Paper Reading

Extractive Summarization as Text Matching

M1 Shogo Fujita

2020/7/23

Introduction

Analysis

Model

Experiment

Conclusion

Introduction

Task Extract summary from documents.

Contribution

- They formulate extractive summarization as a semantic text matching problem and propose a novel summary-level framework.
- They quantify the inherent gap between sentence-level and summary-level methods.

Questions

The following two questions will be analyzed.

- For extractive summarization, is the summary-level extractor better than the sentence-level extractor?
- Given a dataset, which extractor should we choose based on the characteristics of the data, and what is the inherent gap between these two extractors?

Definition

- $D = \{s_1, s_2, ..., s_n\}$ is a document consisting of n sentences.
- $C = \{s_1, s_2, ..., s_k | s_i \in D\}$ is a candidate summary including k sentences extracted from a document.
- C^* is a gold summary.
- $R(\cdot)$ denotes the average F1 of ROUGE-1, ROUGE-2 and ROUGE-L.

Definition

They measure a candidate summary C by calculating the ROUGE value between C and C^* in two levels:

Sentence-Level Score

$$g^{gen}(C) = \frac{1}{|C|} \sum_{s \in C} R(s, C^*)$$

Summary-Level Score

$$g^{sum}(C) = R(C, C^*)$$

Definition

Pearl-Summary A candidate summary C is defined as a pearl-summary if there exists another candidate summary C' that satisfies the inequality: $g^{sen}(C') > g^{sen}(C)$ while $g^{sum}(C') < g^{sum}(C)$.

Best-Summary A summary $\hat{C}$ is defined as the best-summary when it satisfies: $\hat{C} = argmax_{C \in \mathcal{C}} g^{sum}(C)$, where $\mathcal{C}$ denotes all the candidate summaries of the document.

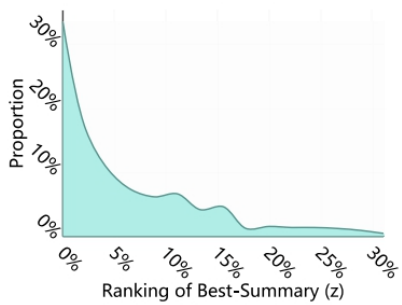
Dataset Overview

Datasets	Source	Type	# Pairs			# Tokens		# Ext
			Train	Valid	Test	Doc.	Sum.	
Reddit	Social Media	SDS	41,675	645	645	482.2	28.0	2
XSum	News	SDS	203,028	11,273	11,332	430.2	23.3	2
CNN/DM	News	SDS	287,084	13,367	11,489	766.1	58.2	3
WikiHow	Knowledge Base	SDS	168,126	6,000	6,000	580.8	62.6	4
PubMed	Scientific Paper	SDS	83,233	4,946	5,025	444.0	209.5	6
Multi-News	News	MDS	44,972	5,622	5,622	487.3	262.0	9

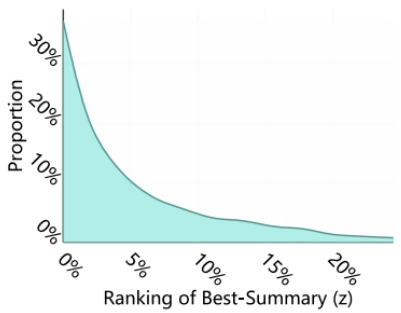
Table 1: Datasets overview. SDS represents single-document summarization and MDS represents multi-document summarization. The data in Doc. and Sum. indicates the average length of document and summary in the test set respectively. # Ext denotes the number of sentences should extract in different datasets.

Distribution of $z\%$ on datasets

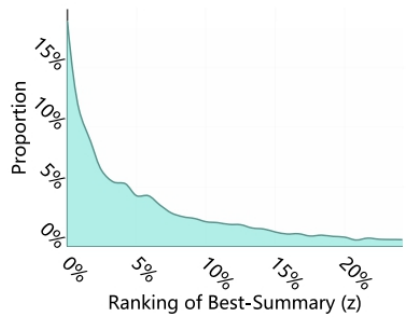
For each document, we sort all candidate summaries in descending order based on the sentence-level score, and then define z as the rank index of the best-summary $\hat{C}$.



