SummaRerankér: A Multi-Task Mixture-of-Experts Re-ranking Framework for Abstractive Summarization - Mathieu Ravaut, Shafiq Joty, Nancy F. Chen

Presentation Author
Mathieu Ravaut

Nanyang Technological University

Table of content

Introduction

Typical leading approaches in abstractive summarization (PEGASUS [9], BART [3]) all share the same following ingredients :

- A **sequence-to-sequence** model architecture.
- Trained with the pre-training then fine-tuning paradigm.
- Summaries are generated with a **decoding method**. **Beam search** is widely used, with 4 to 10 beams.

Introduction

Decoding methods generate a **diverse set of candidates** :
Beam search example on XSum [6] :

- Candidate 1/15 : *The Turkish authorities have lifted a ban on female police officers wearing headscarves.*
  R-1 : 50.0, R-2 : 27.3, R-L : 41.7 // Mean-R rank : 11/15
- Candidate 2/15 : *Turkey has lifted a ban on female police officers wearing headscarves, the interior ministry says.*
  R-1 : 61.5, R-2 : 41.7, R-L : 52.1 // Mean-R rank : 1/15
- Candidate 3/15 : *The Turkish authorities have lifted a ban on female police officers wearing headscarves, state media report.*
  R-1 : 53.9, R-2 : 25.0, R-L : 53.9 // Mean-R rank : 8/15
- ...

Introduction

Only one candidate is kept : the **top beam** in beam search.

Not all candidates are equally good when compared to the target. The candidate maximizing score is called the **oracle**.

The oracle scores are up to **20% higher** than the baseline! This is more than the progress in the whole field of neural abstractive summarization since 2016.

Introduction

Decoding methods	# Summary candidates	R-1	R-2	R-L	BS	BaS
Beam search (top beam)	1	44.23	21.48	41.21	87.39	-2.78
Beam search	15	51.06	27.74	48.05	88.50	-2.48
Diverse beam search	15	**54.30**	**30.02**	**51.33**	**88.97**	**-2.40**
Top-k sampling	15	52.31	27.41	49.17	88.64	-2.56
Top-p sampling	15	53.52	28.88	50.46	88.87	-2.46
Adding all four methods above	60	**57.70**	**33.77**	**54.72**	**89.58**	**-2.25**

Table – Oracle scores with PEGASUS on CNN/DM.

All oracle scores **keep increasing** when mixing summaries from several decoding methods.

Introduction

Given the previous observations :

Sequence-to-sequence models in summarization are obviously not used to their full potential.

Can we fix this and train a 2nd-stage model selecting the best (oracle) summary candidate ? Or at least a better candidate ?

We propose **SummaReranker (SR)**, a model addressing this question.

Approach

We treat this problem as a **binary classification** : the candidate maximizing the score becomes the positive one, all the other candidates are negative.

- **Candidate 1/15 [label : 0]** : *The Turkish authorities have lifted a ban on female police officers wearing headscarves.*
  R-1 : 50.0, R-2 : 27.3, R-L : 41.7 // Mean-R rank : 11/15
- **Candidate 2/15 [label : 1]** : *Turkey has lifted a ban on female police officers wearing headscarves, the interior ministry says.*
  R-1 : 61.5, R-2 : 41.7, R-L : 52.1 // Mean-R rank : 1/15
- **Candidate 3/15 [label : 0]** : *The Turkish authorities have lifted a ban on female police officers wearing headscarves, state media report.*
  R-1 : 53.9, R-2 : 25.0, R-L : 53.9 // Mean-R rank : 8/15
- ...

Approach

What does ***maximizing the score*** mean in the first place ?
Which score ?

