UIT

University of Information Technology

MemSum: Extractive Summarization of Long Documents Using Multi-Step Episodic Markov Decision Processes

Le Nhat Minh - 21522339 Tran Duy Tan - 21522576
Nguyen Duy Thai - 21522581

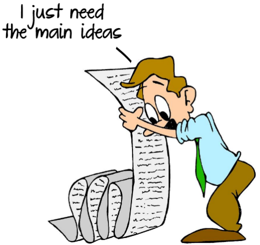
December 16, 2023

Outline:

Section 1: Introduction

Introduction

In the era of abundant online data, particularly unstructured textual information, there has been a significant increase in the demand for automatic text summarization tools. The adoption of these tools is crucial because they excel at extracting pertinent information while eliminating unnecessary details, enabling more data to fit within a confined space.



Section 1: Introduction

Introduction

There are 2 types of text summaries:

- Extractive summarization
- Abstractive summarization

Section 2: Method

- 2.1: Overview Method
- 2.2: Detail Method

2.1: Overview

Components

- Policy Gradient
- Multi-step Episodic MDP Policy
- Policy Network

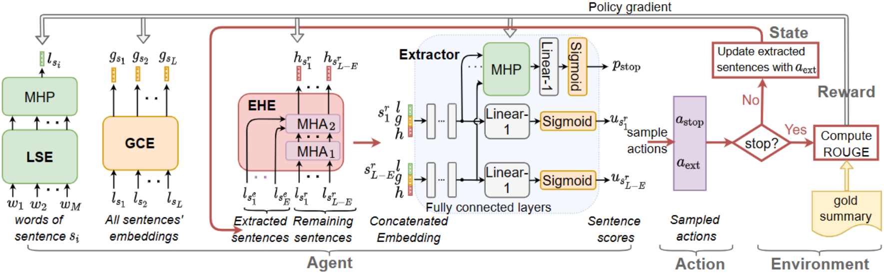


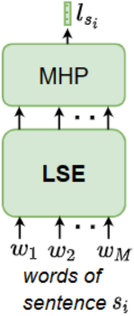
Figure: Architecture

2.2: Detail Method

2.2.1: Local Sentence Encoder (LSE)

LSE

The local content of the sentence.
Input: Ordered words $(w_1, w_2, \ldots, w_M)$ in a sentence s_i.
Output: Sentence embeddings l_{s_i}.

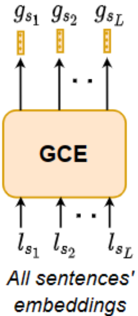


2.2: Detail Method

2.2.2: Global Context Encoder (GCE)

GCE

The global context of the sentence within the document.
Input: The L local sentence embeddings $(l_{s_1}, l_{s_2}, \dots l_{s_L})$
Output: An embedding g_{s_i} that encodes global contextual information such as the sentence's position in the document and information on neighboring sentences.

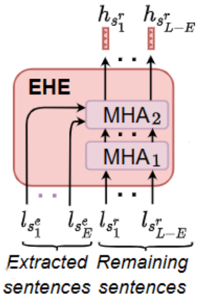


2.2: Detail Method

2.2.3: Extraction History Encoder (EHE)

EHE

Encodes the extraction history information and produces the extraction history embedding $h_{s_i^r}$ for each remaining sentence s_i^r.

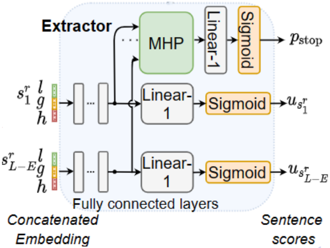


2.2: Detail Method

2.2.4: Extractor

Extractor

Computes the score of each remaining sentence and outputs an extraction stop signal.
Input: Each of the remaining sentences s_i^r an aggregated embedding by concatenating $l_{s_i^r}$, $g_{s_i^r}$ and $h_{s_i^r}$.
Output: A stopping probability p_{stop} and the score of each remaining sentence $(u_{s_1^r}, u_{s_2^r}, \ldots, u_{s_{L-E}^r})$.



2.2: Detail Method

2.2.5: Training

Training

Using Reinforcement learning:

- Objective function: $J(\theta) = \mathbb{E}_{\pi_\theta}[R_0]$
- The reward r is usually expressed:

$$r = \frac{1}{3}(ROUGE - 1_f + ROUGE - 2_f + ROUGE - L_f) \quad (1)$$

- The policy gradient is defined:

$$\nabla J(\theta) = \mathbb{E}_\pi[R_t \nabla log\pi(A_t|S_t, \theta)] \quad (2)$$

- Update rule:

$$\theta_{t+1} = \theta_t + \alpha R_t \nabla log\pi(A_t|S_t, \theta) \quad (3)$$

