Fast Text Generation with Text-Editing Models

KDD 2023

Eric Malmi, Yue Dong, Jonathan Mallinson, Aleksandr Chuklin, Jakub Adamek, Daniil Mirylenka, Felix Stahlberg, Sebastian Krause, Shankar Kumar, Aliaksei Severyn

text-editing-tutorial@google.com

kdd2023-text-editing.github.io

Slides, schedule, etc.

kdd2023-text-editing.github.io

Organizers

Eric Malmi

Yue Dong

Jonathan Mallinson

Aleksandr Chuklin

Jakub Adamek

Daniil Mirylenka

Felix Stahlberg

Sebastian Krause

Shankar Kumar

Aliaksei Severyn

Affiliations

Google Zürich

University of California, Riverside

Google Berlin / NYC

Eric Malmi

Yue Dong

Jonathan Mallinson

Aleksandr Chuklin

Jakub Adamek

Daniil Mirylenka

Felix Stahlberg

Sebastian Krause

Shankar Kumar

Aliaksei Severyn

Presenting today

Eric Malmi

Yue Dong

Jonathan Mallinson

Aleksandr Chuklin

Jakub Adamek

Daniil Mirylenka

Felix Stahlberg

Sebastian Krause

Shankar Kumar

Aliaksei Severyn

Goals

1. Present an overview of the research on Text-Editing models
    a. Focus on general themes rather than individual models
2. Provide practical guidelines for *when* and *how* to apply Text-Editing models
3. Present methods for speeding up LLM inference

Outline

1. **What are text-editing models?** [15 min; Eric]
2. **Model design** [35 min; Eric, Jonathan]
   - Main components of editing models; obtaining target edits
3. **Applications** [35 min; Yue]
   - GEC, Style Transfer, Utterance Rewriting, Simplification

11:25-11:30 Break

4. **Controllable generation** [15 min; Yue]
   - Hallucinations, dataset generation, etc.
5. **Multilingual text editing** [10 min; Eric]
6. **Faster (Large) Language Models** [30min; Jonathan]
7. **Recommendations and future directions** [5 min; Eric]

1. What Are Text-Editing Models?

Presenter: Eric

Text-editing models **generate** natural language by applying **edit operations** to the **input text** to produce the **target text**

Motivation

- Most NLP tasks besides MT are **monolingual**
- Sources and targets often **overlap**
  - Generating the target from scratch is **wasteful**
  - Target can be reconstructed from the source via basic ops like KEEP, DELETE, **INSERT**

Turing	was	born	in	1912	.	Turing	died	in	1954	.
KEEP	KEEP	KEEP	KEEP	KEEP	DEL INS	PRON	KEEP	KEEP	KEEP	KEEP
Turing	was	born	in	1912	and	he	died	in	1954	.

Poll:

How many of you have used
a text-editing model?

Let’s review some Natural Language Generation tasks

Application	**Example** Source (S) and target (T) text	**Use Text Editing?**
Machine translation	S: Turing studied at King's College, where he was awarded first-class honours in mathematics. T: Turing studierte am King's College, wo er erstklassige Auszeichnungen in Mathematik erhielt.	✗

Let’s review some Natural Language Generation tasks

Application	**Example** Source (S) and target (T) text	**Use Text Editing?**
Machine translation	S: Turing studied at King's College, where he was awarded first-class honours in mathematics. T: Turing studierte am King's College, wo er erstklassige Auszeichnungen in Mathematik erhielt.	✕
Summarization	S: Court members Deborah Poritz and Peter Verniero didn’t participate in the Nelson case. T: Two court members didn’t participate in the case.	?