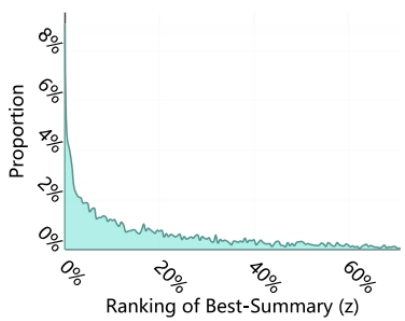
(a) Reddit



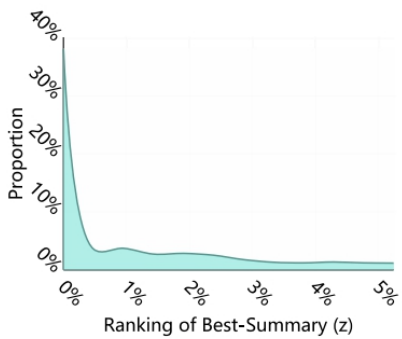
(b) XSum



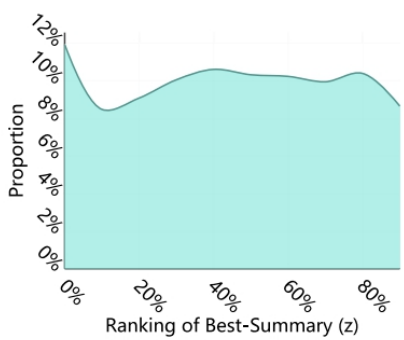
(c) CNN/DM



(d) WikiHow



(e) PubMed



(f) Multi-News

Dataset analysis

- Specifically, for CNN/DM, only 18.9% of best-summaries are not pearl-summary.
- PubMed is most suitable for sentence-level summarizers, because most of best-summary sets are not pearl-summary.
- WikiHow and Multi-News are most evenly distributed.

Potential gain

A dataset-level potential gain$\Delta(\mathcal{D})$ can be obtained as:

$$\Delta(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{D \in \mathcal{D}} \alpha^{sum}(D) - \alpha^{sen}(D) \quad (1)$$

$$\alpha^{sen}(D) = max_{C \in \mathcal{C}_D} g^{sen}(C) \quad (2)$$

$$\alpha^{sum}(D) = max_{C \in \mathcal{C}_D} g^{sum}(C) \quad (3)$$

Potential gain

For a medium-length summary(`CNN/DM` and `WikiHow`, about 60 words), the summary-level learning process is rewarding.

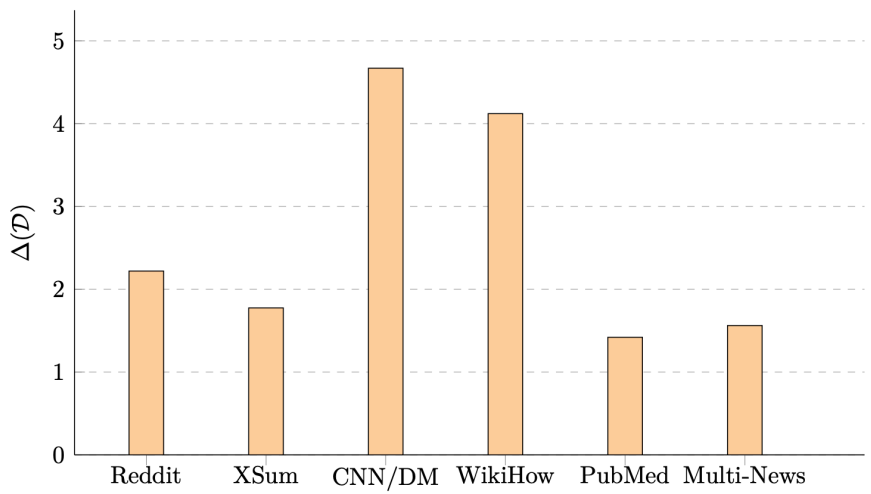


Figure 3: $\Delta(\mathcal{D})$ for different datasets.

Siamese-BERT

Similarity function $f(D, C)$ is defined as follows:

$$f(D, C) = cosine(r_D, r_C) \tag{4}$$

Let r_D and r_C denote the embeddings of the document D and candidate summary C. These are the vectors corresponding to `[CLS]` of the output of the BERT mechanism.

