A Noisy Channel Model Framework for Grammatical Correction

L. Amber Wilcox-O'Hearn
Department of Computer Science
University of Toronto
amber@cs.toronto.edu

Abstract

We report on the TOR system that participated in the 2013 CoNLL shared task on grammatical correction. The system was a provisional implementation of a beam search correction over a noisy channel model. Although the results on the shared task test set were poor, the approach may still be promising, as there are many aspects of the current implementation that could be optimised. Grammatical correction is inherently difficult both to perform and to evaluate. As such, possible improvements to the evaluation are also discussed.

1 Introduction

Grammatical correction covers many subproblems including spelling correction, lexical choice, and even paraphrasing. There is a sense in which syntax is separable from semantics and discourse. A sentence may be parsable in a language, even if it is nonsensical. On the other hand, many errors that we consider a matter of grammar, such as some instances of determiner choice, are only incorrect because of the semantic or discourse properties of the sentence in its context.

Another complexity is that there are degrees of grammatical correctness. Some sentences are not parsable, but others are just awkward sounding, or unconventional.

So a grammatical error may manifest in a message that doesn't code a meaning in the language at all, and the task becomes inferring a plausible meaning and coding it correctly. This is analogous to non-word spelling errors. Alternatively, it may result in a meaning that is not exactly what was intended. This is more like a real-word spelling error.

In either case, the implication is that in order to detect and correct a grammatical error, we must be able to infer the intended meaning. This points to the depth of the problem.

1.1 Confusion Sets

A common and useful way to construe error correction, including grammatical correction, is to first classify sets of alternatives that are mutually confusable. This is typically done at the lexical level, though the idea is generalizable to multiword expressions, constructions, or phrases. Then the text under examination is searched for instances of members of these confusion sets. Finally a heuristic is used to decide whether one of its alternatives would have been a more appropriate choice in its context. Within this framework, there are different approaches to these steps.

In choosing our confusion sets, we wanted to be flexible and extensible. Therefore, we did not want to depend on corpora of errors with annotated corrections to infer alternatives. So we collected general statistics from corpora that were assumed to be correct, and used those to evaluate proposed corrections to observed sentences. This approach is not unique to this model. It is seen, for example, in (De Felice and Pulman, 2007), (Tetreault and Chodorow, 2008), and (Gamon et al., 2009), among others.

However, the main difference between our system and previous ones is that we do not select our confusion sets in advance of statistical modelling. That is, although the confusion sets we used were based on POS tagsets, there was no classifying or learning to discriminate among members of a confusion set before the task. The aim of this choice was to make our system more general and flexible. We can now modify our confusion sets at runtime without retraining any models. The provisional confusion sets we used are somewhat arbitrary, but this can be changed independently of the rest of the system.

Although our system was not competitive at this

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 109–114,
Sofia, Bulgaria, August 8-9 2013.

stage, it provides a preliminary basis for further experiments.

The remainder of this paper describes the framework and the initial implementation of that framework that was used in the shared task, as well as future improvements to the model. We also discuss the difficulty in evaluating such systems. All of the code used to generate our submission is freely available for examination and use on GitHub (Wilcox-O'Hearn and Wilcox-O'Hearn, 2013).

2 Overview of the system

We approach grammatical error correction using a noisy channel model. Such a model is also used by (Park and Levy, 2011) and (West, Park, and Levy, 2011). One appealing aspect of this model is that it makes explicit the cost of error, such that a correction must not only be more likely than the observation to be proposed, but it must be more likely even given that errors are less likely than non-errors to a degree specified by the properties of the channel. In practice this can mitigate false positives that result from overconfidence in a language model.

A grammatical error is treated as a transformation of some original, correct sentence, S, generated by a language model M. We attempt to recover the original sentence by hypothesizing possible transformations that could have resulted in the observed sentence S'. If we estimate that it is more likely that S was generated by M and transformed into S' than that S' was generated by M and left unchanged, we propose S as a correction.

In this preliminary implementation of the framework, we use a combination of word and POS n-gram models as the language generation model, while POS tags form the basis of our channel model.

