Explore Unsupervised Sentence Summarization : *NewBottleSum* Model

Kai Liao
Courant Institute of
Mathematical Sciences
New York University
kl3199@nyu.edu

Qucheng Peng
Tandon School
of Engineering
New York University
qp276@nyu.edu

Qiyu Xiao
Courant Institute of
Mathematical Sciences
New York University
qx344@nyu.edu

Abstract

In this work, we propose an unsupervised extractive sentence summarization method ***NewBottleSum*** which is inspired by ***BottleSum*** model and Information Bottleneck principle. To overcome the limitations of the original model, which only uses relation to next sentence as relevance measure of the summary, we further include the relevance between summary and article's title in our new model. A new objective function is thus proposed and we implement an concrete algorithm based on it. Our method is evaluated on partial DUC-2003 and DUC-2004 datasets with ROUGE and significantly outperforms the original ***BottleSum*** model. We also finish a human evaluation based on a small dataset and ***NewBottleSum*** performs well on it. Finally, we conclude our work and point out our model's drawbacks and directions for future work.

1 Introduction

Recent developments in sentence summarization depends heavily on the application of neural networks, such as supervised extractive models (Rush et al., 2015; Li et al., 2017a) that reach state of art performance on some common datasets. However this kind of approach requires intensive computation and more importantly, reliable human written gold summaries which are not easy to obtain. Large datasets that will be too costly to have human summaries are more common cases in the field and this limits the extension of these model's further application. Therefore with less requirement on reference summaries, unsupervised extractive summarization models (Fevry and Phang, 2018; Baziotis et al., 2019) are still very competitive in NLP research.

The autoencoder-based approaches of unsupervised model mentioned above focus on optimizing the reconstruction loss between original sentence and the compressed sentence. But this doesn't necessarily retain the main information of a sentence. ***BottleSum*** (West et al., 2019) model tries to resolve this issue by applying Information Bottleneck principle (Tishby et al., 1999). Instead of evaluating a summary by if it's easy to recover the full original sentence, ***BottleSum*** focuses on retaining relevant information, which is the continually discussed information in following sentences. This is a more natural way to summarize when relating to human's writing habit.

However, the original ***BottleSum*** approach may be limited, especially when the next sentence is turning to a new topic. In fact, we think there may be more relevant information in the titles, which are often used as summary-like text in NLP tasks when real summaries are not available. When the next sentence may not always be a significant part in the whole text, if the method relies solely on information from next sentence, the final result may loss some meaningful words and prominent information. As we show in Figure 1, with only the influence from next sentence, the ***BottleSum*** summary deletes anything after the conjunction *although* and twists the meaning of the sentence, while our summary keeps it.

In this work, we try to combine usage of titles into the original ***BottleSum*** model and investigate the possible improvement by testing it on DUC-2003 and DUC-2004 with ROUGE.

2 Method Description

2.1 Theory

The core of the ***BottleSum*** method is the Information Bottleneck principle (Tishby et al., 1999). This principle is mainly about compressing a sentence that can best preserve some relevant information. ***BottleSum*** model considers the notion of information relevance as information persisting in the next

