
[Data Science](#)[Natural Language Processing](#)[NLP Papers Summary](#)

Day 160: NLP Papers Summary – Extractive Summarization As Text Matching

By Ryan 8th June 2020 No Comments

Objective and Contribution

We tackle extractive summarisation task as a semantic text matching problem rather than the common used sequence labelling problem. We proposed MATCHSUM, a novel summary-level framework that uses Siamese-BERT to match source document and candidate summaries in the semantic space. The idea is that a good summary should be semantically similar to the source document as a whole. This method achieved 44.41 ROUGE-1 score in CNN/I  :aset

and achieved similar results in other evaluation datasets. We also analyse the performance difference between sentence-level and summary-level extractive models.

Datasets

We have six evaluation datasets as shown below and our evaluation metrics are ROUGE-1, ROUGE-2, and ROUGE-L.

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.

MATCHSUM

SIAMESE-BERT

We use siamese-BERT, which consists of two BERTs with the same weight and a cosine similarity layer, to match document and candidate summary. We use BERT to encode both the document and candidate summaries and compute the similarity between the two embeddings. The basic idea is that gold summary has the highest matching score to the source document and good candidate summary should obtain high score.

CANDIDATES PRUNING

To avoid having to score all possible candidates, we use a simple candidate pruning strategy. Specifically, we used a content selection technique to select key sentences where each sentence gets a salience score. We will use these scores to prune the sentences that are irrelevant with the current document. Here, we use BERTSUM as the content selection module.



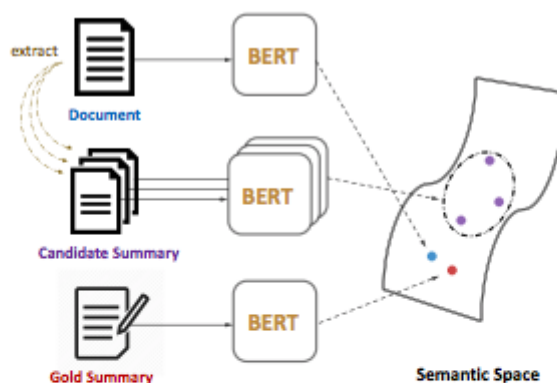


Figure 1: MATCHSUM framework. We match the contextual representations of the document with gold summary and candidate summaries (extracted from the document). Intuitively, better candidate summaries should be semantically closer to the document, while the gold summary should be the closest.

Sentence-level vs Summary-level Extractive Summarisation

We explored 6 evaluation datasets to find out which type of extractor performs best based on the characteristics of the data. In the experiment, each document has a list of candidate extractive summaries. These candidate summaries are compared to the gold summary using ROUGE in two levels:

1. *Sentence-level*. Computing the average overlaps between each sentence in the candidate summary and the gold summary
2. *Summary-level*. Computing the ROUGE score between candidate summary as a whole and the gold summary

We define two types of summary: Pearl-summary and Best-summary. Pearl-summary is a summary that has a lower sentence-level score but a higher summary-level score than another candidate summary. Pearl summaries are difficult to be extracted by sentence-level extractors. Best-summary is the summary with the highest summary-level score among the candidate summaries. We rank best-summary by first sorting all candidate summaries in descending order using the sentence-level score and then assign a rank index z to the best-summary C . If $z = 1$, it means the best-summary also has the highest sentence-level score. If $z > 1$, the best-summary is a pearl-summary and as z increases, it makes it more difficult to find the best-summary as there are more and more candidate summaries with a higher score than the best-summary.

We explored the proportion of pearl-summary in different datasets as shown in the figure below. We can see that most of the best-summaries are not made up of the highest scoring sentences. Only 18.9% of best-summaries are not pearl-summary in CNN/DM, meaning that sentence-level extractive models can easily miss better candidate summaries. PubMed is the most suitable for sentence-level extractors as they have the highest proportion of best-summary that are not pearl-summary. Summary-level extractors work best in WikiHow and Multi-News as they have more equally distribution. Overall, the proportion of pearl-summaries in all the best-summaries will help determine which type of extractive models to choose.