To generate sentence hypotheses that can include multiple interacting errors interleaved with non-errors while putting a bound on the size of the search space, we use a left-to-right beam search. This differs from the beam search used by Dalheimer and Ng (2012a). In their work, the search space is constructed by generating variations of the entire sentence. Just as here, at each iteration, they make every variation appropriate at a single position, but they evaluate the whole sentence containing that correction. Although sentences that require multiple interacting corrections will initially have a low score under this method, a large enough beam width will allow the corrections to be made one at a time without being lost from consideration. In our model, by evaluating partial sentences from left-to-right, we hope to lessen the need for a large beam width, by holding off integration of the continuation of the sentence, and letting it unfold in a way that more closely mimics human sentence comprehension.

2.1 The language model

To model language generation, we used an interpolation of two n-gram models, a trigram model based on regular word types, and a 5-gram model of POS tags. The data for these models was derived by combining the corrected version of the NUCLE corpus (Dalheimer, Ng, and Wu, 2013) with a randomly chosen selection of articles from Wikipedia as provided by the Westbury Lab Wikipedia corpus (Shaoul and Westbury, 2010), which we tokenised using NLTK (Bird, Loper, and Klein, 2009) to match the format of the shared task. The precise set of articles used is included in our GutHub repository (Wilcox-O'Hearn and Wilcox-O'Hearn, 2013). We used SRILM 1.7.0 (Stolcke, 2002) to generate a modest trigram model of 5K words. We then passed the same data through the Stanford POS tagger v3.1.4 (Toutanova, Klein, Manning, and Singer, 2003) and again through SRILM to produce a POS 5-gram model.

2.2 The channel model

The channel model provides a definition of transformations that could have been applied to a sentence before we observed it. Our system considers only transformations of single words, specifically, only single word insertions, deletions, and substitutions. This cannot represent every grammar error we might encounter, but makes a good first approximation, and it represents all errors in this iteration of the shared task. To simplify the description and implementation, we equivalently consider the empty string to be a valid word included in some substitution (confusion) sets, and define the channel as one that sometimes replaces a word with one of the alternatives in its confusion set. The probability of such replacement is a parameter α to be inferred.

As explained in the introduction, one goal of our system is to allow flexible confusion sets that do not need to be fully specified in advance of learning statistics about them. Therefore, we define our

confusion sets in terms of the standard POS tagsets as given by the Stanford tagger, using a notion of closed vs. open word classes.

2.2.1 Closed Classes

For our purposes, a closed word class is a set of words that has a relatively small, finite number of members. We composed the following closed classes out of POS tagsets for the purposes of this task:

- DT $\cup \{\epsilon\}$,
- MD $\cup \{\epsilon\}$,
- IN $\cup$ TO $\cup \{\epsilon\}$,
- a hand-built class called AUX, consisting of 'be', 'do', 'have', and 'get' verbs, $\cup$ TO $\cup \{\epsilon\}$.

We then restricted each class to the k most frequently occurring words within it. Our provisional system used $k = 5$.

In the standard tagset, the set TO contains only the word "to". We have put "to" into two different classes, because the same word form represents both the preposition and infinitive verb marker. Although the second such class is labelled "AUX", it does not correspond directly to the standard definition of auxiliary as given by grammars of English. First, "to" does not meet all of the properties of auxiliaries. For example, because it does not occur with a subject, it cannot participate in subject-auxiliary inversion. On the other hand, although modals are traditionally a subclass of auxiliaries, we have left them separate as defined in the tagset.

The intuition guiding those decisions was based on grammatical function and patterns of alternatives. Verb forms in English often consist of a closed class word w, followed by a main verb, the form of which combines with the particular w to indicate the tense and aspect. In other words, w functions as a verb form marker, and doesn't carry other information. Modals, in contrast, have uniform grammatical co-occurrence patterns, essentially all being followed by bare infinitives. They have the semantic function of expressing modality, and are alternatives to one another.

Ultimately, which words are best classed as alternatives should be determined empirically.

2.2.2 Open Classes

We used two open classes specific to this task, verbs and nouns.

