Learning by Semantic Similarity Makes Abstractive Summarization Better

Wonjin Yoon [1] **Yoon Sun Yeo** [1] **Minbyul Jeong** [1] **Bong-Jun Yi** [2] **Jaewoo Kang** [1 3]

Abstract

One of the obstacles of abstractive summarization is the presence of various potentially correct predictions. Widely used objective functions for supervised learning, such as cross-entropy loss, cannot handle alternative answers effectively. Rather, they act as a training noise. In this paper, we propose Semantic Similarity strategy that can consider semantic meanings of generated summaries while training. Our training objective includes maximizing semantic similarity score which is calculated by an additional layer that estimates semantic similarity between generated summary and reference summary. By leveraging pre-trained language models, our model achieves a new state-of-the-art performance, ROUGE-L score of 41.5 on CNN/DM dataset. To support automatic evaluation, we also conducted human evaluation and received higher scores relative to both baseline and reference summaries.

1. Introduction

Text summarization is a process of automatically generating a compact summary from a document while minimizing the loss of important information. There are two dominant methods for text summarization- namely *Extractive* and *Abstractive*. Extractive summarization is a method of creating summaries by extracting important parts from the document (Zhang et al., 2018; Narayan et al., 2018; Liu & Lapata, 2019), whereas abstractive summarization is more like generating sentences using salient information from the document (See et al., 2017; Paulus et al., 2018; Lewis et al., 2019).

We mainly focus on abstractive summarization rather than extractive summarization. For abstractive summarization, supervised learning model and reinforcement learning (RL) model are widely used. The supervised learning approach is straight-forward and requires relatively less time to train (You et al., 2019; Gehrmann et al., 2018). However, the existing supervised learning model has the risk of mishandling potentially valid summaries by considering them as wrong predictions. Potentially valid summaries are semantically analogous to reference summaries, and they are often found by arranging or replacing tokens with a synonym from the reference summary. Since existing supervised learning models are trained to reproduce the reference summary exactly, producing analogous summaries will count toward wrong prediction and this phenomenon will eventually harm the training process.

Reference Summary

Summary: "Bed dumped in a hotel car park . . ."

Generated Summary

Summary 1: "Bed dumped in car park of a hotel . . ."

Summary 2: "Bed discarded in a hotel car park. . ."

Summary 3: "Bed flying in a hotel car park . . ."

Figure 1. Example sentences from summaries

To alleviate this problem, many Reinforcement Learning (RL) based models are proposed (Paulus et al., 2018; Böhm et al., 2019). RL-based models, using ROUGE metric (Paulus et al., 2018) or neural network (Böhm et al., 2019) as reward, showed remarkable performance. However, the optimization is slow and requires considerable computational effort to converge (Chen et al., 2018; Gehrmann et al., 2018).

Recently, large-scale pre-trained language models (LM), such as ElMo (Peters et al., 2018), BERT (Devlin et al., 2018), GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2019), demonstrated benefits of contextualized language representations through diverse natural language processing (NLP) tasks, including text summarization (Liu & Lapata, 2019; Zhang et al., 2019a). BART, which is composed of bidirectional transformers (encoder) and autoregressive transformers (decoder), is designed as a sequence-to-sequence model and has shown stunning performance for CNN/DailyMail dataset. However, since BART is trained

[1]Department of Computer Science and Engineering, Korea University, Seoul, South Korea [2]Clova AI, Naver Corporation, Seong-Nam, Korea [3]Interdisciplinary Graduate Program in Bioinformatics, Korea University, Seoul, South Korea. Correspondence to: Jaewoo Kang <kangj@korea.ac.kr>.

Pre-print.

by optimizing word-level cross-entropy loss between the reference summary and the generated summary, the model fails to consider semantically analogous summaries while training.

In this paper, we propose the Semantic Similarity (SemSim) strategy, which utilizes semantic distance as a loss when summarizing a text. By maximizing the semantic similarity between the generated and the reference summary, we develop a model that has more flexibility and reasonability in considering potentially valid summaries. Our approach can effectively reduce the problem that supervised learning models are faced with such as mismanagement of analogous summaries.

In experiment on CNN/DM dataset, our model BART with the SemSim approach achieved 41.5 in ROUGE-L metric, showing better performance than our baseline model BART and the current state-of-the-art model PEGASUS (Zhang et al., 2019b). More interestingly, the human evaluation result indicates that there is a statistically significant difference between the model generated summaries and human generated reference summaries. Normally, a model trained on a given reference dataset cannot outperform its own reference, which corresponds to the golden-standard dataset. However, our experiment shows conflicting results. We address this phenomenon in the Discussion section, owing this improvement to the underlying pre-trained language model structure.

