CS410 Tech Review

Tianfei Chen, Robert Cottrell, Lenny Giannachi, Jingwei Li, Johnathan Montwell

Datasets for Grammatical Error Correction (GEC)

In the ever-growing interconnected world of communication, grammatical nuances are often the hardest and last part an individual will grasp in a new nonnative language. Grammatical correction, which is the focus of this paper, is also one of the more difficult tasks in Natural Language Processing. Below we will explore the data resources needed to design a successful Grammatical Error Correction system, using Neural sequence-to-sequence (seq2seq) models, and evaluate its results.

In order to create a seq2seq model to perform Grammatical Error Correction, we need a large and accurate collection of grammatically corrected text. In the research paper we are attempting to reproduce (*Reaching Human-Level Performance in Automatic Grammatical Error Correction: An Empirical Study*), there are four major corpora that were used; Lang-8 Corpus (Lang-8), Lang-8 extended (Lang-8-Ext), NUS Corpus of Learner English (NUCLE), and Cambridge Learner Corpus (CLC). These corpora contain original text from nonnative speakers and their grammatical corrections by native speakers of English. In addition to training we also use datasets for evaluation, JHU- FLuency-Extended GUG corpus (JFLEG) and Conference on Computational Natural Language Learning in 2014 corpus (CoNLL-2014).

Our goal was to repeat this study with the same datasets as the research technical report. Starting through web search we visited numerous sites including the perceived central location of some of these corpora. Some of these sites appear to be actively maintained. Others required registration before gaining access to any materials. In quite a few of those instances, the registration is never completed. Sites like Lang-8 are actively refusing registration at this time. Another approach was following links in similar projects on GitHub to their data sources. And lastly attempting to reach out by email and phone.

For training, we were able to obtain (Lang-8) Corpus of Learner English and (CLC-FCE) Cambridge Learner Corpus – First Certificate in English, a subset of CLC. Lang-8 as a service is a website where non-native speakers of a language can post text that is then corrected by native speakers. The Lang-8 English Corpus is made up of just over 100k of these entries. CLC-FCE is comprised of 1,244 exam scripts of people obtaining the Cambridge ESOL First Certificate in English and its corrections. Both of these provide a decent amount of grammatical corrections to be used as our training data.

For evaluation, JFLEG and CoNLL-2014 are widely used for benchmarking GEC results. In the Research Technical Report these datasets are used to compare their results against previous state of the art GEC systems. For CoNLL-2014, 25 non-native English speakers from Singapore were given that task to write new essays. The essays were then corrected by two native speakers. JFLEG was created to address the fact that even though corrected sentences may be grammatically correct, they lack the

fluency that a native speaker would produce. JFLEG, by its own admission, sets out to be a gold standard for GEC evaluation.

Obtaining all of the corpora used in the research paper proved difficult. For starters the Cambridge Learner Corpus (CLC) appears to be a commercial learner corpus. Most links to this corpus have moved or now redirect you to sites that don't host it at all. Contacting these sites for more information left us with zero responses. Lang-8 Extended also requires registration for the Lang-8 service which is no longer accepting registrations. NUCLE registration also looks to be largely abandoned, we received no responses for either of these.

Working with Datasets

After locating several of the required datasets, we began to analyze and process the data. From our initial analysis we determined that JFLEG and CoNNL-2014 were the cleanest. This was somewhat expected since JFLEG is widely used and CoNNL-2014 was being contained in a GitHub repo for a Deep Text Corrector. Lang-8 and CLC FCE were not as clean and required a bit of processing. Our GCE model was expecting pairs of grammatically incorrect and correct text data for training. Therefore, we would have to pull that out of the datasets.

As an example, each exam script in CLC FCE was heavily annotated in XML. Below we show an example of what that data looked like and how it was parsed.