Summarization has always been **evaluated with multiple metrics** :

- ROUGE [4] and its 3 popular versions :
  - ROUGE-1 : word overlap
  - ROUGE-2 : proxy for fluency
  - ROUGE-L : longest common subsequence
- Recently proposed model-based metrics :
  - BERTScore [10]
  - BARTScore [8]
  - MoverScore [11]
  - ... (plenty others)

Approach

To optimize for multiple metrics, we transform the problem into a **multi-label binary classification** :

No.	**Candidate**	**R-1 Label**	**R-2 Label**	**R-L Label**
1/15	The Turkish authorities have lifted a ban on female police officers wearing headscarves.	0	0	0
2/15	Turkey has lifted a ban on female police officers wearing headscarves, the interior ministry says.	**1**	**1**	0
3/15	The Turkish authorities have lifted a ban on female police officers wearing headscarves, state media report.	0	0	**1**
...	...	...	...	...

Table – XSum summary candidates with multiple binary labels.

Approach

How to represent a candidate ?

Simply concatenate it after the source :
[CLS] `Source` [SEP] `Candidate`

Such concatenation enables **cross-attention** between the candidate and relevant parts of the source.

Approach

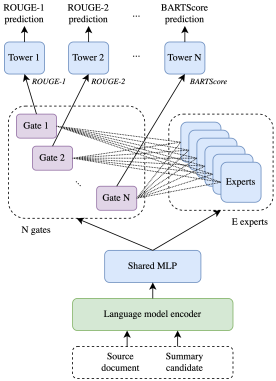


Figure – SummaReranker.

Key architecture elements :

- Encoder : RoBERTa-large [5].
- Multi-gate : one gate + prediction tower for each metric being optimized.
- Sparsely gated mixture-of-experts [7] : twice as many experts as gates and 50% expert dropout.
- Experts and towers : 2-layers MLPs.

Approach

How to train the model ?

For a metric μ, the re-ranker f_θ is trained with a **binary cross-entropy loss** :

$$\mathcal{L}_\mu = -y_i \log p_\theta^\mu(C_i) - (1 - y_i) \log(1 - p_\theta^\mu(C_i)) \qquad (1)$$

where $y_i = 1$ if candidate C_i is maximizing the metric μ.

The final loss is simply the **average** over metric losses defined as :

$$\mathcal{L} = \frac{1}{N} \sum_{\mu \in \mathbb{M}} \mathcal{L}_\mu \qquad (2)$$

Approach

To help the model, we **subsample candidates** during training :

- Rank candidates by sum of normalized metrics being optimized.
- Take the top m_{top} and bottom m_{bottom}.
- This yields positives in all metrics being optimized for.
- In practice, $m_{top} = 1$ and $m_{bottom} = 1$ work well. This means, only 2 candidates are used during training.
- Also tried an alternative sampling : sample a metric, and sample a positive candidate.

Approach

Why binary instead of multi-class classification ?
There is not enough separation between candidates.

Dataset	Model	Generation method	Scoring metric R-1	R-2	R-L	BS	BaS
CNN/DM	PEGASUS	{1}	11.51	10.87	11.54	14.96	14.96
		{2}	14.34	14.09	14.34	14.99	14.99
XSum	PEGASUS	{1}	8.90	7.91	8.56	14.99	14.99
		{2}	12.05	10.92	12.11	14.97	14.98
Reddit TIFU	PEGASUS	{1}	9.19	6.31	8.85	14.99	14.99
		{2}	7.84	5.06	7.77	14.89	14.97

Table – **Number of unique scores** among pools of 15 candidates.

BERTScore and BARTScore almost always assign different scores to all candidates.

Approach

How to tackle the **training/inference discrepancy** inherent to 2nd-stage approaches ?

- At training time :
  - Split the training dataset into two equal-size halves.
  - Train a model on one half, infer on the other half.
  - Train SummaReranker on the union of both inferred halves.
- At inference time, two options :
  - **Base setup** : infer on the validation or test set with one of the two models trained to build the training set.
  - **Transfer setup** : infer on the validation or test set with a model trained on the whole training set.