Our contributions are as follow:

- We propose a strategy called Semantic Similarity, that trains model with the semantic difference between the generated summary and the reference summary. The Semantic Similarity approach allows the proposed model to handle various valid summaries, which might act as training noise when trained with maximum-likelihood loss.
- Our model obtains new state-of-the-art performances on the CNN/DailyMail dataset, 44.72 in ROUGE-1 and 41.53 in ROUGE-L. Human evaluation results demonstrate, with statistical significance, that our model produces better summaries than not only the baseline but also the reference summaries. Especially, our model shows superiority in Creativity and Relevance.
- By taking advantage of pre-trained language models and transfer learning, our model can be fine-tuned with minimum computational effort. The code and pre-trained models are available at `https://github.com/icml-2020-nlp/semsim` [1]

[1]We will update the address after the Author Notification period.

2. Related Work

2.1. Reinforcement Learning

Reinforcement Learning (RL) is a widely used learning technique for text summarization task. Paulus et al. (2018) pointed out that the loss of the supervised model is not closely related to the evaluation metric and therefore, introduced an end-to-end RL model that employs the ROUGE metric (Lin, 2004) as a rewarder.

Böhm et al. (2019) point out the limitations of ROUGE-based rewarders and proposed neural network-based rewarders to predict the similarity between document and summary. Specifically, the model is trained to predict the similarity score between the document and the summaries of various quality. The pre-trained language model BERT is used to encode the input sequences so that the semantics of the two inputs are adequately reflected in the model.

2.2. Supervised Learning

Supervised Learning is an actively researched area in summarization. See et al. (2017) introduced a sequence-to-seq attentional model that combines *coverage* vector and the copy mechanism. Gehrmann et al. (2018) proposed a bottom-up attention model by incorporating the content selection system that selects the important parts of a document. Liu et al. (2019) presented a document-level encoder using BERT (Devlin et al., 2018) and showed benefits of using pre-trained language model (LM) as embeddings. Jeh (2020) developed an approach to stack an additional encoder-decoder network, on top of an attentional encoder-decoder network to alleviate the *exposure bias* issue that comes from teacher forcing (Williams & Zipser, 1989).

Pre-trained models Recent works on pre-trained language models made significant advances in NLP tasks. BERT (Devlin et al., 2018) is a bidirectional encoder that is pre-trained by predicting the original document with the corrupted document as an input. GPT (Radford et al., 2018) and GPT-2 (Radford et al., 2019) are auto-regressive LMs. BART (Lewis et al., 2019)) is a pre-trained language model that combines bidirectional transformer as an encoder and auto-regressive transformers as a decoder. Concurrent to our work, ProphetNet (Yan et al., 2020) and PEGASUS (Zhang et al., 2019b) also use pre-trained LMs to solve text summarization task. Both model shows stunning performance by using encoder-decoder settings.

3. Method

Our proposed model, SemSim, takes account of semantic similarity, and it follows the architecture of BART (Lewis et al., 2019) to take advantage of pre-trained language model (Figure 2). We first introduce the underlying struc-