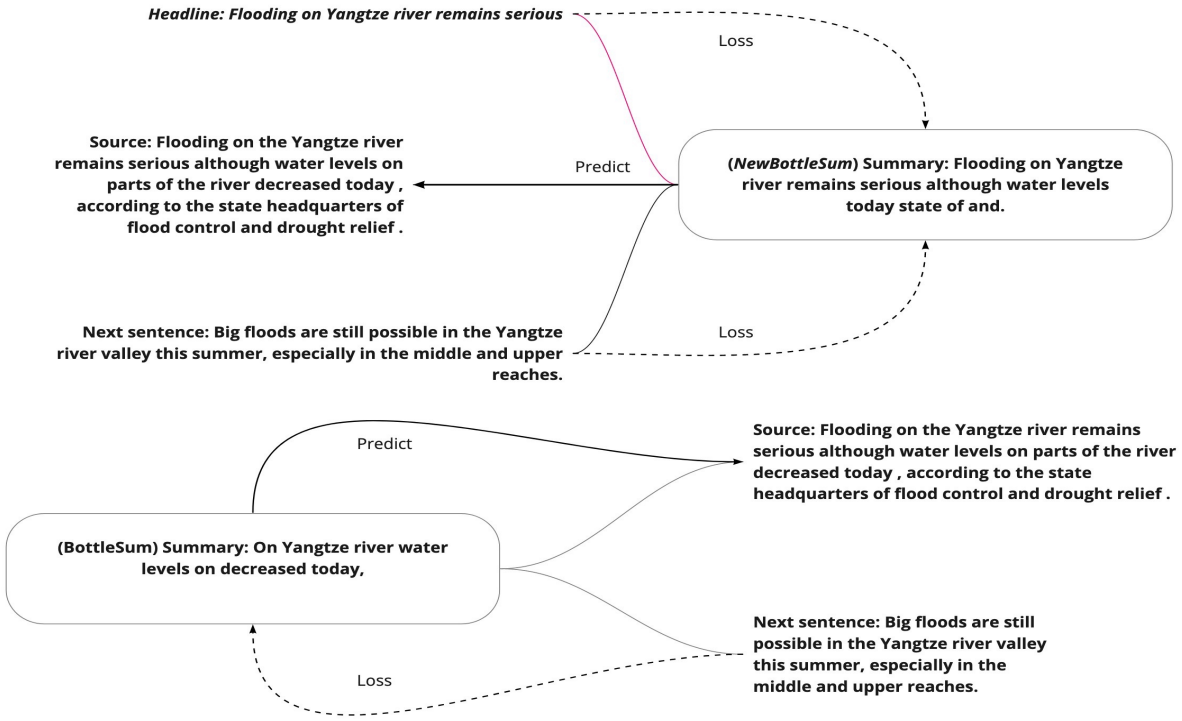

Figure 1: Comparison between original ***BottleSum*** and ***NewBottleSum*** that includes headline in the input

sentence.

We can also illustrate our motivation in a theoretical way. The origin aim of Information Bottleneck Theory is to detect and estimate signals better (Tishby et al., 1999), and that remains a popular research area in the signal processing field (Jaynes, 1957). For detection, we need filter to highlight the information we need. For estimation, we use estimator to predict the information we expect. During filtering, the next signal is easier to interfere. During prediction, the previous signal plays a more important role (Poor, 2013). Thus, we can think of ***BottleSum*** treating the summarization task as a filtering problem. On the other hand, ***NewBottleSum*** also regards it as a prediction problem. Therefore, headline matters since it appears before our target sentence to be summarized and has contributions to the prediction aspect.

In the light of this idea, our model extends the ***BottleSum*** by encoding an additional relevance term which is title. It is mathematically equivalent to optimizing the following loss function

$$I(s', s) - \beta_1 I(s', s_1) - \beta_2 I(s', s_2) \qquad (1)$$

where I is the point-wise mutual information function ($I(p, q) = E_{a,b} \frac{p(a,b)}{p(a)p(b)}$), s' is the target summary, s is the source sentence, and s_1, s_2 are the relevant information. In the original model, only first two terms exist and s_1 is defined as the next sentence. Here we add the third term and use titles as s_2. The first term of this loss function is minimized when we prune the source sentence and the latter two terms are minimized when we keep overlap information. β_1, β_2 are the weights representing how much we encourage the result to contain relevant information.

2.2 Algorithm

The main idea of the algorithm is applying the equivalence between mutual information function and cross entropy when s' is given. Namely, when summary s' is fixed, the loss function (1) is equivalent to

$$\log \frac{1}{p(s')} - \sum_i \beta_i p(s_i|s') \log p(s_i|s') \qquad (2)$$

(detailed derivation can be found on West et al. (2019)).

A detailed algorithm is attached. We design the outer loop by removing a single word from the source sentence s gradually in order to generate summary candidates. As shown in line 3, the outer loop terminates when the length of compressed sentence is 1. In each iteration, two important operations are processed. The fist one is to filter all the

candidates by their pruning term. We set a upper bound for it (line 2), which is the mutual information of the original sentence itself. All the sentences with a pruning value lower than the bound are able to pass the filter. The filtering process (line 4) ensures that our summary is compressed enough. The second one is to sort all the candidates by their relevance score in an increasing order. Our default $\beta_2 : \beta_1$ setting is $1 : 1$. This is represented in line 5. Then several truncated sentences are considered in line 6-10, which also follow the order of the previous steps. After that, we throw the first k candidates into our candidate list. In order to control the time complexity, we set k as 1. Finally, we pick the optimal candidate as our summary.