Scope

Varying all dimensions of this problem :

- 3 datasets :
  - CNN/DM
  - XSum
  - Reddit-TIFU
- 2 base models :
  - PEGASUS
  - BART
- Generate 15 candidates per model and decoding method.

- 4 decoding methods :
  - Beam search
  - Diverse beam search
  - Top-k sampling [1]
  - Top-p sampling [2]
- 5 evaluation metrics :
  - ROUGE-1
  - ROUGE-2
  - ROUGE-L
  - BERTScore
  - BARTScore

Decoding methods

We dissociate the set of decoding methods used for training $\mathbb{D}_{\text{train}}$, and the one used for testing $\mathbb{D}_{\text{test}}$.

Adding decoding methods to $\mathbb{D}_{\text{train}}$ **does not slow down training** because of the sampling.

We enforce $\mathbb{D}_{\text{test}} \subset \mathbb{D}_{\text{train}}$.

Metrics correlation

We note that metrics are heavily correlated :

	R-1	**R-2**	**R-L**	**BS**	**BaS**
R-1	1.000	0.884	0.977	0.858	0.662
R-2	0.884	1.000	0.910	0.833	0.665
R-L	0.977	0.910	1.000	0.855	0.669
BS	0.858	0.833	0.855	1.000	0.682
BaS	0.662	0.665	0.669	0.682	1.000

Table – **Pearson correlation coefficient** for a base PEGASUS with beam search on CNN/DM.

Base setup

Model	**Model stage**	**Decoding methods ($\mathbb{D}$)**	**R-1**	**R-2**	**R-L**	**Gain (%)**
PEGASUS - 1st half	1	{1}	42.23	19.62	38.90	_
PEGASUS - 1st half	1	{2}	42.50	19.75	39.55	_
PEGASUS - 2nd half	1	{1}	42.46	**19.95**	39.19	_
PEGASUS - 2nd half	1	{2}	**42.75**	19.93	**39.86**	_
PEGASUS - 1st half + **SR**	2	{1}	44.02	20.97	40.68	5.23
PEGASUS - 1st half + **SR**	2	{2}	**45.66**	**21.31**	42.51	7.61
PEGASUS - 2nd half + **SR**	2	{1}	44.11	21.08	40.82	4.57
PEGASUS - 2nd half + **SR**	2	{2}	45.73	**21.31**	**42.62**	6.94
PEGASUS - 1st half + **SR**	2	{1, 2}	46.12	21.97	42.84	9.36
PEGASUS - 2nd half + **SR**	2	{1, 2}	**46.19**	**22.02**	**42.92**	8.70

Table – **Base setup results** for PEGASUS+SummaReranker on **CNN/DM**.

Base setup

Model	Model stage	Decoding methods ($\mathbb{D}$)	R-1	R-2	R-L	Gain (%)
BART - 1st half	1	{1}	42.79	20.25	39.66	_
BART - 1st half	1	{2}	40.70	18.99	37.88	_
BART - 2nd half	1	{1}	**42.93**	**20.36**	**39.73**	_
BART - 2nd half	1	{2}	41.93	19.79	39.06	_
BART - 1st half + **SR**	2	{1}	44.23	21.23	41.09	3.94
BART - 1st half + **SR**	2	{2}	45.05	21.47	42.12	11.65
BART - 2nd half + **SR**	2	{1}	44.51	21.52	41.29	4.44
BART - 2nd half + **SR**	2	{2}	**45.61**	**21.78**	**42.62**	9.32
BART - 1st half + **SR**	2	{1, 2}	45.76	22.14	42.71	7.99
BART - 2nd half + **SR**	2	{1, 2}	**45.96**	**22.18**	**42.88**	7.98