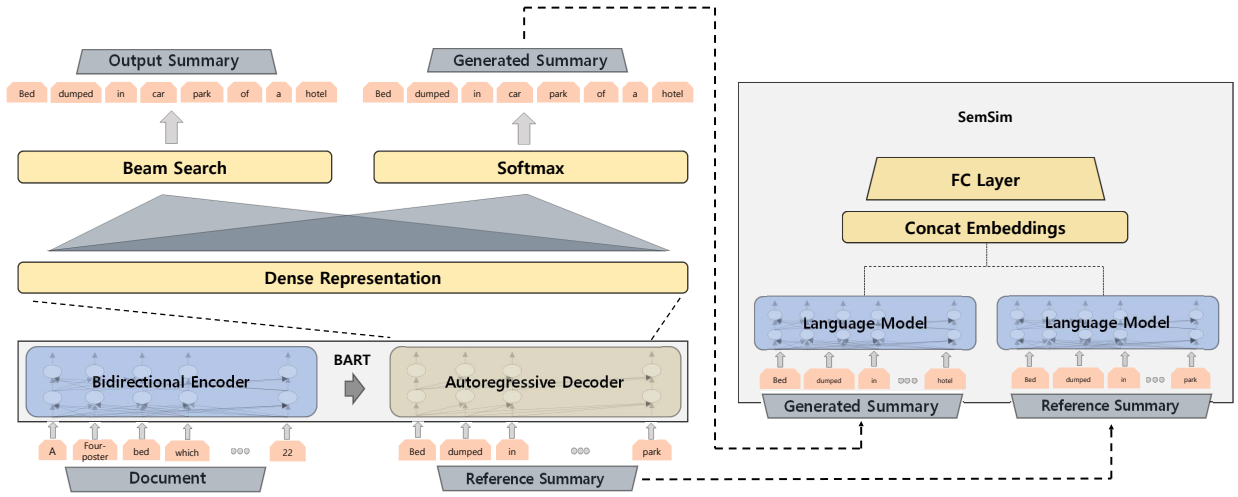

Figure 2. **SemSim** Overall Architecture, underlying structure of BART was used to represent generated summary. In SemSim layer, Language Model, which is encoding the generated summary and the reference summary, is not updating the weights. However, gradient is still calculated and flows through the SemSim layer.

ture of BART in Section 3.1, and then we elaborate in detail about our Semantic Similarity strategy in Section 3.2.

3.1. BART

BART is a denoising autoencoder that uses sequence-to-sequence transformer architecture of Vaswani et al. (2017). The structure of BART consists of two parts: an encoder and a decoder. The encoder part is a bidirectional encoder which corresponds to the structure of BERT (Devlin et al., 2018), and the decoder part is an auto-regressive decoder following the settings of GPT.

During the pre-training process, BART receives the corrupted document as input and performs the task of predicting the original uncorrupted document. In this way, BART can effectively learn contextual representations.

BART can be fine-tuned for various tasks such as token classification, sequence classification and sequence generations. When fine-tuned for summarization task, the bidirectional encoder part encodes the original document and the decoder part predicts the reference summary.

3.2. The Semantic Similarity (SemSim) Strategy

3.2.1. Motives

For sequence generation task, auto-regressive decoder models are trained to predict the next token given previous tokens. They mostly use maximum-likelihood method as a training objective to minimize loss when the input and output sequence are identical.

However, we hypothesize that this maximum-likelihood method, which minimizes cross-entropy loss, is not appropriate for summarization. This results from summarization being an open-ended task and maximum-likelihood method being too strict to account multiple valid answers.

For instance, consider the example sentences in Figure 1. Although the generated summary 1 is semantically parallel to the reference summary, the model receives penalties for the tokens car park of a hotel from the generated summary 1. Similarly, even though the generated summary 2 is closely related to the reference summary, the model mishandles this sentence as it does not include the word dumped.

By incorporating semantic similarity during training, we can obtain flexibility and handle multiple valid answers, which are otherwise penalized to act as a training noise.

3.2.2. Our approach

The Semantic Similarity strategy is a straight-forward approach that calculates the semantic similarity score. The semantic similarity score measures the similarity between the generated summary and the reference summary.

Problem Definition Let us define a set of word tokens, $S_{doc} = \{s_1^d, s_2^d, \ldots, s_n^d\}$ as a sequence of the original document and $S_{ref} = \{s_1^r, s_2^r, \ldots, s_m^r\}$ as a sequence of the reference summary. Likewise, we define a set of word tokens $S_{gen} = \{s_1^g, s_2^g, \ldots, s_l^g\}$ as a generated summary.

Baseline Model Our underlying model, BART is used to generate S_{gen} by auto-regressive process. The encoder part encodes S_{doc} and the decoder part computes, at time step t,

probability distribution $p(s_t^g|s_1^g, s_2^g, s_{t-1}^g, S_{doc})$ of token s_t^g given previous tokens and a sequence of the original document, S_{doc}. The maximum-likelihood loss can be expressed as the following equation:

$$L_{ml} = -\sum_{t=1}^{m} \log p(s_t^g|s_1^g, \ldots, s_{t-1}^g, S_{doc}) \quad (1)$$