The verb errors of this year's task included verb form and subject-verb agreement. Ideally, to find candidates for the confusion set of a verb v, we would want to produce morphological variations of v whose POS tag is different from that of v. This was approximated with the following heuristic. We defined the prefix of v to be the initial characters of v, including least the first character, and not any of the final four, except when the first character was one of those. We collected all words in the vocabulary starting with that prefix, whose stem given by the NLTK Porter stemmer matched the corresponding stem of v and that had appeared at least once with a POS tag indicating a verb of a different form from that of v.

Similarly, the only noun errors under consideration were noun number errors, meaning a change from singular to plural or vice versa. We used the same prefix and stem-matching heuristic as in the verb case to find opposite-numbered nouns for this task.

3 The correction process

In order to detect multiple interacting errors, we would like to consider every possible variation of every word in the sentence. To mitigate the combinatorial expense, we use a beam search as follows.

Proceeding word-by-word through the sentence, we keep a list of the n most likely sentence beginning fragments. Our provisional system used $n = 5$. When we reach the observed word w, then for each sentence fragment s_i in the list, we compute the estimated probability that the correct sentence next contained w' instead of w, using our n-gram probability estimate $P(w'|s_i)$, and that the channel model transformed it to w, by dividing the probability of error α by the number of variations in the confusion set of w', $C(w')$. We also estimate the probability that w was the original word. Because our closed classes each include the empty string, every empty string in the observed sentence could have been produced by the deletion of a member of any of the closed classes. Therefore, we also consider the possibility of inserting each word x, from each closed class. In total, the following probabilities are estimated:

(no error)
$p = P(w|s_i) \times (1 - \alpha)$

and for each word x in each closed class, other than the empty string:

(a deletion, no substitution)
$p = P(xw|s_i) \times \alpha/|C(x)| \times (1 - \alpha)$

and for each variation of w, w':

(a substitution)
$p = P(w'|s_i) \times \alpha/|C(w')|$

and for each variation of w, w', and each word x in each closed class, other than the empty string:

(a deletion and a substitution)
$p = P(xw'|s_i) \times \alpha/|C(x)| \times \alpha/|C(w')|$

The n most likely such extended fragments are then kept for the next iteration. Finally, at the end of the sentence, the sentence with the highest probability is returned as the correction. Probabilities are treated as per-word perplexity in order not to penalise longer sentences.

4 Evaluation

The shared task was evaluated using a section of the NUCLE corpus (see (Dalheimer, Ng, and Wu, 2013)), and the corresponding corrections as annotated by English instructors. The types of corrections ranged from simple and well-defined, such as the addition, removal, or exchange of an article or determiner, to the entire rephrasing of a sentence. Sometimes the corrections were strictly grammatical, in that the original was not well-formed English. Some were more stylistic; what the student had written was awkward, or sounded disfluent, even if it could have been parsed acceptably. This is appropriate and consistent with the nature of the problem. However, it does make evaluation almost as challenging as the task itself.

Often if a sentence has grammatical errors, there are many different ways to repair the error. Teams were encouraged to submit alternative corrections when it was believed that their systems' output ought to be considered valid, even if it did not match the particular annotation given by the grader.

Another problem with the evaluation, however, actually stemmed from the simplification of the task. Because grammatical correction is inherently difficult, and because some of the difficulty increases gradually by type as just described, the task for this year was made more moderate by selecting only 5 error types from the 27 types defined in the corpus. However, this resulted in two difficulties.

The first was that some error types were closely related. Errors of verb form, verb tense, verb modal, and subject-verb agreement may have overlapping interpretation. Those error types are not necessarily distinguishable by our method.

For example, there is a sentence in the test set:

Firstly , security systems are improved in many areas such as school campus or at the workplace .

which is corrected to:

Firstly , security systems have improved in many areas such as school campus or at the workplace .

with the annotation of verb tense error type, and thus not part of this task.

On the other hand, there is also a sentence:

... the electric systems were short circuited...

which is corrected to:

... the electric systems short circuited...

with the annotation of verb form error type, and thus part of this task.

Second, sometimes an annotation not evaluated in this task that resulted in a change of word form was necessarily accompanied by changes to words that were included in the task. This meant that in order for the system to match the gold annotations, it would have to propose a sentence that was grammatically incorrect. This is suboptimal. Although it could sometimes be mitigated by the alternative correction appeal process, that may not have been adequate to address all such occurrences. More accurate scoring might be obtained if only the sentences that do not contain other correction types are included in the test set.