Siamese-BERT

In order to fine-tune Siamese-BERT, they use a margin-based triplet loss $\mathcal{L}$ to update the weights.

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \tag{5}$$

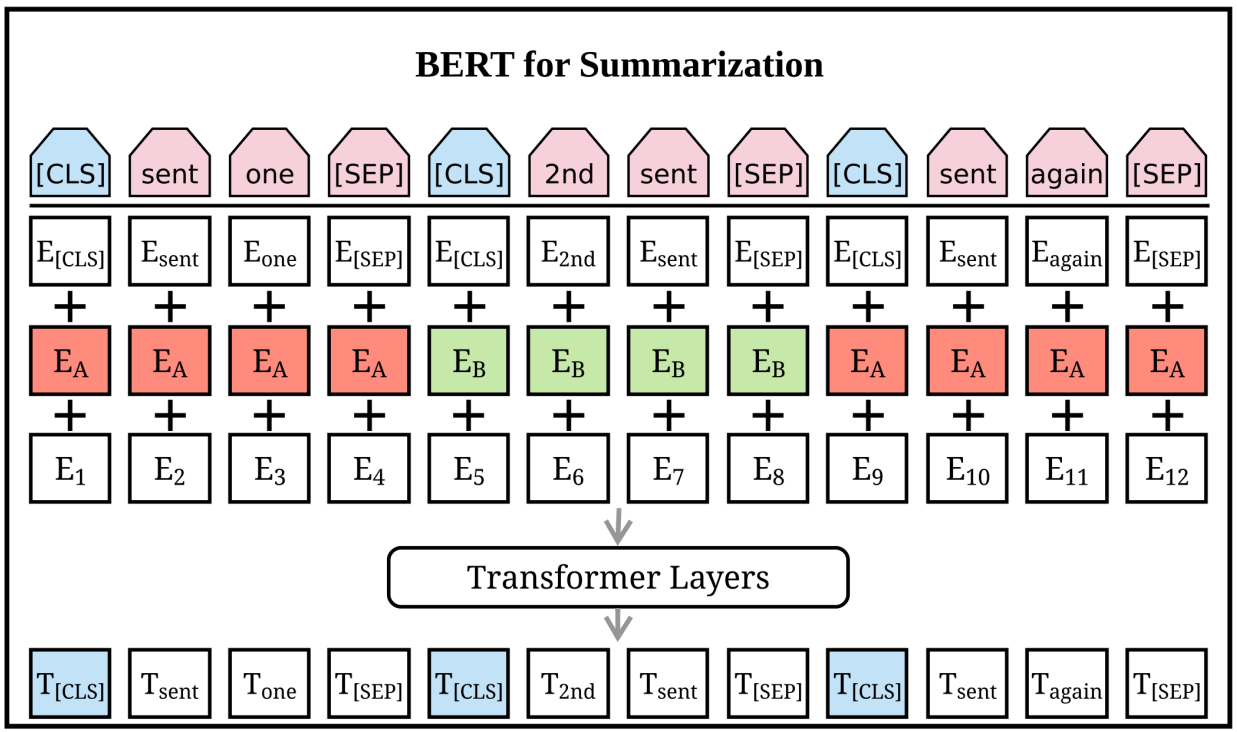
$$\mathcal{L}_1 = max(0, f(D, C) - f(D, C^*) + \gamma_1) \tag{6}$$

$$\mathcal{L}_2 = max(0, f(D, C_j) - f(D, C_i) + (j - i) * \gamma_2) \ (i < j) \tag{7}$$

When predicting, the model choose $\hat{C} = max_{C \in \mathcal{C}} f(D, C)$.

Candidates Pruning

They have introduced a content selection module, much like BERTSUM(Liu and Lapata, 2019), for pre-selecting salient sentenses.



Experiment

MATCHSUM The model of choice for $\hat{C}$.

BERTEXT A model that uses BERTSUM to score each sentence and selects them in order of the score.

