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

arXiv preprint: 2107.08929

Overview

This work focused on the extractive summarisation of long documents on the sentence-level. The proposed model, MemSum, iteratively selects sentences based on (1) their content, (2) the context provided by the other sentences in the document, and (3) the sentences that have already been selected.

Contributions

MemSum outperforms both extractive and abstractive summarisation models across multiple datasets. This is despite being a lightweight model that contains just 4.4 million trainable parameters – in comparison, extractive summarization models that rely on language models like BERT can contain >100 million parameters.

Methodology

MemSum is trained using **policy gradient reinforcement learning** that optimizes ROUGE F1 against the gold summary.

- At each time step t, the agent either stops extraction or continues by selecting from the remaining sentences.
- When the agent stops extraction, the reward *r* for the episode is computed. This *r* is used to update the policy parameters at each time step *t*.

The policy network for MemSum has three main components that cater to the three sentence-selection criteria highlighted above.

- **Local Sentence Encoder (LSE)**: a multi-layer bi-directional LSTM that computes a representation of each sentence's content (based on the words in the sentence).
- **Global Context Encoder (GCE)**: another multi-layer bi-drectional LSTM that computes a representation of how each sentence relates to the other sentences in the document.
- **Extraction History Encoder (EHE)**: a multi-layer network, where each layer has two multi-head attention sublayers. The EHE produces a representation of the extraction history for each remaining sentence, based on the context of the sentences extracted thus far and the context of the remaining sentences.
- The **Extractor** computes a score for each remaining sentence, based on that sentence's content, the global context embedding for that sentence, and that sentence's extraction history. These scores are used to compute the stopping probability p_{stop} and to sample the next sentence for extraction.

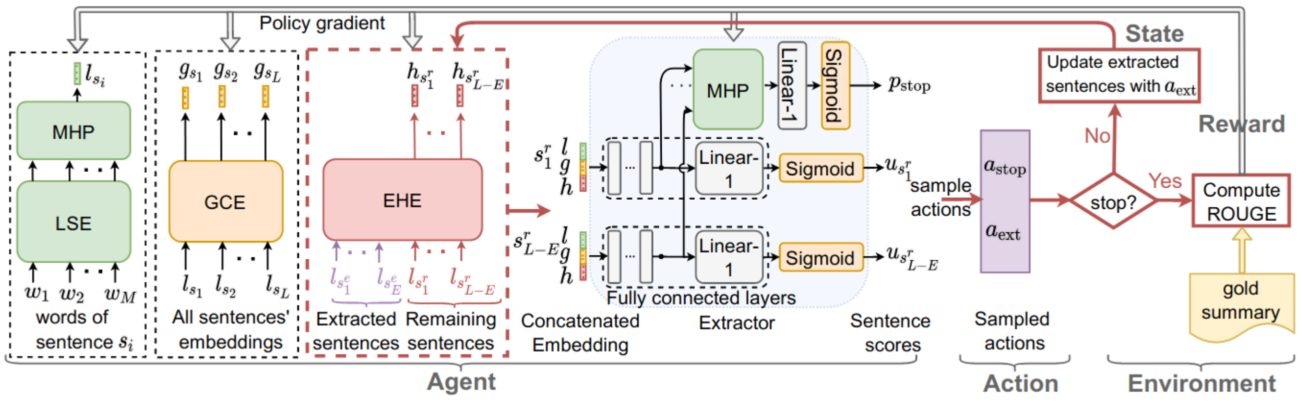


Figure 1: The architecture of our MemSum extractive summarizer with a multi-step episodic MDP policy. Because the extraction-history embeddings h are updated at each time step t, the scores u of remaining sentences and the stopping probability p_{stop} are also updated.

Experiments and Results

MemSum is evaluated via **ROUGE F1** on the PubMed, arXiv, and GovReport datasets.

Datasets	**avg. doc. length**		**avg. summ. length**		**# of doc.-summ. pairs**		
	# of words	# of sentences	# of words	# of sentences	Train	Valid	Test
PubMed	3016	88	203	7	116,937	6,633	6,657
arXiv	4938	206	220	10	202,880	6,436	6,440
GovReport	9409	307	553	18	17,517	974	973

