Automated Sentence Correction using Machine Learning

Zhantong Liang

Advisor: Abdou Youssef

Department of Computer Science

School of Engineering & Applied Science

THE GEORGE WASHINGTON UNIVERSITY

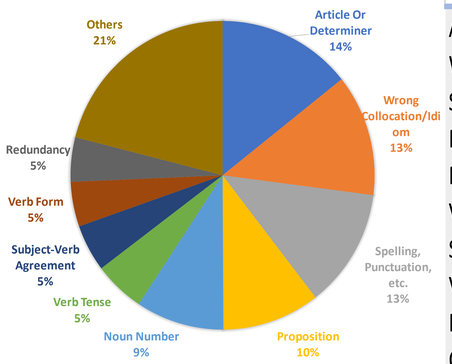
Abstract

- Machine learning can close the gap of misunderstanding between people by automating sentence correction.
- We define categories of errors (aka *noise*) in an English sentence, beyond grammatical errors, such as *synonym errors*, *sentence ordering errors*, etc.
- Methods are proposed to inject targeted errors into sentences for building training sets.
- We will build an error-correction (aka *denoising*) system comprised of two parts: individual error classifiers followed by a number of error correctors (one per error type).

Background

- Recent studies suggest that machine learning sequence-to-sequence (seq2seq) models can outperform grammatically-based error correction.
- Seq2seq models transform an input sentence to another sentence, as in Machine Translation (MT). It's also called an end-to-end approach.
- Unlike in MT where there is plenty of training data (e.g. Chinese-English parallel texts), there is little training data for error correction.
- End-to-end correction systems not only need huge training datasets which are lacking, but also get 'clogged' by mixed errors.
- Grammatical errors are not the only reason to account for confusion.

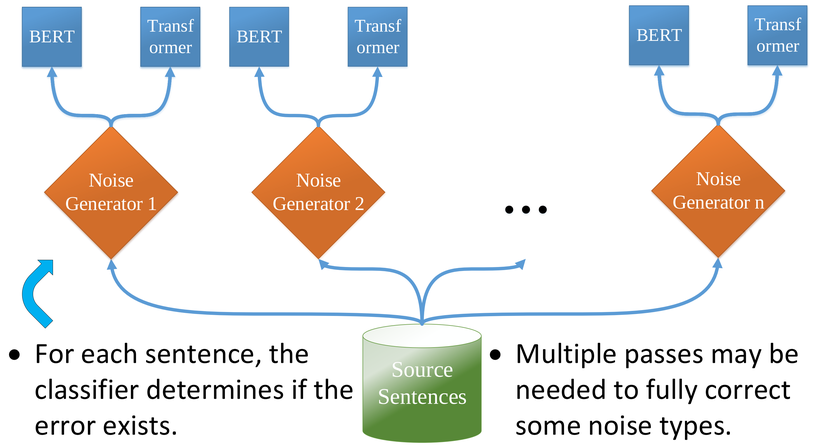
ERROR DISTRIBUTION IN A PUBLIC GEC DATASET



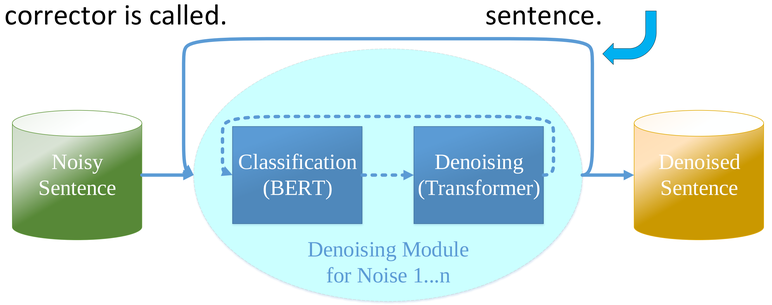
Error Type	%	Recall
Article Or Determiner	14.31%	44.54%
Wrong Collocation/Idiom	12.75%	10.38%
Spelling, Punctuation, etc.	12.47%	45.66%
Proposition	10.38%	49.03%
Noun Number	9.38%	72.65%
Verb Tense	5.41%	28.15%
Subject-Verb Agreement	4.93%	61.79%
Verb Form	4.69%	57.26%
Redundancy	4.65%	25.86%
Others	21.03%	23.28%

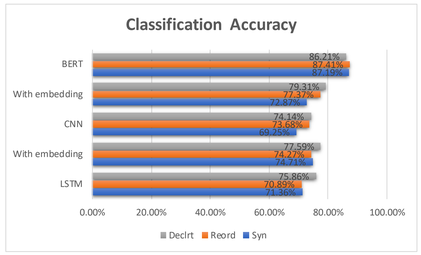
Method & Results

- Build a training dataset generator that generates synonym errors, reordering errors, and question-in-declarative-form errors (see *Examples*).
- Develop an error-factoring approach: Train error classifiers and a number of single-error correctors. The classifier will make use of the state-of-the-art word-embedding system called *BERT*, and the correctors will be based on so-called *Transformers*.

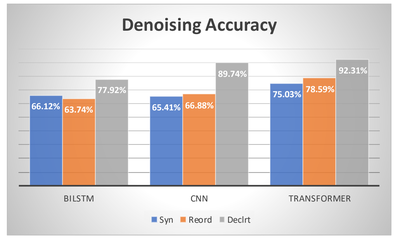


- For each sentence, the classifier determines if the error exists.
- Based on the type, the corresponding error corrector is called.
- Multiple passes may be needed to fully correct some noise types.
- Denoising modules are chained for an input sentence.





- *BERT* is good at attending to both local context and long term dependency.



- Accuracy is not the ideal performance metric: models are better than the metric indicates.

Examples

Adjacent month the student will get his degree?

↓ Correct *synonym noise*

Next month the student will get his degree?

↓ Correct *reordering noise*

The student will get his degree next month?

↓ Correct *reordering noise*

Will the student get his degree next month?

At home the dear manner to produce an apple pie is what?

↓ Correct *synonym noise*

At home the best way to make an apple pie is what?

↓ Correct *reordering noise*

The best way to make an apple pie at home is what?

↓ Correct *reordering noise*

What is the best way to make an apple pie at home?

Conclusion

- We define a novel sentence denoising problem and prove it solvable.
- Our choice of models (*BERT* and *Transformers*) outperforms other neural network architecture.
- Factoring noise makes sense due to the lack of training data and to the difficulty of precise black-box correction.
- In addition to denoising, our system merges synergistically with grammatical error correction (GEC) systems.

Future Work

- Understand the order in which denoising modules are called to optimize outcomes.
- Solve the catastrophic forgetting problem and transfer the knowledge of denoising to existing GEC models.
- Explore attention architectures for denoising problem.
- Automate noise generation with generative models like Generative Adversarial Networks (GANs).