Semantic Similarity (SemSim) layer The Semantic Similarity layer calculates the similarity score through embeddings and simple linear function. Reference and generated summary sequence, S_{ref} and S_{gen}, are encoded by a pre-trained language model (LM). Pre-trained LM, such as BERT (Devlin et al., 2018), encodes each token of the input sequence as a dense vector and then computes the embeddings of the entire sequence e_{seq}. The semantic similarity score which we denote as $Score_{semsim}$ is then calculated as:

$$e_{ref} = LM(S_{ref}) \quad (2)$$
$$e_{gen} = LM(S_{gen}) \quad (3)$$
$$e = [e_{ref}; e_{gen}] \quad (4)$$
$$Score_{semsim} = We + b \quad (5)$$

where $[\,;\,]$ denotes concatenation and LM is a language model. $e_{ref} \in R^d$ and $e_{gen} \in R^d$ indicate the embeddings of reference and generated summary respectively, where d is the number of hidden layers of LM. Both $W \in R^{1*2d}$ and $b \in R^1$ are trainable parameters of SemSim layer, and the semantic similarity loss can be defined as follows:

$$L_{semsim} = -Score_{semsim} \quad (6)$$

Training Objective We found that if we solely depend our training objective on minimizing L_{semsim}, the model requires exceptionally long training time. To alleviate this problem, we include maximum-likelihood loss as one of our training objectives and applied teacher forcing (Williams & Zipser, 1989) algorithm. Hence, our training objective is to minimize the $Loss$ defined as follows:

$$Loss = L_{ml} + L_{semsim} \quad (7)$$

4. Experiments

4.1. Dataset

We used the non-anonymized CNN/DailyMail (CNN/DM) dataset (Hermann et al., 2015; See et al., 2017) to evaluate our approach. CNN/DM dataset is composed of articles and corresponding bullet point summary pairs from the news providers. Following BART, we applied additional preprocessing steps such as replacing escape characters [2]. The preprocessed CNN/DM dataset includes 287k training pairs, 13k validation pairs, and 11k testing pairs.

4.2. Pre-trained Model

We utilized transfer learning to reduce the time spent on the learning process. Transfer learning uses a pre-trained weight as the initial point of the model rather than random values. The use of pre-trained weights helps the model to easily learn the task and reduces the learning time (Pan & Yang, 2009).

BART We utilize weights of BART large fine-tuned on CNN/DM dataset (*bart-large-cnn*) as the initial point of our model [3]. Compared to the maximum-likelihood method, learning by the SemSim approach is a relatively difficult task. When the model is learned by maximizing Maximum-likelihood, the model directly learns the correct answer token as each token is from the generated sequence. However, the model using SemSim approach learns by sequence level. Hence learning by SemSim requires excessive training time compared to the maximum-likelihood method. We can mitigate this issue by using transferred weights from the *bart-large-cnn*, which already learned how to summarize with the maximum-likelihood.

SemSim Layer Weights of SemSim layer are transferred from the pre-trained *rewarder* model of Böhm et al. (2019). The *rewarder* model calculates the similarity between the generated sequence and the reference summary and uses it as a reward for the RL-based approach. The SemSim layer of our model has identical structure with *rewarder* of Böhm et al. (2019), which is composed of a language model that encodes sequences and a linear layer. We used $BERT_{LARGE}$ as the language model.

For a fair comparison with the baseline model, we *freeze* the SemSim layer by not updating the weights so that the number of parameters does not increase. Please note that although we do not update the weights, the gradient is calculated and flows through the backward path of the SemSim layer, and accordingly, the BART structure is fine-tuned.

4.3. Settings

Following the setting of BART, we tokenized the input sequences with the byte-pair encoding of RoBERTa (Liu et al., 2019). The majority of the hyperparameter settings are the same as BART; however, we excluded training samples that exceed maximum sentence length. We used Adam optimizer (Kingma & Ba, 2014) with a learning rate of 3e-05, and dropout was applied with a probability of 0.1. Since the model requires extensive GPU memories, we restricted the maximum number of tokens in a batch to 1792 and applied

[2] https://github.com/pytorch/fairseq/blob/master/examples/bart/README.cnn.md

[3] https://github.com/pytorch/fairseq/tree/master/examples/bart

Table 1. ROUGE evaluations on CNN/DailyMail dataset and the number of trainable parameters. Since some authors did not reported the number of trainable parameters in their paper/GitHub, we estimated them and marked the cells with the asterisked (*)

