An Enhanced MeanSum Method For Generating Hotel Multi-Review Summarizations

Author: Saibo Geng
Supervisor: Diego Antognini, Prof. Boi Faltings
Lab: Artificial Intelligence Lab(LIA)

EPFL, Switzerland

Abstract

Multi-document summarization is the process of taking multiple texts as input and producing a short summary text based on the content of input texts. Up until recently, multi-document summarizers are mostly supervised extractive. However, supervised methods require datasets of large, paired document-summary examples which are rare and expensive to produce. In 2018, an unsupervised multi-document abstractive summarization method(Meansum) was proposed by Chu and Liu, and demonstrated competitive performances comparing to extractive methods. Despite good evaluation results on automatic metrics, Meansum has multiple limitations, notably the inability of dealing with multiple aspects. The aim of this work was to use Multi-Aspect Masker(MAM) as content selector to address the issue with multi-aspect. Moreover, we propose a regularizer to control the length of the generated summaries. Through a series of experiments on the hotel dataset from Trip Advisor, we validate our assumption and show that our improved model achieves higher ROUGE, Sentiment Accuracy than the original Meansum method and also beats/comparable/close to the supervised baseline.

Keywords

Automatic summarization, Interpreatability of Deep Learning, Multi-Aspect Masker, Meansum, Deep Learning, Neural Network, Abstractive Summarization

I. INTRODUCTION

Text automatic summarization is a challenging task in NLP and it’s also among the most focused research topics recently. In industry, text automatic summarization has various application, in addition to the direct generation of text such as generating headlines, it can also play an important role in other NLP tasks as intermediate step. For example, in text sentiment analysis, search engine and recommendation system etc, comparing to use original text, employ a summarized text could enhance the performance without losing much information.

Depending on the method of summarizing, a summarizer can be either extractive, abstractive. Extrative method explicitly uses text segment(sentence level or word level) from input texts to construct the output summary while an abstractive method produces novel text which contains the essential information of input and avoid repeating the texts from input document. Abstractive summarization has been studied using neural sequence transduction methods with datasets of large, paired document-summary examples. However, such datasets are rare and expensive to produce. Recently, some progress has been made in unsupervised abstractive learning. Meansum, an unsupervised multi-document abstractive summarization method proposed by Chu and Liu in 2018 demonstrated competitive performance on multiple metrics with extractive methods. Meansum model takes the similarity between multiple input reviews and generated summary plus an encoder-decoder as objective function. It does not rely on any specific features of given dataset. However, meansum is highly abstractive but not takes into account any information on review’s content. Besides, evaluations on output summaries show that meansum is biased towards high precision and low recall. To address the first issues, we suggest to use masks instead of original reviews as input, in order to reduce the unnecessary information and guide the summarizer. To address the second issue, we suggest a regularizer to constraint the output summaries’ length. Despite the fact that our masks help the meansum model to achive better performance, our study shows it could increase the difficulty of training.
II. METHODS

A. Meansum

We chose Meansum because it’s the latest unsupervised highly abstractive multi-reviews summarization model and demonstrates performance comparable to extractive methods on their validation dataset. Meansum model is composed of two main components: (1) an auto-encoder module that learns representations for each review (in case of no pretrained language model) and guarantees the fluency of generated language, and (2) a summarization module that learns to generate summaries that are semantically similar to each of the input reviews. These two modules contribute a reconstruction loss and similarity loss, respectively. Both components contain an LSTM encoder and decoder – the two encoders’ weights and the two decoders’ weights are tied by default.

The Autoencoder Reconstruction Loss function is a cosinus similarity between the input texts and reconstructed texts (texts can be reviews, masks, or RSAR). The Summary-Review Similarity Loss is defined as the average cross entropy between the generated summary and each input text. The Summary-Review Similarity Loss is the key how Meansum achieves unsupervised learning.