Table – **Base setup results** for BART+SummaReranker on **CNN/DM**

Transfer setup

Model	Model stage	Decoding methods $\mathbb{D}_{\text{train}}$	Decoding methods $\mathbb{D}_{\text{test}}$	Decoding methods m	Optimized Metrics ($\mathbb{M}$)	Evaluation metrics R-1	Evaluation metrics R-2	Evaluation metrics R-L	Evaluation metrics BS	Evaluation metrics BaS	Gain (%)
PEGASUS	1	{1}	{1}	8	_	44.16	21.56	41.30	_	_	_
PEGASUS - *our setup*	1	{1}	{1}	15	_	44.23	21.48	41.21	87.39	-2.78	_
PEGASUS - *our setup*	1	{2}	{2}	15	_	44.56	20.90	41.58	87.36	-2.81	_
BART + R3F	1	{1}	{1}	5	_	44.38	21.53	41.17	_	_	_
GSum	1	{1}	{1}	4	_	**45.94**	**22.32**	**42.48**	_	_	_
GSum + RefSum	2	{1}	{1}	4	_	46.18	22.36	42.91	_	_	_
BART + SimCLS	2	{2}	{2}	16	_	46.67	22.15	43.54	66.14	_	_
PEGASUS + **SR**	2	{1}	{1}	15	{R-1, R-2, R-L}	45.56†	22.23†	42.46†	87.60†	-2.74†	3.18
PEGASUS + **SR**	2	{2}	{2}	15	{R-1, R-2, R-L}	**46.86**†	22.01†	**43.59**†	87.66†	-2.73†	5.10
PEGASUS + **SR**	2	{1, 2}	{1}	15	{R-1, R-2, R-L}	46.13†	**22.61**†	42.94†	87.67†	-2.72†	4.59
PEGASUS + **SR**	2	{1, 2}	{2}	15	{R-1, R-2, R-L}	46.83†	21.88†	43.55†	87.63†	-2.74†	4.84
PEGASUS + **SR** (**new SOTA**)	2	{1, 2}	{1, 2}	30	{R-1, R-2, R-L}	**47.16**†	**22.55**†	**43.87**†	87.74†	-2.71†	**5.44**
PEGASUS + **SR**	2	{1, 2}	{1, 2}	30	{BS, BaS}	45.00†	20.90	41.93†	87.56†	-2.55†	4.23
PEGASUS + **SR**	2	{1, 2}	{1, 2}	30	{R-1, R-2, R-L, BS, BaS}	46.59†	22.41†	43.45†	87.77†	-2.58†	4.39
PEGASUS + **SR**	2	{1, 2, 3, 4}	{1, 2, 3, 4}	60	{R-1, R-2, R-L}	**47.04**†	**22.32**†	**43.72**†	**87.69**†	**-2.74**†	_

Table – **Transfer setup results on CNN/DM.** † marks are results significantly better than the base model counterpart among metrics that SummaReranker was optimized for.

Transfer setup

Model	Model stage	Decoding methods $\mathbb{D}_{train}$	$\mathbb{D}_{test}$	m	XSum R-1	R-2	R-L	BS	BaS	Gain (%)
PEGASUS	1	{1}	{1}	8	47.21	24.56	39.25	_	_	_
PEGASUS - *our setup*	1	{1}	{1}	15	**47.33**	**24.75**	**39.43**	**92.01**	**-1.92**	_
PEGASUS - *our setup*	1	{2}	{2}	15	46.78	23.77	38.70	91.94	-2.00	_
BART	1	{1}	{1}	5	45.14	22.27	37.25	_	_	_
BART - *our setup*	1	{1}	{1}	15	45.24	22.28	37.21	91.58	-1.97	_
BART - *our setup*	1	{2}	{2}	15	44.15	20.84	35.88	91.51	-2.08	_
BART + R3F	1	{1}	{1}	5	_	_	_	_	_	_
GSum + RefSum	2	{1}	{1}	4	47.45	24.55	39.41	_	_	_
PEGASUS + SimCLS	2	{2}	{2}	16	47.61	24.57	39.44	69.81	_	_
PEGASUS + **SR** (**new SOTA**)	2	{1, 2}	{1}	15	**48.12**†	**24.95**	**40.00**†	**92.14**†	**-1.90**†	**1.31**
PEGASUS + **SR**	2	{1, 2}	{2}	15	47.04	23.27	38.55	91.98	-2.01	-0.65
BART + **SR**	2	{1, 2}	{1}	15	45.79†	22.17	37.31	91.69†	-1.97	0.33
BART + **SR**	2	{1, 2}	{2}	15	44.39	20.35	35.66	91.51	-2.16	-0.81
PEGASUS + **SR**	2	{1, 2}	{1, 2}	30	**47.72**	**24.16**	**39.42**	**92.10**†	**-1.94**	-0.53
BART + **SR**	2	{1, 2}	{1, 2}	30	45.32	21.46	36.64	91.64	-2.04	-1.68