Method	System	CNN/DailyMail			# of Parameters
		ROUGE-1	ROUGE-2	ROUGE-L	
Reinforcement Learning	NeuralTD+Learned (Böhm et al., 2019)	39.60	18.10	36.50	–
	REFRESH (Narayan et al., 2018)	40.00	18.20	36.60	–
	ROUGESal+Ent (RL) (Pasunuru & Bansal, 2018)	40.43	18.00	37.10	–
	rnn-ext+abs+RL+rerank (Chen & Bansal, 2018)	40.88	17.80	38.54	–
	BERT-ext+abs+RL+rerank (Bae et al., 2019)	41.90	19.08	39.64	–
Supervised Learning	pgen$_{cov}$+recorder (Jeh, 2020)	40.44	18.15	36.90	–
	Prophetnet (Yan et al., 2020)	43.68	20.64	40.72	400M*
	BART (Reported) (Lewis et al., 2019)	44.16	21.28	40.90	400M
	PEGASUS (Zhang et al., 2019b)	44.17	21.47	41.11	568M
	BART (Baseline)	43.98	21.07	40.82	400M
	SemSim (Ours)	**44.72**	**21.46**	**41.53**	400M

gradient accumulation. Our update frequency was 32. The model was trained with teacher forcing algorithm (Williams & Zipser, 1989). A single NVIDIA TITAN RTX GPU with 24GB graphic memory was used for training the model. Due to transfer learning, we only fine-tuned for 6 epochs and it took about 54 hours.

During the generation process, beam search decoding with a beam size of 5 was used to produce the output summary. Trigram blocking (Paulus et al., 2018), min-len of 55 tokens, max-len of 140 tokens and length penalty were applied during decoding (Lewis et al., 2019).

5. Evaluations

5.1. Automatic Evaluation

We report our automatic evaluation results on the CNN/DM dataset in Table 1. For summarization task, the ROUGE metrics (Lin, 2004) are widely used for evaluations, namely F-1 scores of ROUGE-1, ROUGE-2 and ROUGE-L (Paulus et al., 2018; See et al., 2017; Lewis et al., 2019).

As BART authors do not provide the exact dataset used for the experiments, we preprocess CNN/DM dataset by following the descriptions from the authors. We compare our model with the BART model, using our version of the preprocessed dataset and list the scores of **BART (Baseline)** and **SemSim (Ours)** in Table 1. For other models, we report the scores in accordance with their papers. Moreover, we address the number of trainable parameters for each model as appeared on the paper and Github. For those that are not specified, we calculate the numbers based on the information from the paper. As mentioned earlier in Section 4.2, we freeze the parameters of the SemSim layer, and hence, only the number of parameters for the underlying model, BART, is reported for the SemSim model.

Table 2. Human Evaluation score of the Systems

	Creativity	Readability	Relevance	Total (Avg)
Reference Summary	56.81	68.23	55.96	60.33
BART (Baseline)	60.14	79.65	70.71	70.17
SemSim (Ours)	**65.11**	**80.28**	**74.54**	**73.31**

Our model outperforms the baseline model, BART, in all three ROUGE metrics, and shows absolute performance improvement of 0.74, 0.39, and 0.71 in ROUGE-1, ROUGE-2, and ROUGE-L, respectively (Relative improvement of 1.68%, 1.85%, 1.73%). Our model also showed better performance than PEGASUS (Zhang et al., 2019b), a current state-of-the-art model, with the ROUGE-L score of 41.53.

5.2. Human Evaluation

5.2.1. Evaluation Criteria

In order to assess models proficiency in summarization, we follow the evaluation criteria of International English Language Testing System (IELTS)[4] as it is one of the major English test for non-native speakers across the world. IELTS writing is about summarizing information in a graph or table and writing a letter in response to a problem. Although it has different nature to the summarization task, we believe the fundamental factors should be the same because the model also needs to comprehend the given information, grasp important matters, and write with its own words.

We modified evaluation criteria to Relevance, Readability and Creativity. Both Relevance and Readability are referred from IELTS criteria but Creativity is added specifically for sentence summarization task. Creativity is a meaningful factor because a good summary should not be copied from

[4] https://www.ielts.org

the original text, but rather translated into the models own words to represent the context.

The following questions are asked to adopt score guidelines for our experiment:

- *Creativity*- Is the summary written with its own words and sentence structures?
- *Readability*- Does the summary avoid grammar errors and informal language?
- *Relevance*- Does the summary contain both important and accurate information about the original document?

For more descriptions about score guidelines, please refer to the Appendix.

5.2.2. Evaluation Setup

We use Amazon Mechanical Turk to evaluate the machine-generated summaries. For qualifications, we required all workers to possess a bachelors degree in the United States. Then, we organized a team into ten workers, and each team is requested to answer five questions, where one question includes the original document, the reference summary, and two model generated summaries. These three types of summaries are presented in random order. Overall, ten teams are participated for evaluation, meaning 100 people (1 team x 10 people) and 50 examples (5 examples x 10 teams) in total.

Workers are asked to measure the level of summarization quality from 1 to 4 in terms of Relevance, Readability, and Creativity. For these three criteria, the human examiner will judge whether the summary contains key features, avoids grammar errors, and uses its own words and sentence structures. Please consult Appendix for further details about score requirements.

