CIS5300 Project MS2

Wanqing Ding (dwanqing), Eileen Feng (yazhefen),
Crescent Xiong (zihanx3), Leila Zhu (leilazhu)

December 9, 2023

1 Evaluation Metric

1.1 ROUGE-N

We chose a widely used evaluation metric for text summarization in the literature [1] - ROUGE score. It evaluates the overlap of n-grams (contiguous sequences of n items, usually words) between the generated summary and the reference summary:

$$\text{Precision}(n) = \frac{\text{Overlap count}}{\text{Number of Candidate n-grams}} \tag{1}$$

$$\text{Recall}(n) = \frac{\text{Overlap count}}{\text{Number of Reference n-grams}} \tag{2}$$

$$\text{ROUGE-N F1-Score}(n) = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision}(n) + \text{Recall}(n)} \tag{3}$$

A ROUGE-1 score of 1.0 indicates that the generated summary perfectly matches the reference summary in terms of unigram overlap. And scores near 0 indicate poor performance, suggesting little to no overlap between the generated and reference summaries. In general, a ROUGE-1 score of 0.5 is considered good.

1.2 Cosine Similarity

Since ROUGE-N measures the overlap between the N-grams of the two text, a summary automatically summarized from input document would only be deemed "good", i.e. receiving a high ROUGE-N score, would need to cover exact segments from true summary as many as possible. However, summarizing a text cannot be evaluated the same way as doing a math problem. There are more than one correct answers. Human languages are also powerful and flexible. The same meaning can be expressed using various terms. Even though the summary from extractive approach is originated from the input text, the authors themselves could use different words interchangeably to express the same meaning. Therefore, we introduce a second metric for better evaluation purpose with more fairness: the **cosine similarity** [2], which measures the similarity between two vectors. Cosine distance is the opposite, measuring how different the two are.

$$\text{Cosine Similarity} = \frac{A * B}{\|A\|\|B\|} = 1 - \text{Cosine Distance} \tag{4}$$

2 Simple Baseline: LEAD3

The LEAD3 baseline [4] is a straightforward approach that selects the first three sentences from the input text as the summary. The Python function lead3 implements this method by splitting the text into sentences and selecting the first three. While simplistic, LEAD3 serves as a simple baseline to compare against more sophisticated summarization methods.

3 Strong Baseline: TextRank

From [1], we know that the steps of an extractive method to summarize a document are the following:

i. Construct an intermediate representation of the document;

ii. Score the sentences based on their representation;

iii. Construct a summary composed of selected sentences.

There are several approaches to transform input text as vectors, i.e. representations, and even before that, we need some preprocessing for the raw text.

3.1 Data Preprocessing

The input text is a document. The first step is to use a sentence tokenizer from nltk to split it into sentences. Next, we use a simple and effective way as preprocessing: to replace every non-character with a space and convert every valid character to its lower case. Some advanced ways could be using a lemmatizer like wordnet or a stemmer to find the root of input words, but all these come with an increased cost of run time. We will discuss later about the issue of run time later in Section 4.3.

3.2 TextRank Model

To construct an intermediate representation, we count the number of occurrences for words, excluding stop words, in each sentence for an input document. Then, we measure the cosine similarity between every pair of sentences within the document, which would be used as input to compute the scores of sentences through TextRank algorithm.

Similar to PageRank, TextRank [1, 3] is a graph-based ranking algorithm using iterative method to compute the scoring. Specifically in text summarization, the nodes are the sentences, and the weight of an edge is the similarity scores between the two nodes connected by this edge.

Let $G = (V, E)$ be a directed graph with the set of vertices V and edges E. $E(V_i, V_j)$ represents a directed edge from V_i to V_j. We can iteratively compute the weight of the edges through Eq.5, where $d \in [0, 1]$ is a damping factor. Here we keep $d = 0.85$ in our experiments. This iteration would stop until the change of the weights are within some tolerance or reaching the maximum number of iterations (default $max\_iter = 100$).

$$S(V_i) = (1 - d) + d * \sum_{j \in E(\cdot, V_i)} \frac{1}{|E(V_j, \cdot)|} S(V_j) \tag{5}$$

Eventually, we are able to retain a weight for every sentence and sort them in descending order. We decide to select sentences that achieve top-3 scores as our summary for the input text.