SO HOW MUCH IMPROVEMENT CAN SUMMARY-LEVEL BRING?

We capture the upper bound performance of both sentence-level and summary-level in ⁶ evaluation datasets and the results are displayed below. The performance improvement varies between datasets but it ranges between 1.2 – 4.7, with CNN/DM achieving 4.7 improvement. We observed a relationship between performance gain and the length of summary. Summary-level doesn't work well with short or long summaries. With long summaries, there are already a large semantic overlap making summary-level improvement incremental. The best are medium-length summary like CNN/DM and WikiHow.

Results

The results for CNN/DM are displayed below. As shown, our MATCHSUM outperformed all the other baseline models. We observed the best performance is achieved when when we change the encoder to RoBERTa-base. We believe this is due to RoBERTa was pre-trained using 63 million news articles.




We split out results analysis into two categories: short summaries and long summaries. For the results on short summaries, we evaluated the Reddit and XSum dataset to see if MATCHSUM can improve the performance. The results are displayed in the table 4 below. Our MATCHSUM was able to still outperform BERTEXT, however, the margin is small. We observe that as we increase the number of sentences used to match with the original document, our MATCHSUM performance increases as we need to give more consideration to summary-level semantics.

For the results on long summaries, we evaluated on WikiHow, Multi-News, and PubMed and the results are displayed below. Our MATCHSUM still outperformed all baseline models in all datasets. N-gram blocking really hurts the performance of BERTEXT on PubMed as we believe the technique doesn't lead to understanding of semantics of sentences which it's not suitable in scientific domain.



ANALYSIS

We analyse the performance gaps between MATCHSUM and BERTEXT as z increases on XSUM, CNN/DM, and WikiHow. The results are displayed below. The performance gap is the smallest whenever $z = 1$ (best-summary is not pearl-summary). As z increases, the performance gap increases. For both CNN/DM and WikiHow, this increases continues as z gets larger and larger whereas for XSUM, the performance gap increases up to a certain level of z before decreasing. The rationale is left for future exploration. The results showcase that our semantic-based summary-level model can capture sentences that might not be the highest-scoring sentence but it's still important.

The improvement of MATCHSUM is controlled by the inherent gaps between sentence-level and summary-level performance. We measure the improvement made by MATCHSUM over BERTEXT on different datasets and compare that with the inherent gap to create our  ratio. The results are displayed below. As the length of our ground truth summaries increases, it

becomes more difficult to reach the peak performance of summary-level summarisation. ⁷ ^ is illustrated by the small score by PubMed and Multi-News whose summaries has over 200 tokens. Another observation is that when the summary length are consistent and evenly distributed, our model tends to perform better, which explains why Multi-News outperformed PubMed.

Conclusion and Future Work

In the future, we could work on different forms of matching models to further explore the performance of the proposed framework. In addition, a better understanding of the characteristics of the datasets we are dealing with would give us an advantage over which types of models to use.

Source: <https://arxiv.org/pdf/2004.08795.pdf>

Ryan

Data Scientist



[Previous Post](#)

Day 159: NLP
Papers Summary -
ICD Coding from
<Clinical Text Using
Multi-Filter
Residual
Convolutional
Neural Network

Next Post



Day 161: NLP
Papers Summary -
BLEURT: Learning
Robust Metrics for
Text Generation

[Data Science](#) [Natural Language Processing](#) [NLP Papers Summary](#)

Day 365: NLP Papers Summary – A Survey on Knowledge Graph
Embedding



Ryan

30th December 2020



[Data Science](#) [Implementation](#) [Natural Language Processing](#)

Day 364: Ryan's PhD Journey – OpenKE-PyTorch Library Analysis + code snippets for 11 KE models

Ryan

29th December 2020





[Data Science](#) [Ryan's PhD Journey](#)

Day 363: Ryan's PhD Journey – Literature Review – Knowledge Acquisition – 1st Passes XIV

Ryan

28th December 2020