Turk workers in our experiment had Human Intelligence Task (HIT) approval rate of 98% and completed over 10,000 HITs. For five questions, they spent about 1437.69 seconds(24min) on average, and the standard deviation is 782.4 seconds(13min). These statistics suggest that the time it took for workers to complete the questions varied greatly. It is clear that some workers did not respond faithfully, and therefore we excluded workers whose spent time is in the lower 5%, which is 389 seconds. Upon inspection, we realize that these inattentive workers are not biased to a certain team. Instead, there are about 0 to 2 workers in each team, whose spent time is less than 389 seconds for five questions. In this manner, we tried to not only ensure fairness but also assure the quality of human evaluation.

5.2.3. Results

Our human evaluation results are reported in Table 2. The table shows averaged score across all 475 responses (5 examples X (95 people). We convert the human evaluation scores from 1-4 scale to 0-100 scale and reported total score, which is the average of three scores. We compare the scores of our model, the baseline, and the reference summary. In terms of the total score, generated summaries from our model scored better than BART (p-value < 0.0061) and reference summaries (p-value $< 10^{-10}$) with statistical significance. In detail, our model performed better than BART on two criteria, namely Creativity (p-value < 0.0037) and Relevance (p-value < 0.0084), and outperformed reference summaries on all three criteria. BART was better than reference summaries in terms of Readability (p-value $< 10^{-10}$) and Relevance (p-value $< 10^{-10}$), but not for Creativity. We calculated p-values using the one-tailed t-test (statistical significance of 0.01).


Alessandra Ambrosio shows off her Latino tan and endless legs in edgy new fashion campaign

- **Mother-of-two, 34, snapped up to front Dafiti's AW15 campaign**
- **Latin American e-tailer believe she embodies the style of the brand**
- **Recently named No. 8 on Forbes list of top-earning models**

By BIANCA LONDON FOR MAILONLINE
PUBLISHED: 11:19 GMT, 24 April 2015 | **UPDATED:** 13:08 GMT, 24 April 2015

160 shares 77 View comments

Brazilian supermodel Alessandra Ambrosio goes back to her roots in an edgy new campaign shot in her home country.

The 34-year-old Victoria's Secret Angel shows off her Latino style and golden tan as she poses in a new campaign for online fashion retailer Dafiti, shot in São Paulo.

Dafiti, Latin America's largest online fashion retailer, has launched its own fashion collection, the Dafiti Collection, and signed Alessandra because they believe she embodies the style of the brand.

Figure 3. An example of CNN/DailyMail dataset.

6. Discussion

It is stunning that our model and BART produced a more favorable summaries than the reference summaries. We believe that this is due to the following reasons: 1) The reference summary is not always an ideal summary of the document and 2) A large-scale language model has a strong ability in text generation. We also discuss the performance difference between our model and the baseline model, BART.

As a nature of the CNN/DM dataset, some reference summaries are of poor quality and hence, these summaries received low scores compared to those generated by models. Table 3 and Figure 3 show an example of a document-summary pair. This example summary is missing crucial information to comprehend the document: the subject of the article *Alessandra Ambrosio* and who she is. For this exam-

Table 3. An example of CNN/DailyMail dataset.

CNN/DailyMail dataset	
Document	Brazilian supermodel Alessandra Ambrosio goes back to her roots in an edgy new campaign shot in her home country. The 34-year-old Victoria's Secret Angel shows off her Latino style and golden tan as she poses in a new campaign for online fashion retailer Dafiti, shot in So Paulo. Dafiti, Latin Americas largest online fashion retailer, has launched its own fashion collection, the Dafiti Collection, and signed Alessandra because they believe she embodies the style of the brand. . Scroll down for video . Alessandra Ambrosio, who found fame as a Victoria's Secret Angel, has been snapped up to front a campaign for online fashion retailer Dafiti, which was shot in So Paulo. The mother-of-two, who is No. 8 on the Forbes list of top-earning models, stars in the advertising ...
Reference Summary	Mother-of-two, 34, snapped up to front Dafiti's AW15 campaign. Latin American e-tailer believe she embodies the style of the brand. Recently named No. 8 on Forbeslist of top-earning models.

ple, reference summaries are made up of bullet points that are appeared as part of the headlines (Figure 3). Headlines are often designed to intrigue readers attention, and therefore they usually do not contain enough facts to understand the articles. It is a fatal flaw if the summary is not complete and consequently, such a summary will receive a low score for Readability. We believe that this is one of the reasons why some reference summaries received lower scores than generated summaries.