Parsing XML Structured Data

CLC FCE exam scripts contain many lines of text that candidates wrote in response to exam questions. Each line is made up of <p> elements that contain grammatically correct text and <NS> elements that identify and correct grammatical errors. <NS> elements are further broken out into <i> and <c> elements to indicate incorrect and correct text respectively. In **Figure 1** we have a line of text that has one grammatical error in it. By parsing this text with a Python XML ElementTree we can easily obtain the incorrect and correct sentences. This leaves us with a pair of sentences that would serve as an ideal entry for training our GCE model.

Figure 1

```
<coded_answer>
    <p>I am so <NS type="RJ"><i>exciting</i><c>excited</c></NS> that I have won the first prize.</p>
</coded_answer>
```

```
Incorrect: I am so exciting that I have won the first prize.
Correct: I am so excited that I have won the first prize.
```

While this may seem straight forward, there were many nuances in the data that prevented this from being a perfect science. Depending on the question the candidate answered, some responses were

in the form of letters making the first and last lines useless for training purposes. In addition, there were sentences that contained many corrections. Although this didn't break our parser, it forced us to continuously revisit how we were parsing the data. As an example, in Figure 2 we have a long sentence that contains multiple corrections. The output from our parser is still what we would expect it to be, but we miss out on correcting the sentence in increments to generate several corrections for training.

Figure 2

```
<coded_answer>
    <p>
        In my <NS type="FN"><i>countries</i><c>country's</c></NS> <NS type="FN"><i>school</i><c>schools</c></NS>
        we are not allowed to have an argument with <NS type="UP"><i>class-mates</i><c>classmates</c></NS> and
        <NS type="MV"><c>are</c></NS> not allowed  to eat <NS type="RT"><i>at</i><c>in</c></NS> class.
    <p>
</coded_answer>
```

```
Incorrect: In my countries school we are not allowed to have an argument with class-mates and not allowed to eat at class
Correct: In my country's schools we are not allowed to have an argument with classmates and are not allowed to eat in class
```

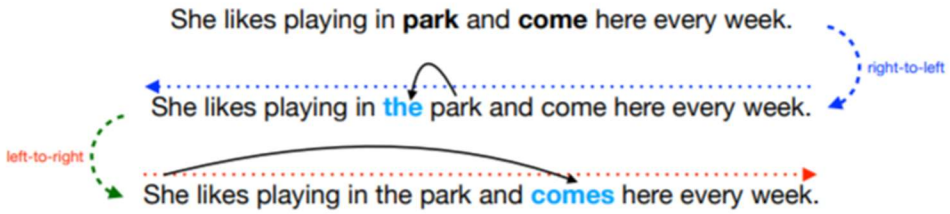
At the time of writing this review we continue to maintain the simplified parser. However, there is clearly an opportunity to continue tweaking the parser and get more usage out of the CLC FCE dataset for training purposes.

Correction Types of CLC FCE

There are 80 different error types annotated in the CLC corpus. The majority of errors fall into subsets of each other; agreement AG, derivation D, form F, inflection I, missing M, replacement R, spelling S, and unnecessary U. The second letter denotes their word class; pronoun A, conjunction C, determiner D, adjective J, noun N, quantifier Q, preposition P, verb V, and adverb Y.

AG Agreement error
AGA Anaphor agreement error
AGD Determiner agreement error
AGN Noun agreement error
AGQ Quantifier agreement error
AGV Verb agreement error
AS Argument structure error
C Countability error
CD Wrong determiner because of noun countability
CE Complex error
CL Collocation or tautology error
CN Countability of noun error
CQ Wrong quantifier because of noun countability
DA Derivation of anaphor error
DC Derivation of link word error
DD Derivation of determiner error
DI Incorrect determiner inflection
DJ Derivation of adjective error
DN Derivation of noun error
DQ Derivation of quantifier error
DT Derivation of preposition error
DV Derivation of verb error
DY Derivation of adverb error
FA Wrong anaphor form
FC Wrong link word form
FD Incorrect determiner form
FJ Wrong adjective form
FN Wrong noun form
FQ Wrong quantifier form
FT Wrong preposition form
FV Wrong verb form
FY Wrong adverb form
IA Incorrect anaphor inflection
ID Idiom wrong
IJ Incorrect adjective inflection
IN Incorrect noun inflection
IQ Incorrect quantifier inflection
IV Incorrect verb inflection
IY Incorrect adverb inflection
L Inappropriate register

