Generating Sequences by Learning to [Self-]Correct
Sean Welleck\textsuperscript{1,2} Ximing Lu\textsuperscript{2} Peter West\textsuperscript{1,2} Faeze Brahman\textsuperscript{1,2} Tianxiao Shen\textsuperscript{1} Daniel Khashabi\textsuperscript{2} Yejin Choi\textsuperscript{1,2}
\textsuperscript{1}University of Washington \hspace{1cm} \textsuperscript{2}Allen Institute for Artificial Intelligence

How do we control and improve a language model’s generations after it is trained?

Key idea: plug in a learned corrector that iteratively improves outputs.

Self-correctors offer several benefits, including:
1. Controlling generators without modifying them
2. Decomposing problems into multiple iterations
3. Using natural language feedback for (1) and (2)

Learning without annotated corrections

Self-correctors improve upon the base generator, and natural language feedback brings additional gains. Diverse tasks: toxicity, lexical constraints, mathematical program synthesis.

Feedback sources:
- Toxicity: fine-grained properties from Perspective API, e.g.
- Lexical constraints: missing words, e.g. add 'bow' and 'prepare'
- Math: few-shot prompted GPT-3, e.g. 2 is missing

Correcting inaccessible larger models & multiple corrections

Discussion
- Natural language feedback: sources (e.g., humans, models) and formats (e.g., line-by-line).
- Other learning algorithms for the corrector: e.g. reinforcement learning

Figure 1. Self-corrective learning iteratively trains a corrector by generating hypotheses and corrections, forming value-improving pairs, and selecting those with high similarity for learning.