In general, we assume that the test set and training set are of the same quality because the test set intrinsically has a very similar distribution to the training set. If the test set is a low-quality data, then the training set, which is used for training, would be low quality as well. And this low quality will be reflected eventually when we train a model. Hence, it is rare for the model to produce results that are better than the dataset. However, our model has shown contradicting results: our model generated summaries are evaluated as better than the reference summaries. We believe that this is due to the transfer learning of large-scale language models. Language models are pre-trained on various corpus and we exploited this advantage. We assume that the performance difference comes from the language model. The idea of the language model is to learn the general understanding of a language during the pre-training process and the specific task during the fine-tuning process. BART, the underlying structure of our model, is trained to extract salient information by using the encoder and to make a complete sentence by using the decoder. Generating a complete sentence is closely related to providing a readable sentence and this can be achieved by extracting salient information, which can enhance relevance. For these reasons, as shown in the Table 2, our model and BART show outstanding performance in both Readability and Relevance. The successful use of pre-trained knowledge is the main reason why our model and BART received favorable scores. In addition, by considering that the human evaluation score is subjective, the Readability score of 80 is regarded as a high score.

Lastly, we would like to emphasize the strength of the SemSim approach over the BART model. Compared to BART, our model shows a significant positive difference in Creativity and Relevance. Moreover, our model allows for more flexible output because it is trained to generate a sequence that has the same meaning as the reference sequence. As a result, our model tends to have higher Creativity scores than BART. Our model also achieves higher Relevance scores due to its distinct structure. Unlike BART that predicts token-level correct answer with identical order, our model focuses on semantics where salient information is concentrated. This distinct structure of our model helps to improve Relevance scores. In addition, since the BART must learn the reference summary itself, BART is subject to learn the incompleteness, if reference summary is incomplete as aforementioned, and this may harm the output results.

7. Conclusions

In this paper, we propose a strategy for training summarization models based on semantic similarity. Our approach, called the Semantic Similarity (SemSim) strategy, allows a model to train with more flexibility than traditional maximum-likelihood methods, and this flexibility allows the model to produce better results. Specifically, a model with SemSim strategy is more likely to write a summary with its own words and sentence structures. Also, the model reflects the salient information of the original document efficiently in summary. In evaluation with ROUGE metrics, our model achieves a state-of-the-art performance of 41.53 ROUGE-L. According to human evaluation results, our model generates a more favorable summaries than the reference summaries and the baseline model summaries.

References

Bae, S., Kim, T., Kim, J., and Lee, S.-g. Summary level training of sentence rewriting for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 10–20, 2019.

Böhm, F., Gao, Y., Meyer, C. M., Shapira, O., Dagan, I., and Gurevych, I. Better rewards yield better summaries: Learning to summarise without references. In *Proceedings of the 2019 Conference on Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, China, 2019.

Chen, L., Dai, S., Tao, C., Zhang, H., Gan, Z., Shen, D., Zhang, Y., Wang, G., Zhang, R., and Carin, L. Adversarial text generation via feature-mover's distance. In *Advances in Neural Information Processing Systems*, pp. 4666–4677, 2018.

Chen, Y.-C. and Bansal, M. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 675–686, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Gehrmann, S., Deng, Y., and Rush, A. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1443. URL https://www.aclweb.org/anthology/D18-1443.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pp. 1693–1701, 2015.

Jeh, G. Encoder-decoder network as loss function for summarization, 2020. URL https://openreview.net/forum?id=SylkzaEYPS.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Liu, Y. and Lapata, M. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1387. URL https://www.aclweb.org/anthology/D19-1387.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Narayan, S., Cohen, S. B., and Lapata, M. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1747–1759, 2018.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359, 2009.

Pasunuru, R. and Bansal, M. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 646–653, 2018.

Paulus, R., Xiong, C., and Socher, R. A deep reinforced model for abstractive summarization. 2018.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Yan, Y., Qi, W., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., and Zhou, M. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*, 2020.

You, Y., Jia, W., Liu, T., and Yang, W. Improving abstractive document summarization with salient information modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2132–2141, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1205. URL https://www.aclweb.org/anthology/P19-1205.

Zhang, H., Cai, J., Xu, J., and Wang, J. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 789–797, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/K19-1074. URL https://www.aclweb.org/anthology/K19-1074.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*, 2019b.

