Improving Neural Abstractive Summarization via Reinforcement Learning with BERTScore

Yuhui Zhang, Ruocheng Wang, Zhengping Zhou

{yuhuiz, rcwang, zpzhou}@stanford.edu

Problem Statement

- Summarization: news, laws, clinical, biomedical.
- Machine learning approach: modeling NLL loss.

Exposure Bias: "gap" between training and inference.

- Reinforcement learning approach[1]: ROUGE as reward.
- Better reward yields better results!

Can we find a better reward?

From ROUGE to BERTScore

- ROUGE: **n-gram hard-match** between gen and ref.
- BERTScore[2]: **token soft-match** between gen and ref.

Ref: *European finance ministers urge Swedes to vote yes to euro.*
Gen: *Swedes were asked to support euro by EU finance ministers.*

ROUGE-L 👎 (Top 31%) **BERTScore** 👍 (Top 1%)

Metric	Spearman ρ	Pearson ρ
ROUGE-L	14.54	14.51
BERTScore	**21.26**	**22.75**

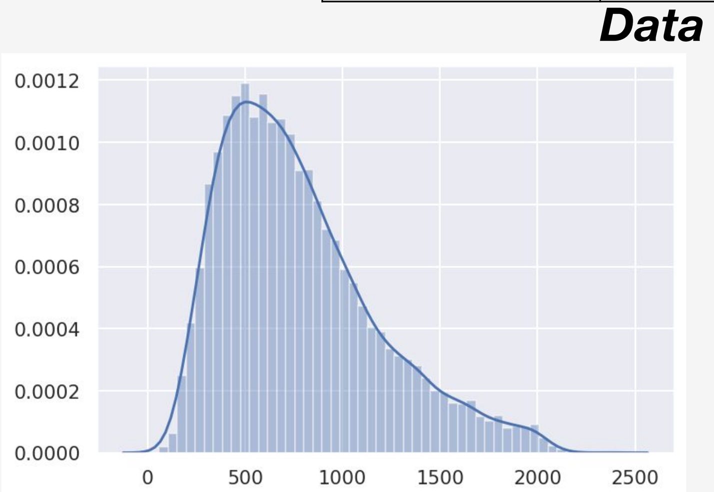
BERTScore correlates with human much better.

Dataset

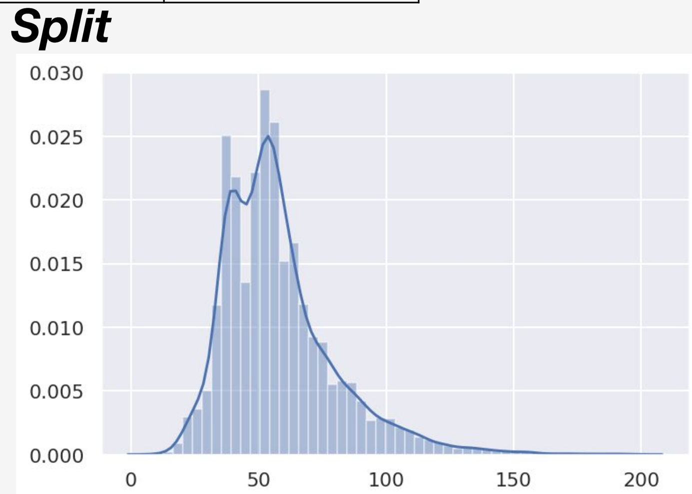
We used **CNN/Daily Mail**[3] dataset which contains online news articles along with human summaries.