Table – **Transfer setup results on XSum.**

Transfer setup

Model	Model stage	Decoding methods $\mathbb{D}_{\text{train}}$	$\mathbb{D}_{\text{test}}$	m	Reddit TIFU R-1	R-2	R-L	BS	BaS	Gain (%)
PEGASUS	1	{1}	{1}	8	*26.63*	*9.01*	*21.60*	_	_	_
PEGASUS - *our setup*	1	{1}	{1}	15	26.28	9.01	21.52	87.34	**-3.46**	_
PEGASUS - *our setup*	1	{2}	{2}	15	25.67	8.07	20.97	87.47	-3.48	_
BART - *our setup*	1	{1}	{1}	15	**27.42**	**9.53**	**22.10**	87.43	-3.78	_
BART - *our setup*	1	{2}	{2}	15	25.43	8.27	20.79	**87.48**	-4.19	_
BART + R3F	1	{1}	{1}	5	*30.31*	*10.98*	*24.74*	_	_	_
PEGASUS + **SR**	2	{1, 2}	{1}	15	**29.57**†	9.70†	**23.29**†	87.63†	**-3.34**†	9.47
PEGASUS + **SR**	2	{1, 2}	{2}	15	28.71†	8.73†	22.79†	**87.84**†	-3.42†	9.57
BART + **SR**	2	{1, 2}	{1}	15	28.99†	**9.82**	22.96†	87.53	-3.78	4.22
BART + **SR**	2	{1, 2}	{2}	15	28.04†	8.66	22.41†	87.73†	-3.91†	7.59
PEGASUS + **SR** (**best Reddit TIFU score**)	2	{1, 2}	{1, 2}	30	**29.83**†	**9.50**†	**23.47**†	**87.81**†	**-3.33**†	**9.34**
BART + **SR**	2	{1, 2}	{1, 2}	30	28.92†	9.16	22.87†	87.70†	-3.83†	1.69

Table – **Transfer setup results on Reddit TIFU**. Results in italic are not directly comparable due to a different data split.

Ranking evaluation

SummaReranker improves the **best candidate recall** compared to random ranking and top beam ranking baselines.

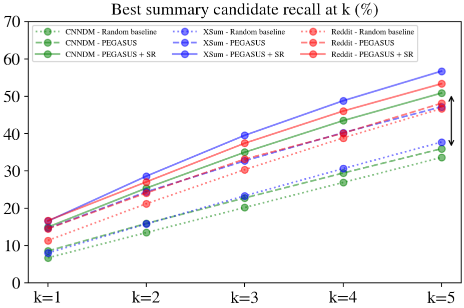


Figure – Best summary candidate recall. PEGASUS base model with diverse beam search.

Human evaluation

SummaReranker selected summaries are deemed **more informative** by humans compared to the top beam summaries.