4 Discussion

4.1 ROUGE-1

From Table 1, we can observe that TextRank score is generally lower than LEAD3. As news datasets tend to narrate main points at the beginning of most documents, which we also observe from our analysis of data in Figure 1 from Appendix, LEAD3 can easily be a relatively 'strong' baseline model when we use ROUGE-N metric, which measures exact matches through n-grams, to evaluate. TextRank, on the other hand, can generalize better to documents that do not necessarily have the main content summarized at the beginning.

Model	Train	Validation	Test
Simple Baseline: LEAD3	0.187, 0.220, 0.187	0.267, 0.353, 0.291	0.257, 0.356, 0.287
Strong Baseline: TextRank	0.157, 0.282, 0.194	0.178, 0.293, 0.213	0.172, 0.297, 0.210

Table 1: ROUGE-1 Evaluation Results [Precision, Recall, F1 Score]

4.2 Cosine Similarity

From Table 2, we can observe that compared with simple baseline LEAD3, TextRank is generally good by surpassing the baseline in training by 10%. Because there's indeed no learning in the training set, we can evaluate all three sets equally. Since we have 287k samples in training while around 13k and 11k in validation and test sets, respectively. We can observe an overall increase of 0 in cosine similarity brought by TextRank over LEAD3.

Model	Train	Validation	Test	All
Simple Baseline: LEAD3	0.590	0.703	0.698	0.599
Strong Baseline: TextRank	0.687	0.705	0.701	0.688

Table 2: Cosine Similarity Evaluation Results

4.3 Runtime Considerations

We experience abnormal runtime in evaluating the training data due to its large size. This limits our preprocessing steps since using lemmatizer or stemmer also increase the runtime a lot. Besides TextRank, we also attempted

using TF-IDF to either vectorize the sentences or scoring them. Unfortunately, this turned out not doing very well. Evaluating training wouldn't stop within 7 or 8 hours.

4.4 Lessons Learned and Next Steps

The lack of LEAD3 and TextRank is pretty obvious: they both cannot take advantage of the training data but apply the same method to all the data. Learning from training data can benefit models a lot if we are using neural networks, especially when we have so many training samples. Extractive methods solely have bounded performances in generalizing summaries since it can only select whole sentences, while summaries are often composed of part of sentences or even include words not appearing in the sentences. Abstractive model using deep learning can compensate both shortcomings we just listed, which is the next step we are heading to.

5 Appendix

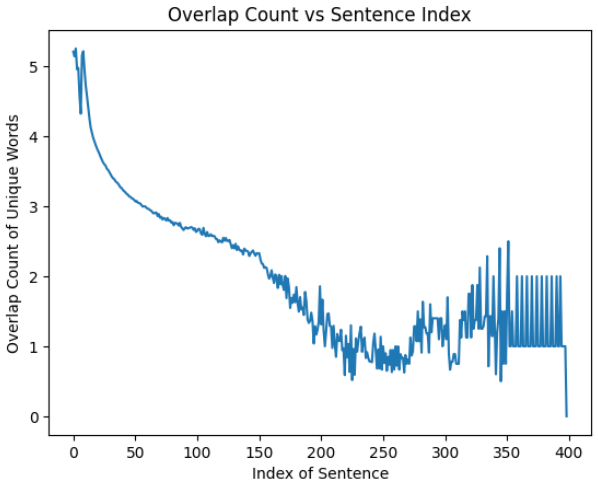

Figure 1: Average Count of Unique Word Overlap in the Train Set

References

[1] Mehdi Allahyari et al. *Text Summarization Techniques: A Brief Survey*. 2017. arXiv: 1707.02268 [cs.CL].

[2] Daniel Jurafsky and James H. Martin. "Chapter 6. Vector Semantics and Embeddings". In: *Speech and language processing*. Pearson, 2014.

[3] Rada Mihalcea and Paul Tarau. "TextRank: Bringing Order into Text". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Ed. by Dekang Lin and Dekai Wu. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 404–411. URL: https://aclanthology.org/W04-3252.

[4] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. *SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents*. 2016. arXiv: 1611.04230 [cs.CL].