Let’s review some Natural Language Generation tasks

Application	**Example** Source (S) and target (T) text	**Use Text Editing?**
Machine translation	S: Turing studied at King's College, where he was awarded first-class honours in mathematics. T: Turing studierte am King's College, wo er erstklassige Auszeichnungen in Mathematik erhielt.	✗
Summarization	S: Court members Deborah Poritz and Peter Verniero didn’t participate in the Nelson case. T: Two court members didn’t participate in the case.	?
Sentence fusion	S: Turing was born in 1912. Turing died in 1954. T: Turing was born in 1912 and he died in 1954.	✓

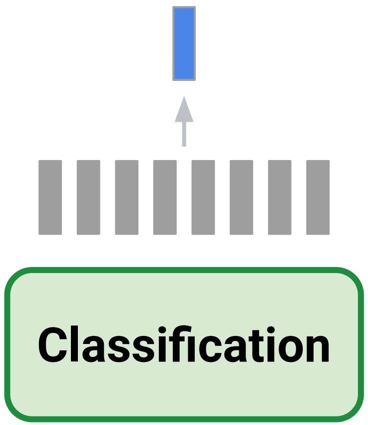
Let’s review some Natural Language Generation tasks

Application	**Example** Source (S) and target (T) text	**Use Text Editing?**
Machine translation	S: Turing studied at King's College, where he was awarded first-class honours in mathematics. T: Turing studierte am King's College, wo er erstklassige Auszeichnungen in Mathematik erhielt.	✗
Summarization	S: Court members Deborah Poritz and Peter Verniero didn’t participate in the Nelson case. T: Two court members didn’t participate in the case.	?
Sentence fusion	S: Turing was born in 1912. Turing died in 1954. T: Turing was born in 1912 and he died in 1954.	✓
Grammar correction	S: New Zealand have a cool weather. T: New Zealand has cool weather.	✓

Applications often studied in the Text-Editing literature

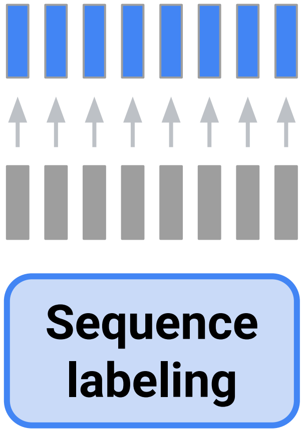
- Grammatical Error Correction (GEC)
- Text Simplification
- Sentence fusion
- Style transfer
- Sentence splitting & rephrasing & fusion
- Text normalization
- Text summarization
- Automatic post-editing for machine translation

NLP tasks map



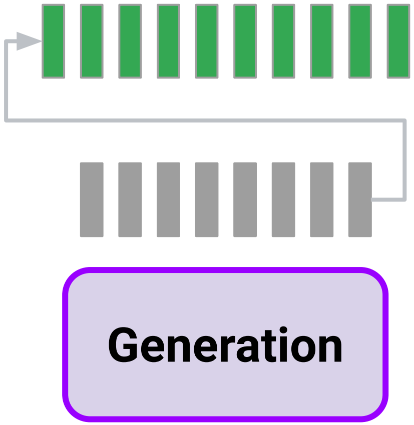
Task
- Single label
- binary, multi-class

