Simple and Effective
Retrieve-Edit-Rerank Text Generation

Nabil Hossain
University of Rochester
nhossain@cs.rochester.edu

Marjan Ghazvininejad
Facebook AI Research

Luke Zettlemoyer
Facebook AI Research
Overview

• Retrieve-and-edit
  • Generate text using retrieved examples from training set
  • Uses: Summarization, Machine Translation, Conversation Generation

• We apply post-generation ranking
  • Retrieve N examples, generate a candidate output with each
  • Rank these candidates

• Post-ranking improves results on:
  • 2 Machine Translation tasks
  • Gigaword Summarization task
Retrieve (Gigaword)

- 1st sentence of news article (x) -> title (y)
- **Retrieval**: given x, find closest x', then obtain its title y'
  - LUCENE (TF-IDF based)

**Examples:**

| Article (x)                                                                 | Best retrieved (y')                                                                 | Title (y)                                                                |
|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| factory orders for manufactured goods rose #.# percent in september, the commerce department said here thursday. | u.s. factory orders rises #.# percent in october                                  | us september factory orders up #.# percent                               |
| france, still high after their convincing ###-## win over new zealand have named the same team for the second test next saturday in paris. | france poised to make history in #nd test                                          | french keep same team for #nd test                                       |
Edit (Generate)

- For each augmented input $x [\text{SEP}] y'_i$, generate $\hat{y}_i$
factory orders for manufactured goods rose #.# percent in September, the commerce department said here Thursday.

u.s. factory orders rises #.# percent in *October*

us *September* factory orders up #.# percent
Post-gen Rerank

Given: \( \{(x \ [\text{SEP}] y'_1; \hat{y}_1), \ldots, (x \ [\text{SEP}] y'_N; \hat{y}_N)\} \)

Estimate: \( \hat{y}^* = \arg \max_{\hat{y}_i} \text{similarity}(\hat{y}_i, y), 1 \leq i \leq N \)
Post-gen Rerank

**Training Set**

$(x', y')$ → **Module 1: Retrieve**

$\{ y_1', y_2', y_3' \}$

**Test Data**

$x$ → Augmented Input

| Article ($x$) | Best retrieved ($y'$) | Title ($y$) |
|---------------|------------------------|-------------|
| factory orders for manufactured goods rose #.# percent in september, the commerce department said here thursday. | u.s. factory orders rises #.# percent in october | us september factory orders up #.# percent |

**Module 2: Generate**

$x$ [SEP] $y_1'$ → $\hat{y}_1$

$x$ [SEP] $y_2'$ → $\hat{y}_2$

$x$ [SEP] $y_3'$ → $\hat{y}_3$

**Module 3: Post-Gen Rerank**

$\hat{y}_1$ → Candidate Outputs

$\hat{y}_2$ → $\hat{y}_3$

$\hat{y}_3$ → $\hat{y}_1$

$\hat{y}_1$

factory orders rises #.# percent in september

$\hat{y}_2$

us september factory orders rose #.# percent

$\hat{y}_3$

factory orders for good rose #.# percent in september
Model

- BPE
- Transformer base
- Segment Embeddings
- A [RANK] token similar to [CLS] token in BERT
  - to estimate salience of the retrieved $y'$
- Generate with beam = 5
Machine Translation

- **Data**: EN-NL (Dutch) and EN-HU (Hungarian), from EU meetings
- **SOTA is NFR**: Retrieval-based LSTM model
  - Uses SetSimilaritySearch for retrieval (retrieves top 3)

| System                                | BLEU  |
|---------------------------------------|-------|
|                                       | EN-NL | EN-HU |
| LSTM                                  | 51.45 | 40.47 |
| NFR                                   | 58.91 | 48.24 |
| Transformer (Tr)                      | 59.88 | 49.61 |
| Tr + sss+ed (NFR equivalent)          | 62.86 | 52.74 |
| Tr + Lucene + x [SEP] $y'_1$          | 64.92 | 55.16 |
Machine Translation

- **Data:** EN-NL (Dutch) and EN-HU (Hungarian), from EU meetings
- Current SOTA is NFR: Retrieval-based LSTM model
  - Uses SetSimilaritySearch for retrieval (retrieves top 3)
- Our ranker: Select highest scored output from the trained MT model

| System                                      | EN-NL | EN-HU |
|---------------------------------------------|-------|-------|
| LSTM                                        | 51.45 | 40.47 |
| NFR                                         | 58.91 | 48.24 |
| Transformer (Tr)                            | 59.88 | 49.61 |
| Tr + sss+ed (NFR equivalent)                | 62.86 | 52.74 |
| Tr + Lucene + x [SEP] y'_1                  | 64.92 | 55.16 |
| Tr + Lucene + pre-rank                      | 65.20 | 55.36 |
| Tr + Lucene + post-rank (ours)              | **65.43** | **55.73** |

- Post-generation ranking amounting to extended beam search

*Bulté, Bram, and Arda Tezcan. "Neural Fuzzy Repair: Integrating Fuzzy Matches into Neural Machine Translation." In ACL 2019.*
Gigaword Summarization

- Metric: Rouge F-scores
- Re³Sum model: LSTM, retrieve-and-edit, pre-ranking
  - uses 30 retrieved examples
- Our ranker: select the most frequent of the 30 candidate outputs

| Method                   | Rouge-1 | Rouge-2 | Rouge-LCS |
|--------------------------|---------|---------|-----------|
| LSTM                     | 35.01   | 16.55   | 32.42     |
| Re³Sum                   | 37.04   | 19.03   | 34.46     |
| Transformer (Tr)         | 37.68   | 18.79   | 34.87     |
| Tr + Lucene + X [SEP]y′ | 37.51   | 19.15   | 34.86     |
| Tr + Lucene + pre-rank   | 36.46   | 18.01   | 33.85     |
| Tr + Luc + post-rank     | **38.23** | **19.58** | **35.60** |
Gigaword oracle experiments

- Room for improvement with better post-ranking
Room for improvement with better post-ranking

- use $x$, $x'$, $y'$, $\hat{y}$ in post-ranking

\[
\{(x \text{ [SEP]} y'_1; \hat{y}_1), \ldots, (x \text{ [SEP]} y'_N; \hat{y}_N)\}
\]

\[
\hat{y}^* = \arg \max_{\hat{y}_i} \text{similarity}(\hat{y}_i, y), 1 \leq i \leq N
\]
We extended the retrieve-and-edit framework with post-generation ranking:

1. Retrieve N training set outputs $y'$ for input $x$
2. Edit each input $x[SEP]y'$ to produce N candidate outputs $\hat{y}$.
3. Re-rank $\hat{y}$ to select best ranked output

- Simple post-ranking improved results on MT and summarization
- Interesting to explore better post-ranking using $x$, $x'$, $y'$, yhat

Questions: nhossain@cs.rochester.edu