Copy That! Editing Sequences by Copying Spans

Sheena Panthaplackel[*1], **Miltiadis Allamanis**[2], **Marc Brockschmidt**[2]
[1]The University of Texas at Austin [2]Microsoft Research Cambridge, UK
*Work done during internship at Microsoft Research Cambridge, UK

Motivation

Sequence-to-sequence models are widely used for *editing* tasks in which most of the input text is preserved. Prevailing token-level copying techniques require the decoder to make several predictions to reproduce long spans of the input.

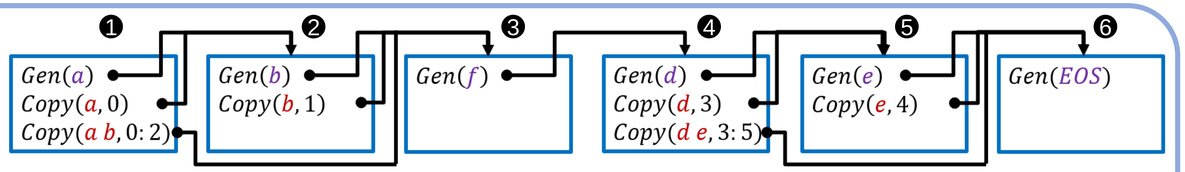
Editing natural language (e.g., Grammatical error correction):
Its there decision to make. → It's their decision to make.

Editing source code (e.g., Bug fixing):
`while(i <= arr.Length)` → `while(i <= arr.Length-1)`

Goal: Learn to copy long spans of the input.

Model Overview

Correct actions $\alpha \in A_k$ at k^{th} step
Kinds of actions (α):
- Gen(token) from vocabulary
- $Copy$(span of tokens, pos range) from input



Training

Marginalize over all possible action sequences leading to correct output sequence.

- Maximize $p(o_0 \dots o_m \mid \boldsymbol{in})$ defined recursively:

$$p(o_k \dots o_m \mid \boldsymbol{in}, o_0 \dots o_{k-1}) = \sum_{\substack{\alpha \in A_k \\ l = [\![\alpha]\!]}} q(\alpha \mid \boldsymbol{in}, o_0 \dots o_{k-1}) \times p(o_{k+l} \dots o_m \mid o_0 \dots o_{k+l-1})$$

- Probability of generating correct suffix conditioned only on subsequence generated so far, not the concrete actions
- Encourages copying longer spans through fewer # of actions

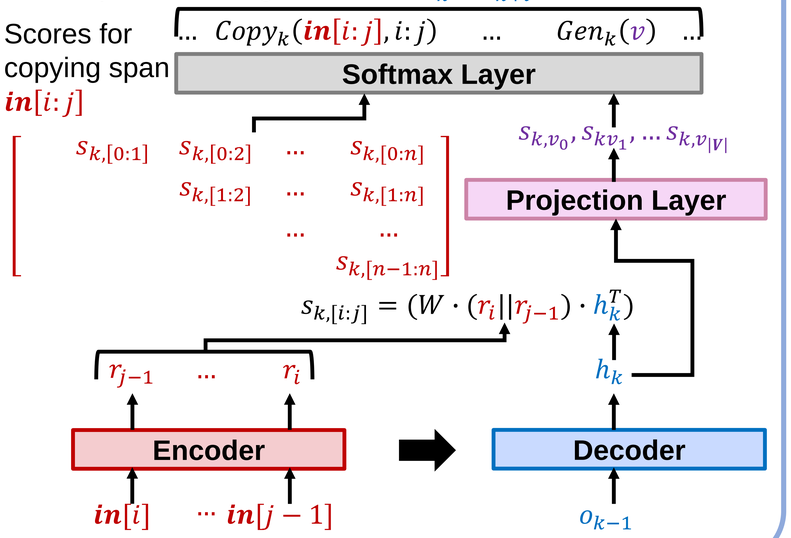
Inference

Generate likely action sequences through a beam search variant that is more compatible with the training objective.

- Pause expansion when token sequence length exceeds action sequence length
- Merge rays yielding identical token sequences

Action α yields a token sequence of length l $\quad \alpha \rightarrow o_k \dots o_{k+l}$

Scores for copying span $\boldsymbol{in}[i:j]$



Evaluation Task #1: Bug Fixing

Given faulty code, generate the corrected version.

Table 1: Accuracy on BFP Dataset *(Tufano et al., 2019)*

	BFP_{small}		BFP_{medium}	
	@1	@20	@1	@20
Tufano et al. (2019)	9.2	43.5	3.2	22.2
Seq2Seq + Token Copying	14.8	42.0	7.0	23.8
Seq2Seq + Span Copying	**17.7**	**45.0**	**8.0**	**25.4**
Always Copy Longest Span	14.2	33.7	7.2	20.0
No Marginalization	16.9	43.4	2.5	11.1

Heuristic $Copy$ action selection fails to capture entire spectrum of correct actions.

Equipping Seq2Seq w/ span copying achieves state-of-the-art on BFP datasets.

Marginalization incentivizes model to use as few actions as possible.

Table 2: Statistics on Lengths of Copied Spans

	BFP_{small}	BFP_{medium}
μ	9.6	29.5
Median	7.0	19.0
% of $Copy$ actions yielding spans of length > 1	88.8	72.9

$Copy$ actions tend to yield long copy spans.

Evaluation Task #2: Learning Edit Representations

Learn to embed similar edit patterns nearby in a vector space.

Span copying leads to more accurate predictions of edited text and code.

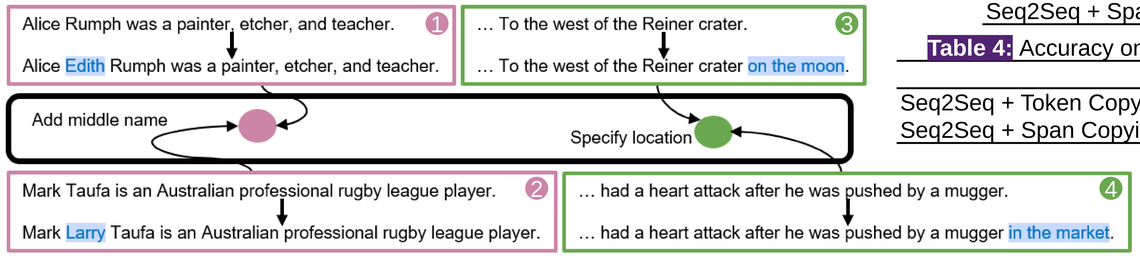


Table 3: Accuracy on Predicting Edits

	WikiAtomicEdits	GitHubEdits
Yin et al. (2019)	72.9	59.6
Seq2Seq + Token Copying	67.8	64.4
Seq2Seq + Span Copying	**78.1**	**67.4**

Table 4: Accuracy on One-Shot Transfer Task

	C# Fixers
Seq2Seq + Token Copying	18.8
Seq2Seq + Span Copying	**24.2**

Span copying facilitates learning more generalizable edit representations.