Model
- Encoder



Task
- Per token label
- Small softmax

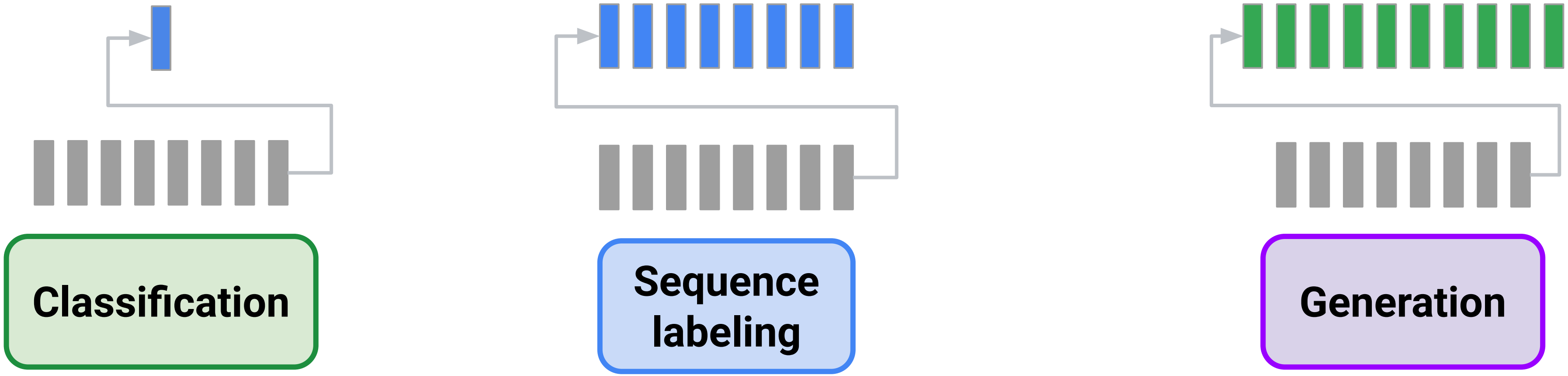
Model
- Encoder



Task
- New sequence
- Large softmax

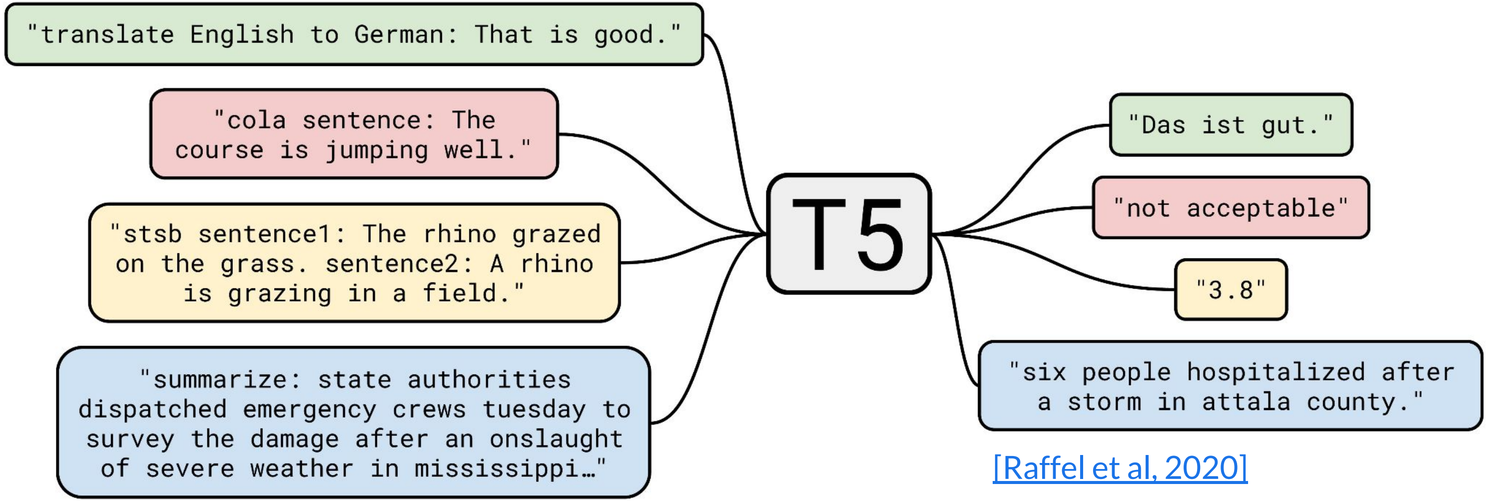
Model
- Encoder + decoder

Generation is all you need?