Train	Dev	Test
287K	13K	11K

Data Split



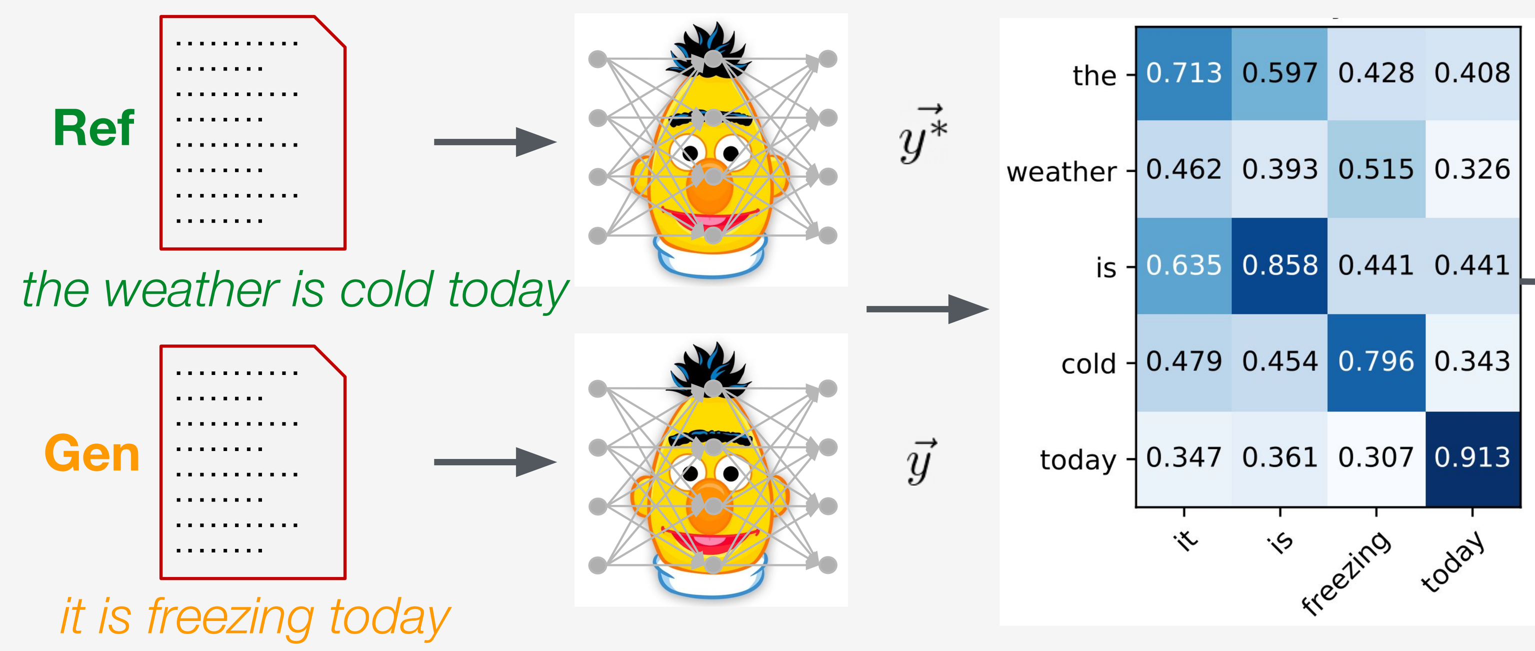
Source Length Distribution



Reference Length Distribution

BERTScore

The overview of **BERTScore**[2] computation:



Contextualized Word Embedding **Word-Word Cosine Similarity**

$$R_{\text{BERT}} = \frac{1}{d^*} \sum_{y_i^* \in y^*} \max_{y_j \in y} \vec{y}_i^{*T} \vec{y}_j$$