B. Masks of reviews

The term ‘mask’ or ‘rationale’ is generally understood as a (long) word sequences from the input text, which suffice for neural network system to make the prediction. We can consider masks as informative text within reviews, and the non mask part of reviews are little informative. Having masks enabled us to implement a content selection on reviews before feeding them to neural network. A first usage of masks is to filter noises from original reviews. A second usage is to aggregate review information by categories, this could help the summarizer

We use the Multi-Aspect Masker(MAM) to build masks on the hotel review dataset [Ant19]. MAM allocates to each review segment a most related aspect from five candidates:

1) aspect0.service,
2) aspect1.cleanliness,
3) aspect2.value,
4) aspect3.location,
5) aspect4.room.

Masks were used in three different ways in this project:

- **Mask** Concatenate all masks as filtered review
- **RSAR** Split all reviews into single aspect segments and regroup them by the associated hotel. Train a single summarizer for the regrouped summaries.

Fig. 1: Meansum

Fig. 2: Regrouped Single Aspect review
• **RSAR I-V** Use the same regrouped reviews as in the previous case, but classify them by aspect and for each aspect train an individual summarizer.

It is critical to note that masks are not homogeneously distributed over different aspects. In particular, Location(Aspect3) has far more reviews than any other aspects (more than half of the whole RSAR dataset), which outlines that customers attach most importance to this aspect when evaluate a hotel. Clealiness(aspect1) and values(aspect2) have less than 1/8 of the whole RSAR dataset’s reviews.

| Dataset | review quantity |
|---------|-----------------|
| RSAR    | 3 174 386       |
| RSAR 3  | 1 537 321       |
| RSAR 0  | 609 577         |
| RSAR 4  | 459 777         |
| RSAR 2  | 334 332         |
| RSAR 1  | 233 379         |

**TABLE I: reviews distributions on aspects**

![Fig. 3: RSAR I-V](image)

**C. Controlling summary length**

This capability of controlling generated text’s length is crucial for various NLP applications notably text summarization. In the context of text summarization, generation of headlines requires a concise summaries with a short length while in legal contract analysis, a summary potentially longer and with high recall is targeted. Length control method for extractive summarization has been investigated by Mehdad [MST+16]. A recent review of the literature on this topic found that Kikuchi [KNS+16] suggested 4 methods of controlling text encoder-decoder system’s output length. In this work, we suggest to control output summary length by adding a length loss term into the objective function of Meansum. To this end, we propose a concept of text *shortness*.

Intuitively, the shortness of a summary is characterized by the length of the summary. But as the loss function built upon this definition is not differentiable, it’s not a good definition in deep learning. We use the average probability of giving </doc> as next token during the whole text generation process as the definition of shortness. In the extreme case where the text generated is very short, *shortness* will be high and in the case where the text generated is very long *shortness* will necessarily be small as it’s not sampled through the whole sampling processes. In particular, we notice that *shortness* is 0 for summaries reaching the maximal length. Here we give an example of the generated summary for readers to better understand this concept:
"I have stayed at the hotel before and it was a lovely experience. It is a very old hotel but the rooms are a little dated and could do with a bit of a facelift. The staff were all very helpful and friendly, nothing was too much trouble. We didn’t use the restaurant but the food was good value for money."

Once we have defined the shortness we can use it to calculate the shortness of input reviews and the shortness of generated summary. The Length loss function is defined as below:

\[ L_{\text{len}} = \alpha || \beta \bar{S}_i - S_s ||^2 \]

\( \alpha, \beta \) are two coefficients used to regularize the effect of length constraint. In case where \( \beta = 1 \), the length constraint is set to be close to the mean length of input reviews. A larger \( \beta \) will lead to a longer summary and a small \( \beta \) to a shorter summary.

D. Supervised baseline

We also designed a supervised model to compare with our model. As showed in the Figure 3, we replace the Autoencoder Reconstruction Loss by a standard cross entropy loss between gold short summaries and generated summaries. The final loss we optimize is simply \( L_{\text{supervised}} = L_{\text{autoencoder}} + L_{\text{crossentropy}} \). We use the summarizer trained on masks, reviews and RSAR to generate summary on our validation set. No tuning was performed—we used the exact same model as in the previous part. After that, we train on each dataset a supervised model and test them on the same validation set as for unsupervised model to compare the performance.