Algorithm 1 Variant ***NewBottleSum*** Algorithm

Require: original sentence s, next sentence s_1, title s_2

1: C $\leftarrow$\{s\} ▷ candidate summaries
2: $CE_{threshold} \leftarrow$CE(s)+1 ▷ filter function
3: **for** x in $length(s)$... 1 **do**
4: $C_x \leftarrow$\{s'$\in$ C | $length(s') = x, CE(s', s') < CE_{threshold}$\}
5: Sort C_x by ascending $\beta_1 CE(s', s_1) + \beta_2 CE(s', s_2)$
6: **for** $s' \in C_x$ **do**
7: **for** j in 2...$length(s')$ **do**
8: **for** i in 1...$length(s') - j$ **do**
9: **if** $CE(s'[1 : i-1] + s'[i + j, x']) < CE(s')$ **then**
10: $C \leftarrow C + s'[1 : i-1] + s'[i + j, x']$
11: **return** $\arg\max_{s' \in C} \beta_1 CE(s', s_1) + \beta_2 CE(s', s_2)$

3 Related Works

There have been successful supervised summarization models like ABS (Rush et al., 2015). This model learns a conditional language model that predicts the next word based on source text and context summary. Other autoencoder-based models includes sequence oriented encoder-decoder model (Li et al., 2017b) that contains recurrent trained layers to learn the latent structure of summaries.

The main issue with these supervised abstractive models is that they require a lot of training data with human written summaries, which is counterintuitive as humans do not need to see huge amount of parallel corpus to learn summmarization.

Pointer-based models (Cheng and Lapata, 2016) are proposed for extractive summarization. It combines CNN based encoder and decoder with attention mechanism. This model is trained on CNN/Daily Mail dataset, by which it does not generalize well to other datasets. On the contrary, ***BottleSum*** and our variant don't suffer from the same limitation.

Fevry and Phang (2018) approached fully unsupervised summarization by adding noises to sentences and denoising through minimizing the reconstruction loss. The model preprocesses each training sentence by taking a sub-sample of another sentence and shuffling them on bi-gram/unigram level to produce a new source sentence. The encoder-decoder is then trained end-to-end by minimizing the reconstruction loss.

Baziotis et al. (2019) proposed two chained encoder-decoder pairs (SEQ^3) using gumbel-softmax that makes the model fully differentiable. The model consists of a compressor and a reconstructor where each of them is an encoder-decoder pair. Three additional loss terms are added to the loss function: Language Model Prior Loss, Topic Loss, and Length Penalty, which are used to encourage the model to produce human-readable, source relevant, and strictly shorter summaries respectively.

The main limitation of the encoder-decoder based approaches is that they tend to retain all information in the source text, which arguably goes against the rationale of the Information Bottleneck Principle. Thus, additional human-designed loss terms need to be added to the reconstruction loss (e.g. (Baziotis et al., 2019)).

Recently, Zhou and Rush (2019) introduced a method using a product-of-experts model (Hinton, 2002) that combines contextual matching and domain fluency to generate grammatical summary close to source text. While fluency language model requires pretraining on some dataset to learn probability distribution, ***BottleSum*** and our variant do not require any sort of further pretraining except an existing pretrained language model.

4 Experiments

4.1 Setup

We evaluate our model's performance on DUC-2003 and DUC-2004 datasets (Over et al., 2007). These consists of first sentences from hundreds of news articles and corresponding human written

summaries as reference. For our model, we recover next sentence from original DUC datasets, and the headline of each article using Nexis Uni, a comprehensive document database. The resulting partial datasets consist of 544 and 447 sentences respectively, compared to 624 and 500 sentences in the original datasets, since we could not find some of the headlines. We use ROUGE (Lin, 2004) as metric to compare our result with some of the related works mentioned previously. ROUGE is mainly computing the proportion of overlap units (like n-grams) between model generated summaries and reference summaries. F-measures are reported.

We think the drawback of using fragmented datasets can be negligible as we recovered more than 80% of the original datasets, and the ROUGE scores compared to those reported on full datasets (West et al., 2019) do not differ significantly.