Zhang, X., Lapata, M., Wei, F., and Zhou, M. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 779–784, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1088. URL https://www.aclweb.org/anthology/D18-1088.

Appendices

A. Code and Data Availability

Pre-processed dataset, source code, and pre-trained weights used for our experiment are available on the GitHub : https://github.com/icml-2020-nlp/semsim. [5]

B. Human Evaluation

B.1. Criteria

Table 4 displays our criteria for the human evaluation. The scores range from 1 to 4, where 1 represents poor summary and 4 represents both natural and good summary.

Relevance			
1	2	3	4
*No clear overview *Many key features are missed *Lack of information *Inaccurate information	*There is an overview *Some key features are not covered *Inaccurate Information	*A clear overview *Key feature is missed (1 or 2) *Accurate information	*A clear overview *All key features are well illustrated *Accurate information

Readability			
1	2	3	4
*Poorly readable summary *Frequent errors in grammar, punctuation or spelling *Wrong words and informal language	*Rather readable summary *Some errors in grammar, punctuation and spelling *Use less common words	*Easily readable summary *Rare errors in grammar, punctuation or spelling	*Highly readable summary *Sentences are free of errors *No grammar errors *No punctuation errors *No spelling mistakes

Creativity			
1	2	3	4
*Completely same as the ”original” *Summary is from the beginning part of the ”original”	*No attempt to create sentences *Copied most sentences from the ”original” *Most sentences are from the front part of the ”original” *Poor understanding of collocations	*Some sentences are generated but have inaccurate meaning *Tries to use complex sentences with limited success *Most sentences are from the ”original” *Some sentences are from the front part of the ”original”	*Creatively used a range of vocabulary to generate summary (Relatively few sentences are from the ”original”) *Has precise meaning *Understand collocations *Use referencing or linking words (ex, this, it, and, however etc.)

Table 4. Human evaluation criteria. Higher score indicates better quality of summary.

[5]The address is subjected to be changed after the AuthorNotification date

B.2. Truncation

As we discussed in Section 5.2.2, we applied truncation to the responses of human evaluation. Figure 4 shows the performance differences between summaries when the sample truncation percentage varies from 0% to 40%. The leftmost values of the graphs indicate the scores when all responses are used to evaluate. The rightmost values are the scores when we exclude the lower 40% of the responses in terms of response time. The graphs illustrate that the performance of our models tend to remain the same regardless of the truncation ratio.

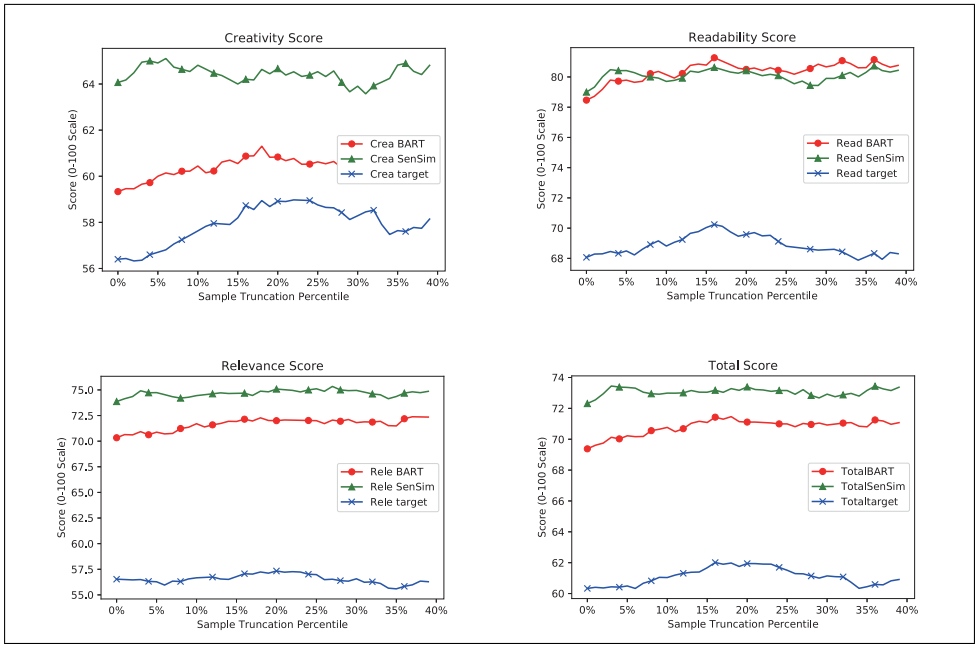

Figure 4. Sample Truncation Percentile and Scores.