2.2: Detail Method

2.2.5: Training

Training

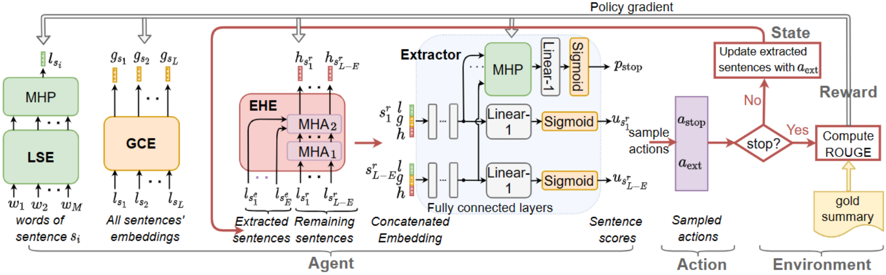
- The agent's policy is:

$$\pi(A_t|S_t,\theta_t) = p(stop|S_t,\theta_t) \cdot p(a_t|stop,S_t,\theta_t) \tag{4}$$

$$p(a_t|stop,S_t,\theta_t) = \begin{cases} \dfrac{u_{a_t}(S_t,\theta_t)}{\sum_{j\in I_t} u_j(S_t,\theta_t)} & \text{if } stop = \text{false} \\ \\ \dfrac{1}{|I_t|} & \text{if } stop = \text{true} \end{cases}$$

2.2: Detail Method

2.2.5: Training



Algorithm 1 The training algorithm.

Parameters: learning rate α

1: **for** each document-summary pair (D_i, G_i) **do**
2: LSE outputs local sent. embed $l_{s_1},\dots,l_{s_L}$
3: GCE outputs global context embed $g_{s_1},\dots,g_{s_L}$
4: Sample an episode $S_0,s_{a_0},\dots,S_{T-1},s_{a_{T-1}},S_T,A_{\text{stop}},r$ from the high-ROUGE episodes set $\mathbb{E}_p$ of document D_i
5: **for** each time step: $t = 0,1,\dots,T$: **do**
6: **if** $t > 0$ **then**
7: EHE outputs extraction history embed $h_{s_1^r},\dots,h_{s_{L-E_t}^r}$ for remaining sentences
8: **else**
9: Initialize $h_{s_1^r},\dots,h_{s_{L-E_0}^r}$ to $\mathbf{0}$
10: Extractor outputs scores $u_{s_1^r},\dots,u_{s_{L-E_t}^r}$ for remaining sentences and outputs p_{stop}
11: Compute the action probability $\pi(A_t|S_t,\boldsymbol{\theta})$ according to Equation (4)
12: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\frac{r}{T+1}\nabla \log \pi(A_t|S_t,\boldsymbol{\theta})$

3.Dataset

- PubMed: Includes 133215 scientific papers.
  - Train : 119924 papers.
  - Val: 6633 papers.
  - Test: 6658 papers.
  - Reference: www.tensorflow.org/datasets/catalog/scientific_papers
- GovReport: Includes 7238 government reports.
  - Train: 6514 reports.
  - Val: 362 reports.
  - Test: 362 reports.
  - Reference: www.tensorflow.org/datasets/catalog/gov_report
- ArXiv: Includes 215,913 scientific papers.
  - Train: 203037 papers.
  - Val: 6436 papers.
  - Test: 6440 papers.
  - Reference: www.tensorflow.org/datasets/catalog/scientific_papers

3.Dataset

- PubMed$_{trunc}$: Includes 92934 scientific papers.
  - Train: 83233 papers.
  - Val: 4676 papers.
  - Test: 5025 papers.
  - Reference: `https://huggingface.co/datasets/nianlong/long-doc-extractive-summarization-truncated-pubmed`
- Custom Dataset: Includes 300 instances.
  - Train: 100 instances.
  - Val: Train: 100 instances.
  - Test: Train: 100 instances.
  - Reference: `https://github.com/nianlonggu/MemSum`

Section 4: Experiments

4.1: Metric

Metric

Using F1 ROUGE including ROUGE-1,2, and L for measuring unigram, bigram, and longest common subsequence.

- ROUGE-1: refers to the overlap of unigrams (each word) between the system and reference summaries.
- ROUGE-2 refers to the overlap of bigrams between the system and reference summaries.
- ROUGE-L: Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence-level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.

Section 4: Experiments

4.2: Result

Dataset	PubMed			ArXiv			PubMedtrunc			GovReport			Custom Dataset		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Validation	46.91	21.64	42.94	46.72	19.28	41.11	43.05	16.59	38.38	58.47	27.98	55.64	37.34	12.79	33.59
Test	47.13	21.62	42.61	47.11	19.45	41.33	42.71	16.53	38.02	58.89	28.01	56.08	39	13.72	35.11

Dataset	PubMed			ArXiv			PubMedtrunc			GovReport		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
R-1	R-2	R-L										
Validation	49.14	22.92	44.33	48.23	20.17	42.31	43.46	16.77	38.65	59.29	28.57	56.46
Test	49.25	22.94	44.61	47.11	19.45	41.33	42.71	16.53	38.02	58.89	28.01	56.08
39	13.72	35.11										

5: Advantages and Disadvantages

Advantages and Disadvantages

Advantages :

- Non-redundant Summaries.
- Efficient Attention.

Disadvantages

- Not optimize the order of the extracted sentences.
- Training Complexity.