Fig. 4: supervised model

III. Metrics and Evaluation

A. Automated Metrics Without Reference Summaries

Here we use the same metrics as proposed in the Measum paper [CL18a]

- **ROUGE**

Rouge score [Lin04] is a common automated metrics in text summarization. As we want to evaluate our summarizer without reference summaries, a variation of rouge score is adopted. Instead of calculating rouge score between reference summary and generated summary, we do the same calculations between each source review and generated summary (also called WordOverlap in Measum paper). Finally we take the average of these score and define it as Rouge score*.

\[ \text{ROUGE}_i = \frac{1}{N} \sum_{i=1}^{N} \left[ \frac{1}{k} \sum_{j=1}^{k} \text{ROUGE}_i(s^{(i)}, x_j^{(i)}) \right] \]
where \( k \) is the number of reviews being summarized (\( k=8 \) in our experiments). In this paper, we used ROUGE*-1, ROUGE*-2 and ROUGE*-L.

- **Sentiment accuracy**
  A useful summary should reflect and be consistent with the overall sentiment of the reviews. We first separately train a CNN-based classification model that given a review \( x \), predicts the star rating of a review, an integer from 1 to 5. For each summary, we check whether the classifier’s predicted rating is equal to the average rating of the reviews being summarized (rounded to the nearest star rating). \( [CL18b] \)

\[
\frac{1}{N} \sum_{i=1}^{N} CLF(s(i)) = \text{round} \left( \frac{1}{k} \sum_{j=1}^{k} r(x_j^{(i)}) \right)
\]

- **NLL (Negative Log Likelihood)**
  We use NLL to measure the fluency of generated text. A low NLL reflects high language fluency in the generated text.

**B. Automated Metrics with Reference Summaries**

- **ROUGE**
  To compare our model with supervised baseline, we use the reference summary to calculate the ROUGE of generated summary. The ROUGE used here is identical as in the one widely used.

**IV. EXPERIMENTAL SETUP**

**A. Datasets description**

To evaluate the performance of our model, we crawled 140k hotel reviews from TripAdvisor. Each review contains a five-star rating for each aspect: Service, Cleanliness, Value, Location, and Room. The average correlation between aspects is high (63.0% on average).

| Dataset | Hotel review Dataset |
|---------|----------------------|
| Number of reviews before filtering | 2 222 273 |
| Number of reviews after filtering | 1 291 638 |
| Number of hotels before filtering | 38 903 |
| Number of hotels after filtering | 26 780 |
| Maximal review length | 837 characters |
| Average of review length | 467.8 characters |

**TABLE II: Statistics on hotel review dataset**

It’s beneficial to compare our model with a supervised model. To enable the training of a supervised model, we scraped 83 850 hotel information, including a short overall summary and multiple summaries on various aspects from trustyou.com. Among the 83 850 hotels, 25090 of them are also present in our hotel review dataset. We take the intersection of the two dataset and use it for supervised learning.

It is important to note that gold summaries are much shorter compared to reviews. A gold summary typically looks like this: "Excellent beach hotel. Close to the beach. Great rooms and fantastic service. Beautiful beach.", while the review generated by Meansum are in general much longer.