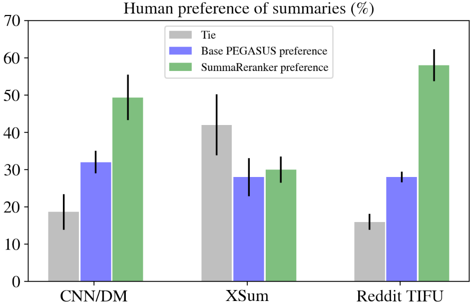
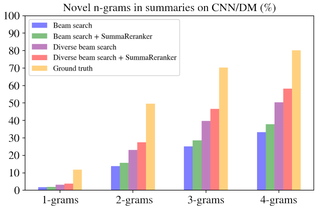
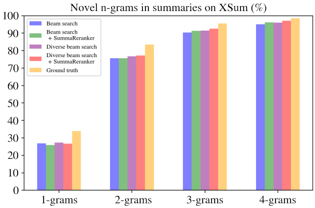


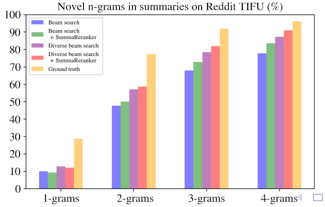
Figure – Human evaluation : 3 humans and 50 samples per dataset. PEGASUS base model with beam search.

Abstractiveness

SummaReranker selected summaries **are more abstractive on CNN/DM and Reddit TIFU**.







Speed-performance trade-off

The more summaries, the greater the gains, but at a higher computation cost. 6-7 candidates is a sweet spot.

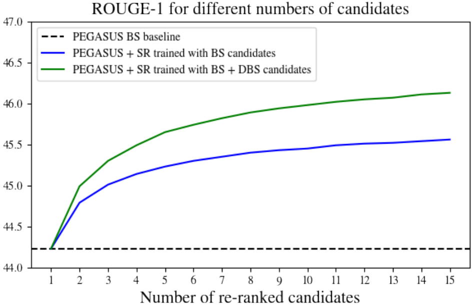


Figure – ROUGE-1 on CNN/DM when ranking an increasing number of sampled summaries.

Multi-tasking

Experts specialize in different tasks (e.g, expert 4 in ROUGE-L).

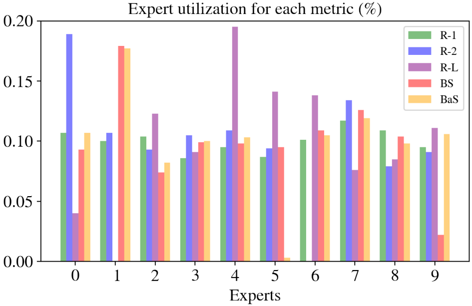


Figure – Expert utilization (10 experts) when optimizing all 5 metrics for PEGASUS on CNN/DM.

Limitations

SummaReranker presents some important limitations :

- We need to already fine-tuned base models.
- We need to generate candidates from the base models.
- Scoring all candidates takes time.
- Encoding the concatenation of the source with a candidate is limited by RoBERTa's context window of size 512.

Conclusion

We introduced the **first multi-task model for 2nd-stage summarization**.

It jointly encodes the source with each candidate and predicts whether the candidate maximizes each metric. The multi-tasking makes it flexible and can optimize any set of metrics.

The method works well :

- It reaches ROUGE SOTA when optimizing for ROUGE.
- The reranking is effective and significantly improves the best candidate recall.
- Summaries are more abstractive.
- Summaries are more informative according to humans.

[1] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv :1805.04833*, 2018.

[2] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv :1904.09751*, 2019.

[3] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart : Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv :1910.13461*, 2019.

[4] Chin-Yew Lin. Rouge : A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*, 2019.

[6] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary ! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv :1808.08745*, 2018.

[7] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks : The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv :1701.06538*, 2017.

[8] Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore :

Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34, 2021.

[9] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus : Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[10] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore : Evaluating text generation with bert. *arXiv preprint arXiv :1904.09675*, 2019.

[11] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. Moverscore : Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv :1909.02622*, 2019.