An example of this is the sentence:

Take Singapore for example , these are installed...

The annotation corrects this sentence to:

Take Singapore for example , surveillance is installed...

However, the replacement of *these* with *surveillance* is not in the task, so to get it correct, a system would have to hypothesize:

Take Singapore for example , these is installed...

Evaluation	Prec.	Rec.	F-meas.
Original task	0.1767	0.0481	0.0756
Strict 5 types	0.2079	0.0568	0.0892
With alternatives	0.3067	0.0877	0.1364

Table 1: Results

5 Results

The results of our system were not competitive. Table 1 lists our scores on the original annotation (line 1), and after alternative answers were considered (line 3). It also shows what our system would have scored if only the sentences in the test set which contained no errors types other than those specified for the task were included (line 2).

6 Future Work

There are several simple steps that we expect will improve our system.

First, the language models could be improved. They could use corpora better matched to the data set, and they could have larger vocabulary sizes. We also observe that the POS models, because of their inherently small vocabulary, seem to be impaired by the backoff paradigm. In this case, if a sequence is unattested, it is unlikely that the probability is better estimated by ignoring the beginning of it. Rather, it is likely to indicate an error. Since error detection and correction is precisely what we are attempting, it may be that backoff smoothing is detrimental to the POS models. This hypothesis should be tested empirically.

Second, there are several parameters that could be tuned for better performance, including for example, α, the probability that the channel inserts an error, the beam width n, and the thresholds for the number of alternatives considered in a closed class.

The stemmer we used was not a very sophisticated proxy for morphological analysis, and it made errors in both directions that affected our results.

Finally, there are more classes of error that could be easily included in the sets we have defined. Because they interact, our system may perform better when the allowable transformations are more comprehensive and can complement one another.

Acknowledgments

This work was supported by the assistance of Zooko Wilcox-O'Hearn, who contributed code, analysis, and review, and by Graeme Hirst who provided encouragement and advice.

References

Bird, Steven, Edward Loper and Ewan Klein 2009. *Natural Language Processing with Python.* O'Reilly Media Inc.

Dahlmeier, Daniel, and Hwee Tou Ng. A beam-search decoder for grammatical error correction. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.* Association for Computational Linguistics, 2012.

Daniel Dahlmeier, Hwee Tou Ng 2012. Better Evaluation for Grammatical Error Correction. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2012).* (pp. 568 572). Montreal, Canada.

Daniel Dahlmeier, Hwee Tou Ng, Siew Mei Wu 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. *To appear in Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2013).* Atlanta, Georgia, USA.

De Felice, Rachele, and Stephen G. Pulman. 2007 Automatically acquiring models of preposition use. *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions. Association for Computational Linguistics*, 2007.

Gamon, Michael, et al. Using contextual speller techniques and language modeling for ESL error correction. *Urbana 51* (2009): 61801.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, Joel Tetreault 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. *To appear in Proceedings of the Seventeenth Conference on Computational Natural Language Learning.*

Albert Park and Roger Levy 2011. Automated whole sentence grammar correction using a noisy channel model. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* Volume 1, pp 934-944.

Shaoul, C. and Westbury C. 2010 The Westbury Lab Wikipedia Corpus *Edmonton, AB: University of Alberta* (downloaded from http://www.psych.ualberta.ca/ westburylab/downloads/westburylab.wikicorp.download.html)

A. Stolcke 2002 SRILM – An Extensible Language Modeling Toolkit. *Proc. Intl. Conf. on Spoken Language Processing*, vol. 2, pp. 901-904, Denver.

Tetreault, Joel R., and Martin Chodorow. The ups and downs of preposition error detection in ESL writing. *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL 2003*, pp. 252-259.

West, Randy, Y. Albert Park, and Roger Levy. Bilingual random walk models for automated grammar correction of esl author-produced text. *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, 2011.

L. Amber Wilcox-O'Hearn and Zooko Wilcox-O'Hearn 2013. gc *https://github.com/lamber/gc/tree/d4bc96f03263b8ed00b9629f22dfe0950b37129b*