data: 7.7 Million corrections from Wikipedia

Many researchers are increasingly turning to Wikipedia as a large-scale data source. The vast majority of this work uses only the most recent version of the articles. In fact, Wikipedia conveniently provides not only the latest version, but the entire revision history of each of its articles, as dramatically visualized by Viégas et al. (2004). Through Wikipedia’s collaborative editing process, articles are iteratively amended and refined by multiple Web users. Users can usually change any aspect of the document’s structure and content, but for our purposes here, we focus only on sentence-level edits.

We use the July 2006 snapshot of the English Wikipedia, consisting of 1.4 million articles, and

mined them for such edits. For each article, we first extract all revisions, and split each revision into a list of its sentences.<sup>2</sup> We run an edit-distance comparison between each such pair, treating each sentence as an atomic “letter”. We look for all replacements of one sentence by another and check whether the sentences differ by one word.

We make the assumption that *all* such edits are actually corrections, and therefore valid training data for our purposes. This assumption is of course patently naïve, as there are cases in which such revisions correct factual information, resolve pronouns<sup>3</sup>, introduce errors, are part of a flame war, etc.

In this way, we extracted about 7.7 million sentence pairs.<sup>4</sup> Wikipedia currently has 2.3 million articles and is constantly expanding both in the number of articles and in the number of revisions.

## 3 Generative Model of Writing

We model the process of writing using a Hidden Markov Model (HMM). We define a correct sequence of words  $C = c_1, c_2, \dots, c_n$  and a corresponding potentially incorrect observation sequence of words  $O = o_1, o_2, \dots, o_n$ . We expand the joint probability  $P(C, O)$  of intending  $C$  but writing  $O$  into a Markov chain with the following independence structure.

$$P(C, O) = \prod_{j=1}^n P(c_j | c_{j-2}, c_{j-1}) \cdot P(o_j | c_j)$$

This implies that during writing the intended correct words are chosen conditioned on the previous two words, and each such word is output as a version of itself, which may or may not be the exact word, therefore generating a potentially incorrect sequence of words.

<sup>2</sup>We used a sentence splitter by Paul Clough, from <http://ir.shef.ac.uk/cloughie/software.html>

<sup>3</sup>The reader may find it amusing that during development the model would often correct “she” into “Rand”, which happened not to be random at all, as it referred to the famous writer Ayn Rand whose article, beginning with the letter A, was in our development data.

<sup>4</sup>The corpus is available by contacting the authors.

## 4 Smoothing Model Parameters

We have found that storing every word encountered during training results in a lexicon of 1.5 million entries and brings huge memory requirements on estimating the model parameters. With the confidence of having observed the infamous “recieve” being corrected more than 4000 times, we opt to map every lexicon entry occurring less than 10 times to a special UNKNOWN token. This not only cuts the lexicon size down to half a million, but also gives us statistics over unseen words, effectively smoothing the model. In an attempt to further smooth our estimates, we linearly interpolate different maximum likelihood estimates as in the following where both sets of  $\lambda$  values sum to 1.

$$P(c_j | c_{j-2}, c_{j-1}) = \lambda_1^1 \cdot P_{ML}(c_j | c_{j-2}, c_{j-1}) \\ + \lambda_2^1 \cdot P_{ML}(c_j | c_{j-1}) \\ + \lambda_3^1 \cdot P_{ML}(c_j)$$

$$P(o_j | c_j) = \lambda_1^2 \cdot P_{ML}(o_j | c_j) + \lambda_2^2 \cdot P_{ML}(c_j)$$

## 5 Decoding

In this setup, decoding is the search for the most likely correct sentence  $C^* = \operatorname{argmax}_C P(C, O)$  for an observed sentence  $O$ . We use the standard Viterbi algorithm to perform this search efficiently. We define a dynamic programming table  $\pi[j][c_{j-1}][c_j]$  which represents the best score for any correct sequence ending at position  $j$  of the observed sentence and in the correct words  $c_{j-1}$  and  $c_j$  where  $c_{j-1}$  belongs in  $\mathcal{C}_{j-1}$ , the set of correct forms seen in data together with  $o_{j-1}$ , similarly for  $c_j$ ,  $\mathcal{C}_j$  and  $o_j$ . We fill this table in  $O(n|\mathcal{C}|^3)$  time, where  $n$  is the length of the observed sentence, and  $|\mathcal{C}|$  is the maximum number of correct forms for any word, which we bound by a constant. We initialize the table with  $\pi[0][\#][\#] = \log 1 = 0$  where  $\#$  is a special symbol used for padding the sentences, and  $\pi[0][.][.] = \log 0 = -\infty$  for any two symbols other than  $\#$ . We use the following recurrence for  $j = 1, 2, \dots, n$ , for all  $c_{j-1} \in \mathcal{C}_{j-1}$ , for all  $c_j \in \mathcal{C}_j$ .

$$\pi[j][c_{j-1}][c_j] = \max_{c_{j-2} \in \mathcal{C}_{j-2}} \pi[j-1][c_{j-2}][c_{j-1}] \\ + \log P(c_j | c_{j-2}, c_{j-1}) \\ + \log P(o_j | c_j)$$

In the end, we use the table and the accompanying back-pointers to obtain the best scoring sequence that yields  $\max_{c_{n-1}, c_n} \pi[n][c_{n-1}][c_n]$ .