Since ROUGE is not a perfect metric, to address it limitations (Schluter, 2017), we do human evaluation to compare the quality of summaries generated from different models on a small scale as sanity check. We sample 40 output summaries on the DUC-2003 dataset. We employ a similar setting used by West et al. (2019). We have one of our teammate and his friend to evaluate on a pairwise basis over two attributes: coherence and agreement. To compare two summaries, the score is aggregated over a scale: 1 (better), 0 (equal), and -1 (worse). The scores reported are average scores from the two people.

4.2 Results

Method	R-1	R-2	R-L
PREFIX	21.52	6.37	19.01
INPUT	23.26	7.42	18.95
SEQ^3	21.52	6.30	19.05
BottleSum^{Ex}	22.63	6.00	19.87
NewBottleSum^{Ex}	**25.86**	**7.54**	**22.32**

Table 1: Averaged ROUGE on the (partial) DUC-2003

Method	R-1	R-2	R-L
PREFIX	22.23	6.27	19.42
INPUT	24.57	**7.45**	20.04
SEQ^3	21.85	5.95	19.09
BottleSum^{Ex}	22.35	5.47	19.55
NewBottleSum^{Ex}	**25.25**	7.19	**21.59**

Table 2: Averaged ROUGE on the (partial) DUC-2004

For both datasets, we include major unsupervised baselines: PREFIX and INPUT, first 75 bytes and full input sentence as baselines, SEQ^3, and the original ***BottleSum***. We compare them on the partial DUC datasets we obtain.

We find that ***NewBottleSum*** achieves highest R-1 and R-L scores on both datasets. It further justifies the effectiveness of Information Bottleneck Principle in unsupervised setting. ***NewBottleSum*** has lower R-2 score than INPUT sentence on the DUC-2004 dataset. We think that is possibly due to the extractive nature of our model, since INPUT sentence is more semantically fluent by copying human written text directly. The performance of ***NewBottleSum*** surpasses original ***BottleSum*** by a significant margin. It shows that there is a benefit in encoding more appropriate relevance terms, comparing to ***BottleSum*** where only next sentence is used.

Model	Comparison	coherence	agreement
NewBottleSum^{Ex}	SEQ^3	0.26	0.43
	BottleSum^{Ex}	0.2	0.23

Table 3: Human evaluation on 30 DUC-2003 sentences (pairwise comparison of model outputs).

For qualitative evalution, we find that ***NewBottleSum*** generates more preferable summaries than SEQ^3 and ***BottleSum***. It shows that encoding additional appropriate relevance term both leads to better performance in automatic evaluation and produces more favorable summaries for human audience.

5 Conclusion

We present ***NewBottleSum***, extending on ***BottleSum*** by encoding an additional relevance term. It is an extractive unsupervised method which can be used without any training except a pretrained language model, and available next sentence and headline. It outperforms previous unsupervised baselines and ***BottleSum*** on automatic ROUGE evaluation by a significant margin. Since the original ***BottleSum*** has already surpassed other autoencoders-based approaches, our empirical results show that encoding an additional appropriate relevance term leads to better automatic performance, and further justify Information Bottleneck principle as a promising direction for progress on sentence summarization.

Reproducibility

All code and data are available here: https://github.com/davidpqc1231/NewBottleSum

Collaboration Statement

All team members contributed equally in building model, recovering datasets, running experiments, and writing the paper.

References

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 673–681, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Thibault Fevry and Jason Phang. 2018. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422, Brussels, Belgium. Association for Computational Linguistics.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800.

Edwin T Jaynes. 1957. Information theory and statistical mechanics. *Physical review*, 106(4):620.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017a. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100, Copenhagen, Denmark. Association for Computational Linguistics.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017b. Deep recurrent generative decoder for abstractive text summarization. *arXiv preprint arXiv:1708.00625*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

H Vincent Poor. 2013. *An introduction to signal detection and estimation*. Springer Science & Business Media.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Naftali Tishby, Fernando C Pereira, and William Bialek. 1999. The information bottleneck method. *In Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. Bottlesum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Jiawei Zhou and Alexander Rush. 2019. Simple unsupervised summarization by contextual matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy. Association for Computational Linguistics.