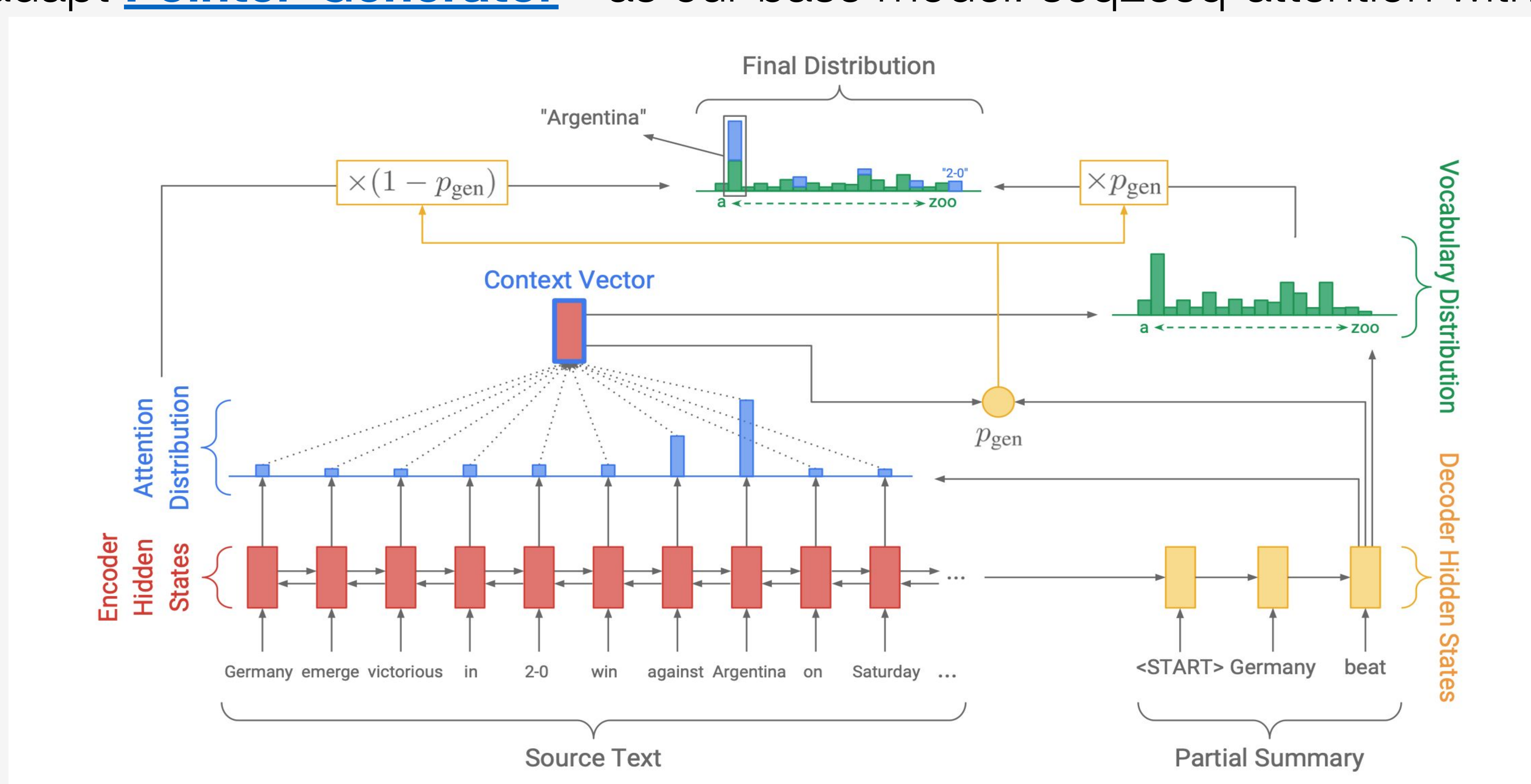
$$P_{\text{BERT}} = \frac{1}{d} \sum_{y_i \in y} \max_{y_j^* \in y^*} \vec{y}_i^{T} \vec{y}_j^*$$

$$F_{\text{BERT}} = \frac{2 R_{\text{BERT}} P_{\text{BERT}}}{R_{\text{BERT}} + P_{\text{BERT}}}$$

used as the final score

Model Architectures

We adapt **Pointer-Generator**[4] as our base model: seq2seq-attention with copy.