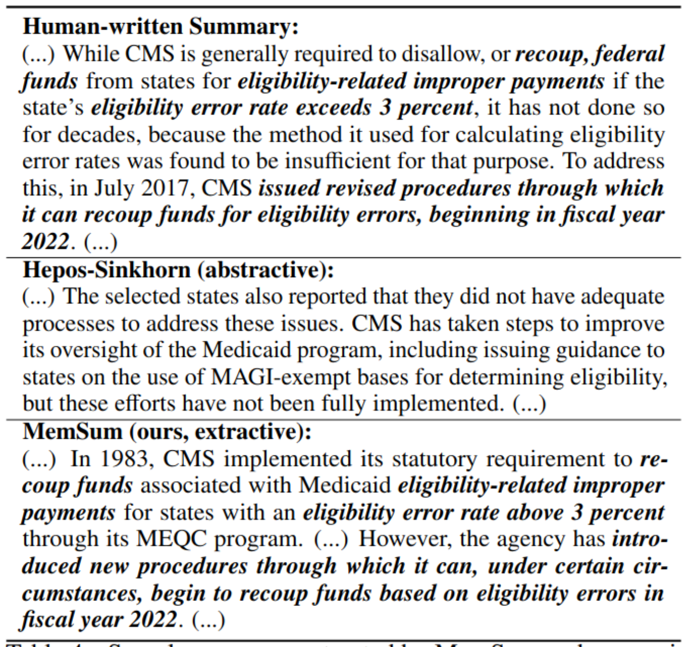
Table 1: An overview of datasets used in this paper.

Model	**PubMed**			**arXiv**		
	R-1	R-2	R-L	R-1	R-2	R-L
ORACLE	61.99	34.95	56.76	60.00	30.60	53.03
Extractive summarization models						
SumBasic	37.15	11.36	33.43	29.47	6.95	26.3
Lead-10	37.45	14.19	34.07	35.52	10.33	31.44
SummaRuNNer	43.89	18.78	30.36	42.81	16.52	28.23
Atten-Cont	44.85	19.70	31.43	43.62	17.36	29.14
Sent-CLF	45.01	19.91	41.16	34.01	8.71	30.41
Sent-PTR	43.30	17.92	39.47	42.32	15.63	38.06
Abstractive summarization models						
PEGASUS	45.97	20.15	41.34	44.21	16.95	38.83
BigBird	46.32	20.65	42.33	46.63	19.02	41.77
Dancer	46.34	19.97	42.42	45.01	17.60	40.56
Hepos-Sinkhorn	47.93	20.74	42.58	47.87	20.00	41.50
Hepos-LSH	48.12	21.06	42.72	48.24	20.26	41.78
Our Models						
Lead-10 (ours)	37.71	14.13	34.30	34.88	10.46	30.96
MemSum (ours)	**49.25**	**22.94**	**44.42**	**48.42**	**20.30**	**42.54**

Table 2: Results on the PubMed and arXiv test sets.

Model	**R-1**	**R-2**	**R-L**
ORACLE	75.56	45.91	72.51
Abstractive summarization baselines			
Hepos-LSH	55.00	21.13	51.67
Hepos-Sinkhorn	56.86	22.62	53.82
Our Models			
Lead-20 (ours)	50.94	19.53	48.45
MemSum (ours)	**59.43**	**28.60**	**56.69**

Table 3: Results on the GovReport test set.



Human-written Summary:
(...) While CMS is generally required to disallow, or ***recoup, federal funds*** from states for ***eligibility-related improper payments*** if the state's ***eligibility error rate exceeds 3 percent***, it has not done so for decades, because the method it used for calculating eligibility error rates was found to be insufficient for that purpose. To address this, in July 2017, CMS ***issued revised procedures through which it can recoup funds for eligibility errors, beginning in fiscal year 2022***. (...)

Hepos-Sinkhorn (abstractive):
(...) The selected states also reported that they did not have adequate processes to address these issues. CMS has taken steps to improve its oversight of the Medicaid program, including issuing guidance to states on the use of MAGI-exempt bases for determining eligibility, but these efforts have not been fully implemented. (...)

MemSum (ours, extractive):
(...) In 1983, CMS implemented its statutory requirement to ***recoup funds*** associated with Medicaid ***eligibility-related improper payments*** for states with an ***eligibility error rate above 3 percent*** through its MEQC program. (...) However, the agency has ***introduced new procedures through which it can, under certain circumstances, begin to recoup funds based on eligibility errors in fiscal year 2022***. (...)

Table 4: Sample summary extracted by MemSum and comparison with the abstractive summary generated by Hepos-Sinkhorn [20].

Analysis

The authors noted the following:

- The performance of MemSum suffers when the EHE component is removed.
- Letting MemSum stop early (instead of after some fixed number of extraction steps) produces summaries with higher ROUGE F1 scores and a lower number of sentences.
- The EHE helps MemSum avoid sentences that are similar to sentences that had already been selected, thereby reducing redundancy in output summaries.
  - Trigram blocking (skipping sentences that have trigram overlap with current summary) can help MemSum avoid duplicates as well; however, the summaries produced have lower ROUGE F1 scores.
- An alternative to choosing a stopping probability threshold is to insert a special STOP sentence into every document and to terminate when the STOP sentence is selected. However, this alternative leads to summaries with fewer sentences and lower ROUGE F1 scores.