M Missing error
MA Missing anaphor
MC Missing link word
MD Missing determiner
MJ Missing adjective
MN Missing noun
MP Missing punctuation
MQ Missing quantifier
MT Missing preposition
MV Missing verb
MY Missing adverb
NE No error
R Replace error
RA Replace anaphor
RC Replace link word
RD Replace determiner
RJ Replace adjective
RN Replace noun
RP Replace punctuation
RQ Replace quantifier
RT Replace preposition
RV Replace verb
RY Replace adverb
S Spelling error
SA Spelling American
SX Spelling confusion
TV Incorrect tense of verb
U Unnecessary error
UA Unnecessary anaphor
UC Unnecessary link word
UD Unnecessary determiner
UJ Unnecessary adjective
UN Unnecessary noun
UP Unnecessary punctuation
UQ Unnecessary quantifier
UT Unnecessary preposition
UV Unnecessary verb
UY Unnecessary adverb
W Word order error
X Incorrect negative formation



Right to Left Sentence Generation

One of the major aspects of this research paper is fluency boosting. The idea is to repeatedly run our sentence through the seq2seq model, improving it iteratively until we see no improvement. Some grammatical corrections are easy to make when moving from left to right, such as a verb agreement error (AGV). Other errors are easier to correct by analyzing the sentence from right to left, such as a missing determiner (MD).

Round-way error correction enhances the iterative approach of fluency boosting by alternating between left-to-right correction and right-to-left correction. This requires two trained seq2seq models trained with sentences from both directions. When we preprocess our dataset, we need to output sentences tokenized both forward and backwards. Example output of the preprocessed data to be fed to our Fairseq seq2seq models is shown below.

```
Incorrect and Corrected Left-to-Right
I am so exciting that I have won the first prize .
I am so excited that I have won the first prize .

Incorrect and Corrected Right-to-Left
. prize first the won have I that exciting so am I
. prize first the won have I that excited so am I
```

Summary

With the Grammatical Error Correction model as our intended application, it was necessary that the data we collected contained grammatically incorrect text and corrections. When this wasn't readily available, we had to pivot and find datasets that were at least annotated for corrections and readable. Although we were not able to retrieve all the datasets from the original study, we were happy with the data we were able to find and will continue to find ways to improve our usage of it.

It is our shared opinion that with the human effort required for generating some of these datasets, they are highly valued by those who created them and are therefore not always readily available to the public. We would hope to see a more open source approach to generating and maintaining this type of data in the future. This is especially true of sites like Lang-8 where datasets are the result of a community effort to help each other learn new languages. This happens organically and at scale, therefore the data should not be as difficult to obtain. We would also like to see updates to the CLC FCE dataset. The dataset is over 15 years old and exam taking is another example of an organic interaction between student and teacher that can generate tons of data with minimal human effort.

Resources

1. Reaching Human-Level Performance in Automatic Grammatical Error Correction: An Empirical Study
https://arxiv.org/pdf/1807.01270.pdf

2. CLC FCE
https://ilexir.co.uk/datasets/index.html

3. Lang-8 Learner Corpora
http://cl.naist.jp/nldata/lang-8/

4. JFLEG (JHU FLuency-Extended GUG) Corpus for Grammatical Error Correction Evaluation
https://github.com/keisks/jfleg

5. Deep Text Corrector
https://github.com/andabi/deep-text-corrector

Special thanks to Ismini Lourentzou and the TAs who guided us throughout the course of this project.