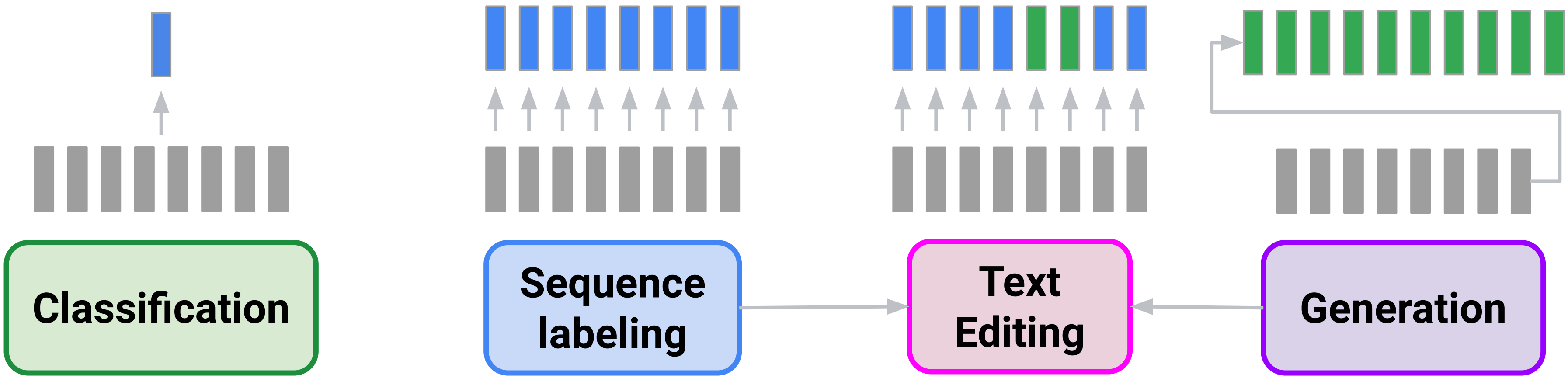
- Autoregressive LMs (Generation models) can also generate classification labels and sequence labels

LLMs like T5 and GPT excel across various NLP tasks



[Raffel et al, 2020]

Where does Text Editing fit?



Task
- Single label (binary, multi-class)

Model
- Encoder + softmax

Latency
- Fast (feed-forward)

Task
- Per token label
- Small softmax

Model
- Encoder + softmax

Latency
- Fast (feed-forward)

Task
- Tagging + Insertion
- small/large softmax

Model
- Encoder + decoder

Latency
- Fast

Task
- New sequence
- Large softmax

Model
- Encoder + decoder

Latency
- Slow (autoregressive)

Text-Editing models leverage inductive bias (high overlap) to:

1. Make **inference** faster without compromising the quality
2. Simplify the task (smaller output space) to make models more **data efficient**

Text Editing Advantages

Data efficient	Text Editing models need less training data.
Latency	Can be >10x faster inference.
Faithfulness	Constraining decoders in seq2seq is an active area of research
Control	We can control the word a model can add / remove. Can incorporate external knowledge (e.g., pronoun).

Are Text-Editing models relevant in the LLM era?

- **IF** you:

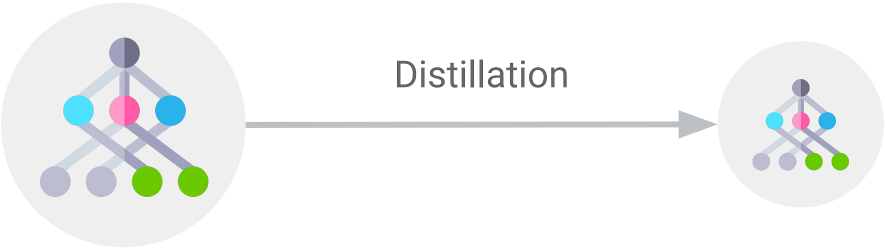
  (1) only care about **quality / generalization**,

  (2) don't have latency, cost, or infra constraints, and

  (3) don't need fine-tuning,

  the answer is: ***maybe not***
- But that's a big **IF**!
- LLMs and Text Editing can nicely complement each other via **distillation** [Hinton et al. 2015]

Distilling LLMs into Text-Editing models



LLM

Text-Editing model

1. Take a sample of model inputs
2. Generate target outputs with an LLM
3. Train a Text-Editing model on this data and serve it

⟶ may allow combining the quality of LLM and the advantages of Text Editing

Questions?