**B. Experimental Details**

**V. RESULTS**

*In subword
### TABLE III: Statistics on hotel with ground true summary dataset

| Dataset                          | Supervised Dataset |
|----------------------------------|--------------------|
| Average short gold summary length| 83 characters      |
| Average long gold summary length  | 663 characters     |
| Number of reviews before filtering| 1,524,907         |
| Number of reviews after filtering| 871,330           |
| Number of hotels before filtering| 25,090            |
| Number of hotels after filtering | 17,221            |
| Average of review length         | 851 characters     |

### TABLE IV: Automated metric results with k = 8 reviews being summarized

| Input reviews | Rouge-1 | Rouge-2 | Rouge-L | Sentiment Accuracy | NLL   | mean summ len | Comment               |
|---------------|---------|---------|---------|--------------------|-------|---------------|-----------------------|
| review        | 0.313   | 0.0441  | 0.167   | 0.669              | 1.16  | 87.3          | /                     |
| masks         | 0.312   | 0.0556  | 0.180   | 0.639              | 1.01  | 56.8          | /                     |
| RSAR          | **0.315** | 0.0639 | 0.191   | **0.685**          | 1.14  | 60.6          | Best overall          |
| RSAR-0        | 0.297   | 0.0658  | 0.202   | 0.783              | 1.00  | 49.0          | repetitive            |
| RSAR-1        | 0.285   | 0.0903  | 0.220   | 0.715              | 0.879 | 42.0          | highly repetitive      |
| RSAR-2        | 0.297   | **0.431** | 0.436  | **0.884**          | 4.37  | /             | lr too large          |
| RSAR-3        | 0.310   | **0.0660** | **0.201** | 0.633            | 1.10  | 68.1          | /                     |
| RSAR-4        | 0.284   | 0.0815  | 0.104   | 0.613              | 1.04  | 43.2          | highly repetitive      |

#### A. Main Results

The automated metrics for our model and the baselines are shown in Table III. Using the test split of each input review dataset, our experiments confirm that using RSAR as input text provided a boost to Meansum compared to feeding original reviews on automatic metric evaluations.  

As we mentioned in the dataset description section II-B, masks are not homogeneously distributed over different aspects.² We find empirically that the dataset size is correlated with the repetitiveness of generated summaries. In case of small training dataset such as RSAR1 and RSAR4, the output summaries are highly repetitive despite good Rouge scores and NLL. Based on this consideration, we only consider "review", "masks", "RSAR", "RSAR-3" and "RSAR-0" as robust input review datasets. Among all these datasets, RSAR outperforms in Rouge-1, Sentiment Accuracy³ and gives second best result in Rouge-2 and Rouge-L while RSAR-3 outperforms on Rouge-2, Rouge-L.  

Although RSAR-3 and other single aspect dataset fails to beat the performance of RSAR on Rouge-1, we believe this is due to the huge difference on size of dataset. It could be imagined that given similar size of data, single aspect summarizer would outperform multi-aspect summarizer(RSAR) on other metrics.  

The apparent repetitiveness on the summary of RSAR-1 and RSAR-4 is mostly due to the lack of data. We aware that the inhomogeneity of review number of different aspects is intrinsic and it reveals the priority that customers give while making reviews on the hotel. A direct solution to this would be shorten the output summary. A simple but brutal method that we tested was to truncate the output summary after a given number of sentences. This method did solve the repetitiveness issue and the output summary are quite fluent. Unfortunately, as the summary truncated are very short, the metric on it has very low recall and a very high precision. Thus brutal truncation is not satisfactory and needs to be improved in the future.

#### B. Issue with language fluency

The introduction of mask into summarization model aims to guide the summarizer model to focus on the important, meaningful part of the input reviews. However, one direct counter-effect is the regrouped masks lack language fluency.

²The Rouge score used in this evaluation is different from the traditional one as Meansum it’s an unsupervised learning without true summary.  
³cf evaluations section

⁴The highly biased sentiment distribution in RSAR-1 undermines its high sentiment accuracy
fluency. In contrary to reviews written by humans, masks are generated by concatenating the selected segments of corresponding reviews. This direct concatenation leads to a fragmented language style. This counter-effect is more significant on RSAR as reviews are split into more fine-grained pieces and regrouped together. In the first example below, there two places where the mask lost language fluency due to the sentences discarded. In the second example above, the first phrase is not semantically clear as we do not know who the ‘he’ refers to. And this may confused language model as in real texts, a undefined ‘he’ rarely appears as the beginning of the text. The 2nd and 3rd sentences have the same subject which are syntactically repetitive. This happens from time to time, adding difficulties for model to capture the right syntax and semantics.