Experiment

Model	R-1	R-2	R-L
LEAD	40.43	17.62	36.67
ORACLE	52.59	31.23	48.87
MATCH-ORACLE	51.08	26.94	47.22
BANDITSUM (Dong et al., 2018)	41.50	18.70	37.60
NEUSUM (Zhou et al., 2018)	41.59	19.01	37.98
JECS (Xu and Durrett, 2019)	41.70	18.50	37.90
HIBERT (Zhang et al., 2019b)	42.37	19.95	38.83
PNBERT (Zhong et al., 2019a)	42.39	19.51	38.69
PNBERT + RL	42.69	19.60	38.85
BERTEXT† (Bae et al., 2019)	42.29	19.38	38.63
BERTEXT† + RL	42.76	19.87	39.11
BERTEXT (Liu, 2019)	42.57	19.96	39.04
BERTEXT + Tri-Blocking	43.23	20.22	39.60
BERTSUM* (Liu and Lapata, 2019)	43.85	20.34	39.90
BERTEXT (Ours)	42.73	20.13	39.20
BERTEXT + Tri-Blocking (Ours)	43.18	20.16	39.56
MATCHSUM (BERT-base)	44.22	20.62	40.38
MATCHSUM (RoBERTa-base)	**44.41**	**20.86**	**40.55**

Table 3: Results on CNN/DM test set. The model with * indicates that the large version of BERT is used. BERTEXT† add an additional Pointer Network compared to other BERTEXT in this table.

Model	R-1	R-2	R-L
Reddit			
BERTEXT (Num = 1)	21.99	5.21	16.99
BERTEXT (Num = 2)	23.86	5.85	19.11
MATCHSUM (Sel = 1)	22.87	5.15	17.40
MATCHSUM (Sel = 2)	24.90	5.91	20.03
MATCHSUM (Sel = 1, 2)	**25.09**	**6.17**	**20.13**
XSum			
BERTEXT (Num = 1)	22.53	4.36	16.23
BERTEXT (Num = 2)	22.86	4.48	17.16
MATCHSUM (Sel = 1)	23.35	4.46	16.71
MATCHSUM (Sel = 2)	24.48	4.58	18.31
MATCHSUM (Sel = 1, 2)	**24.86**	**4.66**	**18.41**

Table 4: Results on test sets of Reddit and XSum. *Num* indicates how many sentences BERTEXT extracts as a summary and *Sel* indicates the number of sentences we choose to form a candidate summary.