Credit: https://arxiv.org/pdf/1704.04368.pdf

$$\mathcal{L}_{\text{nll}}(\theta) = -\sum_{t=1}^{d^*} \log p(y_t^* | y_{t-1}^*, \dots, y_1^*, x_n, \dots, x_1; \theta)$$ ***ML Approach: NLL loss of the ref***

$$\mathcal{L}_{\text{rl}}(\theta) = (r(y^g) - r(y^s)) \sum_{t=1}^{d^s} \log p(y_t^s | y_{t-1}^s, \dots, y_1^s, x_n, \dots, x_1; \theta)$$ ***RL Approach: policy gradient***

y^s: summary generated by randomly sampling from the word distribution.
y^g: summary generated by greedy decoding (argmax for each word).
(reduce the variance of the policy gradient estimation)

$$\mathcal{L}(\theta) = (1-\gamma)\mathcal{L}_{\text{nll}}(\theta) + \gamma \mathcal{L}_{\text{rl}}(\theta)$$ ***ML+RL Approach: mixed objective***

Results & Analysis

Numerical Evaluation:

- ***Baseline*** (Pointer-Generator[4])
- ***RL-ROUGE*** (Baseline + RL on ROUGE-L)
- ***RL-BERTScore*** (Baseline + RL on BERTScore)

Method	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Baseline	39.37	17.15	34.67	60.77
RL-ROUGE	**43.28**↑	**19.11**↑	**38.55**↑	59.26↓
RL-BERTScore	42.60↑	18.69↑	36.58↑	**62.77**↑

RL-BERTScore: *ROUGE↑ and BERTScore↑*

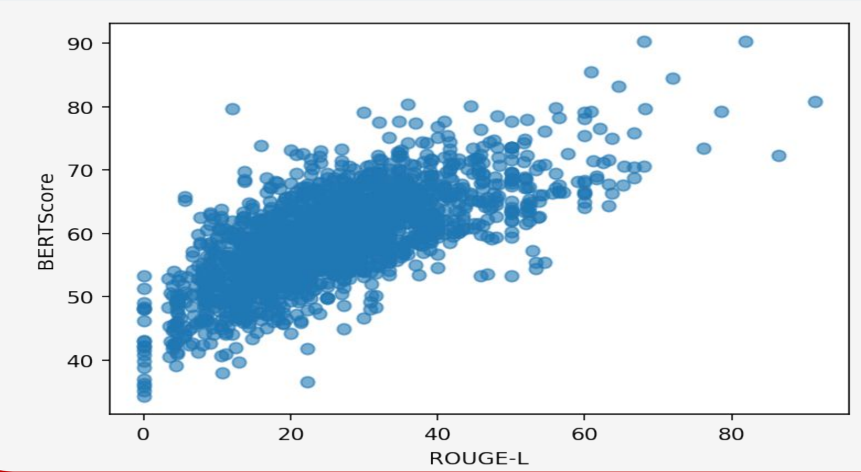
Human Evaluation:

*Manually compare **50 summaries** generated by **RL-BERTScore** and **RL-ROUGE**.*

Metric	Win	Tie	Loss
Fluency	**32%**	56%	12%
Redundancy	**62%**	18%	20%
Overall	**46%**	40%	14%

Better quality for RL-BERTScore

Correlation between BERTScore and ROUGE:



X-axis: ROUGE-L
Y-axis: BERTScore

strongly correlated

Discussion & Future Work

- Further improvement → **hyperparameters tuning**.
- Single reward → **multiple rewards** from ROUGE-1, ROUGE-L, BLEU, BERTScore.
- Learning to learn → use linear regression on annotated data to **learn coefficients of different rewards**.

[1] Paulus, Romain, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In ICLR (2018).
[2] Zhang, Tianyi, et al. *BERTScore: Evaluating Text Generation with BERT.* arXiv:1904.09675.
[3] Hermann, Karl Moritz, et al. Teaching machines to read and comprehend. In NIPS (2015).
[4] See, Abigail, Peter J. Liu, and Christopher D. Manning. *Get To The Point: Summarization with Pointer-Generator Networks.* In ACL (2017).