**Example 1**

**Original review:**

“Check-in was quick and Jessie (spelling) was very friendly. She explained everything and went out of her way to check on things for us. The room was nice, but older. The beds were metal frames (weight rating unknown...) with nice mattresses. The headboards...interesting and had reading lights attached. The room was clean...until day 2 when we had a roach wave hello from the side table. We killed it and forgot about it. Then the last day, we saw more. In all stages of growth, from nymph to full adult. One had even gotten into the fridge. So, we were happy to be leaving. The A/C unit worked well. The pool hours were enforced and the place was quiet during quiet time. Wifi worked good enough. Nice sized TV with an okay line up. Parking was a pain in the butt...”

“taken from 330th review in hotel test dataset: HotelReview – g34225 – d84403 – Reviews – QualityInnFloridaCity – FloridaCityTypeFlorida.html

**Masks concatenated:**

“check - in was quick and jessie ( spelling ) was very friendly . she explained everything and went out of her way to check on things for us . the room was nice , but older . the room was clean ... until day 2 when we had a roach wave hello from the side table . we killed it and forgot about it . then the last day , we saw more . in all stages of growth , from nymph to full adult . one had even gotten into the fridge . so , we were happy to be leaving . the A/C unit worked well . the pool hours were enforced and the place was quiet during quiet time . wifi worked good enough . nice sized tv with an okay line up . parking was a pain in the butt ...”

**Example 2**

**RSAR:**

‘he was very friendly to the point of sleeping on our bed when we fell asleep . the young man who greeted us was friendly , courteous , helpful and very accommodating . the young man was not so pleased and mentioned that they had been looking for the cat all day . there were a lot of nice touches , such as a happy hour with complimentary cheese and wine each day - which we sampled on our first night . in the afternoons there was complimentary cake a beverage available , but we did not try it .’

“taken from 201th entry from RSAR test dataset: aspect 0 Review – g1006175 – d653093 – Reviews – RoxbroHouse – WorkworthAmbleNorthumberlandEngland.html

Further observations on the tensorboard for language models confirmed with our initial findings. We notice that language model with the same configuration has higher loss on hotel masks after a given number of batches. Contrary to expectations, the language model loss function on RSAR are lower than both hotels and masks, which could be due to the fact that RSAR texts are single aspect thus helps to reduce the difficulty of predicting next word given
previously stated.

\( \text{(a) LM Loss function on Hotel mask} \)

\( \text{(b) LM Loss function on Hotel review} \)

Fig. 5: Loss function curve for reviews and masks

\section*{C. Unbalanced recall and precision}

Interestingly, for all input review datasets, we find that Rouge score are always biased towards high recall and low precision. Below is the example of RSAR. As precision and recall are related to the summary length, longer the summary is, higher its recall could potentially be and lower its precision could be. This is not always true, especially when the summary are just recopying its self, then no matter how long the summary it’s, its recall and precision both remain the same. But if we can control the output summary length by not just repeating, it may allow us to improve the the Rouge F1 score by balancing the precision and recall.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
Rouge & F1 & Precision & Recall \\
\hline
Rouge1 & 0.316 & 0.348 & 0.313 \\
\hline
Rouge2 & 0.0632 & 0.0696 & 0.0628 \\
\hline
RougeL & 0.189 & 0.208 & 0.189 \\
\hline
\end{tabular}
\caption{Detailed rouge scores of RSAR}
\end{table}