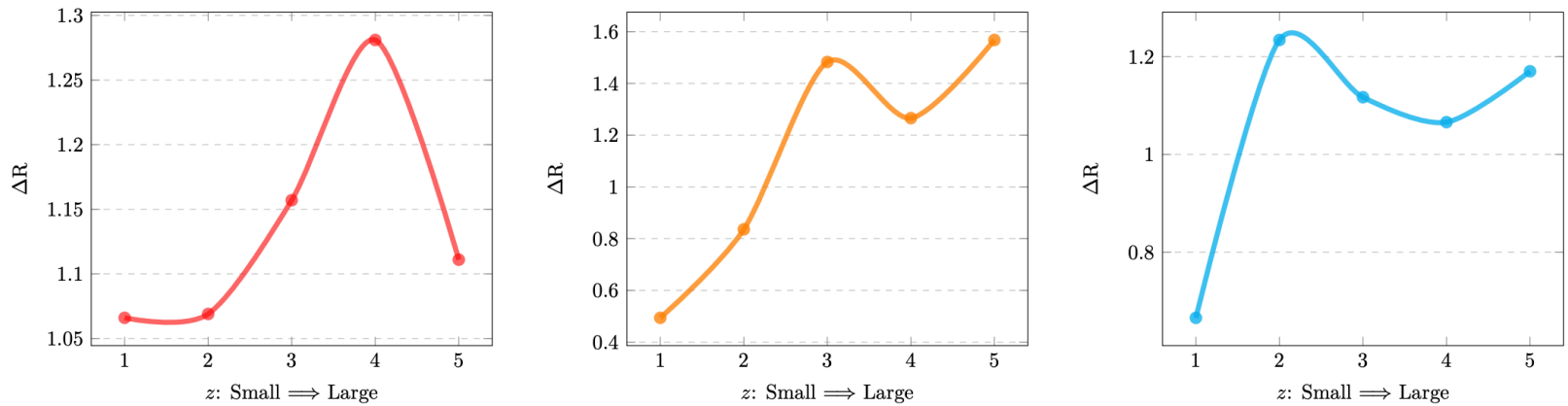
Experiment

Model	WikiHow			PubMed			Multi-News		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LEAD	24.97	5.83	23.24	37.58	12.22	33.44	43.08	14.27	38.97
ORACLE	35.59	12.98	32.68	45.12	20.33	40.19	49.06	21.54	44.27
MATCH-ORACLE	35.22	10.55	32.87	42.21	15.42	37.67	47.45	17.41	43.14
BERTEXT	30.31	8.71	28.24	41.05	14.88	36.57	45.80	16.42	41.53
+ 3gram-Blocking	30.37	8.45	28.28	38.81	13.62	34.52	44.94	15.47	40.63
+ 4gram-Blocking	30.40	8.67	28.32	40.29	14.37	35.88	45.86	16.23	41.57
MATCHSUM (BERT-base)	**31.85**	**8.98**	**29.58**	**41.21**	**14.91**	**36.75**	**46.20**	**16.51**	**41.89**

Table 5: Results on test sets of WikiHow, PubMed and Multi-News. MATCHSUM beats the state-of-the-art BERT model with Ngram Blocking on all different domain datasets.

Dataset Splitting Testing

The figure shows that the performance gap between MATCHSUM and BERTEXT is always the smallest when the best-summary is not a pearlsummary



(a) XSum (b) CNN/DM (c) WikiHow

Figure 4: Datasets splitting experiment. We split test sets into five parts according to z described in Section 3.2. The X-axis from left to right indicates the subsets of the test set with the value of z from small to large, and the Y-axis represents the ROUGE improvement of MATCHSUM over BERTEXT on this subset.

Comparison Across Datasets

They introduce $\Delta(D)^*$ as follows:

$$\psi(\mathcal{D}) = \Delta(D)^* / \Delta(\mathcal{D}) \tag{8}$$

$$\Delta(D)^* = g^{sum}(C_{MS}) - g^{sum}(C_{BE}) \tag{9}$$

$$\Delta(\mathcal{D})^* = \frac{1}{|\mathcal{D}|} \sum_{D \in \mathcal{D}} \Delta(D)^* \tag{10}$$

where C_{MS} and C_{BE} represent the candidate summary selected by MATCHSUM and BERTEXT.

Comparison Across Datasets

As the gold summaries get longer, the upper bound of summary-level approaches becomes more difficult for the proposed model to reach.

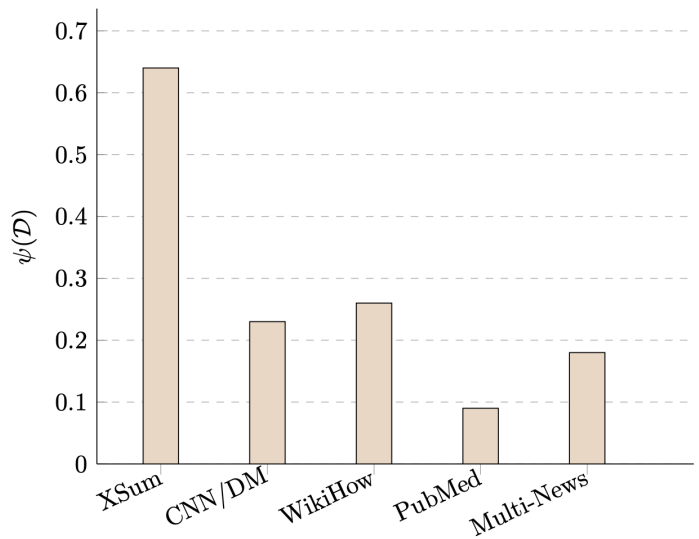


Figure 5: ψ of different datasets. Reddit is excluded because it has too few samples in the test set.

Conclusion

- They propose a summary-level framework to match the source document and candidate summaries in the semantic space.
- They conduct an analysis to show how their model could better fit the characteristic of the data.
- Experimental results show MATCHSUM outperforms the current SOTA extractive model on six benchmark datasets.