## 6 Evaluation and Discussion

We evaluated our system on 13 binary confusion sets, using 11,208 test sentences from the Brown corpus (Kucera and Francis, 1967). Our preliminary results are listed in Table 1. We compare our results with those of Carlson et al. (2001). For their system, we use their notions of *performance*, the percentage of the predictions the system makes that are correct, and *willingness*, the percentage of queries (occurrences of confusion set members) on which the system makes a prediction. For our system, we looked at precision and recall (counting changing the target word as a positive) and noticed that our system has a recall problem rather than a precision problem. Willingness is not an appropriate parameter to tune for our system, since, unlike Carlson et al. (2001), we would like our model to be more aggressive. This is expected, as leaving words intact is the most common pattern in our data. More analysis is required to increase the aggressiveness of the model without lowering our impressive level of precision.

The non-uniformity of our results across confusion sets is also notable. Unlike the compared method, our encoding of the context is simplistic. Therefore our method will err in cases where the correction relies heavily on deep contextual clues as it does for *country-county* more so than it does for *passed-past*.

In making a prediction, we choose a word out of the entire lexicon, not between the two given candidates that constitute a confusion set. We exemplify this characteristic using some examples from our “misclassifications”. For each example, the first sentence is the gold standard, the second is the test sentence, and the third one gives our model’s prediction.

1. (a) They might benefit from *their* treatment there.  
 (b) They might benefit from *there* treatment there.  
 (c) They might benefit from *the* treatment there.
2. (a) However, my *principal* objection in this sort of novel is to the hackneyed treatment of (...)  
 (b) However, my *principal* objection in this sort of novel is to the hackneyed treatment of (...)  
 (c) However, my *main* objection in this sort of novel is to the hackneyed treatment of (...)
3. (a) Living pictures of the early boroughs, *country* life in Tudor and Stuart times, the impact of (...)  
 (b) Living pictures of the early boroughs, *country* life in Tudor and Stuart times, the impact of (...)  
 (c) Living pictures of the early boroughs, *rural* life in Tudor and Stuart times, the impact of (...)

In the following set of examples we demonstrate our system’s behavior on three errors that we found in the text from the paper itself by Carlson et al. (2001) where the first subexample is the erroneous sentence and the second subexample is our output. These examples also show that our capabilities are well beyond the state of the art, in that our model is not limited to confusion sets, or context-sensitive spelling correction for that matter, and apparently it can easily catch errors not visible to the human eye.

1. (a) This work makes *used* of the concept of confusion sets (...)  
 (b) This work makes *use* of the concept of confusion sets (...)
2. (a) All of the experiments were performed using (...) and *a* initial weight of 0.2.  
 (b) All of the experiments were performed using (...) and *an* initial weight of 0.2.
3. (a) Overall effects of eligibility on all our *confusions* sets as well as (...)  
 (b) Overall effects of eligibility on all our *confusion* sets as well as (...)

One limitation of this work as a general copyeditor is that, since the method depends on Wikipedia data for corrections, there is only a finite number of errors that it can correct whereas there exists an infinite number of misspellings for any word. Intuitively, it will be harder to catch a spelling error less typical than “recieve” such as “receive” which

is not a problem at all for a simple spell-checker. Adding to this is the infrequent word mapping. A simple remedy however would be to cascade our system with a traditional spell-checker.

## 7 Future Directions

Our experiments confirmed the intuition that successful text correction can be achieved by better understanding the context, especially now that we have a way of reliably estimating the frequency by which any word in English may be confused with any other. Although in various tasks understanding linguistic context has proven to be difficult, the recent success of discriminative methods gives us an obvious future direction to extend this work. The hope is to carefully design a set of features in a way that encodes the most pertinent information in a context without making strong independence assumptions that hurt prediction while still being computationally feasible. Such a model would not bear the weaknesses of the HMM such as independence between the target word and its surroundings.

Another extension is to introduce some knowledge engineering at the data preprocessing to eliminate training examples that resolve pronouns, replace antonyms, come from flame wars, or correct factual information such as dates and figures.

## Acknowledgments

The author thanks Rani Nelken and Stuart Shieber for providing constant feedback.

## References

- A. J. Carlson, J. Rosen, and D. Roth. 2001. Scaling up context sensitive text correction. In *IAAI2001*, pages 45–50.
- Henry Kucera and W. Nelson Francis. 1967. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI.
- Rani Nelken and Elif Yamangil. 2008. Mining wikipedia’s article revision history for training computational linguistics algorithms. In *Proceedings of the AAAI workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*.
- Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. 2004. Studying cooperation and conflict between authors with *history flow* visualizations. In *CHI*, pages 575–582.

<b>Confusion Set</b>	<b>This Work</b>		<b>Carlson et al. (2001)</b>	
	<b>Precision</b>	<b>Recall</b>	<b>Performance</b>	<b>Willingness</b>
accept-except	1.00	0.34	0.99	0.80
affect-effect	1.00	0.58	0.97	0.75
among-between	0.94	0.21	0.98	0.63
amount-number	0.98	0.23	0.98	0.63
country-county	0.60	0.07	0.98	0.77
fewer-less	0.88	0.25	0.98	0.73
I-me	0.99	0.31	0.99	0.95
passed-past	1.00	0.42	0.99	0.80
peace-piece	0.95	0.35	0.99	0.82
principal-principle	0.91	0.48	0.96	0.51
raise-rise	1.00	0.32	0.98	0.72
than-then	0.98	0.54	0.99	0.88
weather-whether	1.00	0.56	0.99	0.91

Table 1: Results from preliminary evaluation.