To address this issue, we introduced length loss and tested it with different parameter values.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
model & ROUGE*1F & ROUGE*1R & ROUGE*1P & mean summ len(#char) & comment \\
\hline
RSAR-1(\(\alpha = 1e0\)) & 0.276 & 0.227 & 0.536 & 36.0 & / \\
\hline
RSAR-1(\(\alpha = 1e1\)) & 0.268 & 0.222 & 0.535 & 37.6 & / \\
\hline
RSAR-1(no length loss) & 0.274 & 0.232 & 0.528 & 40.7 & baseline \\
\hline
RSAR-1(\(\alpha = 1e2\)) & 0.271 & 0.231 & 0.518 & 41.0 & / \\
\hline
RSAR-1(\(\alpha = 1e3\)) & 0.258 & 0.238 & 0.453 & 57.0 & / \\
\hline
\end{tabular}
\caption{Effect of length loss parameter \(\alpha\)}
\end{table}

The results, in particular the column of mean summary length confirmed our hypothesis upon the regularizer effect of length loss. Both \(\alpha, \beta\) can regularize the output summary length. In addition to this, we notice the recall and precision were balanced by the variation of summary length, as in TABLE VI and TABLE VII, recall grows and precision keeps decreasing. This remarkable result shows that the effect of length loss is not simply repeating but generating new content. Unfortunately, we did not see any improvement on ROUGE*F. This is due to the fact that the decreasing rate of precision is faster than the increasing speed of recall, thus the gain on recall is
not enough to fill the loss on precision. Further fine tune of parameters such as learning rate might resolve this issue. Besides, the control on the generated summaries length is only in a qualitative manner. We could not constraint the length of generated summaries into a given interval.

\textbf{D. Supervised baseline}

| summarizer       | R1   | R2   | RL      |
|------------------|------|------|---------|
| Meansum on review| 0.119| 0.0081| 0.082   |
| Meansum on mask  | 0.131| 0.0079| 0.092   |
| Meansum on RSAR  | /    | in progress | /      |
| Supervised on review | 0.131 | 0.0063 | 0.119  |
| Supervised on mask| /    | in progress | /      |
| Supervised on RSAR | /    | in progress | /      |

\textbf{TABLE VIII: Supervised VS Meansum}

This part of experiment is still in progress. So far we have the results for Meansum on review, Meansum on mask and SUpervised model on review. From current results, we could say our model has comparable performance with supervised model and even higher ROUGE-2. This is not trivial as we could generate comparable results without requirement on the reference summaries. We expect our model trained on RSAR could surpass supervised model.

\textbf{VI. Conclusion, Limitations, and Future Work}

Our work consists of improving the recent unsupervised abstractive Multi-review summarization: Meansum. As a highly abstractive summarizer model, Meansum lacks attention mechanism. To address this limitation, we introduced masks on original reviews to help summarizer focus on important points. Our model demonstrates remarkable improvement on ROUGE scores. Besides, to address the unbalanced precision recall phenomenon on meansum, we suggested a regularizer on the output summary length which has a notable effect.

Regularizer on output summary length did not lead to improvement on ROUGE scores, this is probably due to the inappropriate parameters of training. Further fine tune on parameters are necessary. The comparison of our model with supervised model is still in progress, which would help us better position our model’s performance.

\textbf{REFERENCES}

[Ant19] Diego Antognini. Multi-dimensional explanation of reviews. 2019.
[CEG19] Maximin Coavoux, Hady Elsahar, and Matthias Gallé. Unsupervised aspect-based multi-document abstractive summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 42–47, Hong Kong, China, November 2019. Association for Computational Linguistics.
[CL18a] Eric Chu and Peter J. Liu. Unsupervised neural multi-document abstractive summarization. \textit{CoRR}, abs/1810.05739, 2018.
[CL18b] Eric Chu and Peter J. Liu. Unsupervised neural multi-document abstractive summarization. \textit{CoRR}, abs/1810.05739, 2018.
[KNS+16] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1328–1338, Austin, Texas, November 2016. Association for Computational Linguistics.

[Lin04] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[MST+16] Yashar Mehdad, Amanda Stent, Kapil Thadani, Dragomir Radev, Youssef Billawala, and Karolina Buchner. Extractive summarization under strict length constraints. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), pages 3089–3093, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).