FAD: Feature Alignment Discriminator for Abstractive Text Summarization

Zixuan Pan Muzhe Wu Jiarui Liu
Department of Computer Science Engineering
University of Michigan
Ann Arbor, USA
{zxp,henrw,ljrjerry}@umich.edu

Abstract

Existing abstractive approaches for automatic text summarization have shown low preciseness and coherence in their generated summaries. In this paper, we present a special fine-tuning process for text generators like BART with the FAD, the Feature Alignment Discriminator. We propose that with the token replacement detecting mechanism in feature space, the FAD greatly addresses problems of discreteness in adversarial learning for NLP and better captures the word distribution of the original texts. With extensive experiments, we find that using the first layer of BART decoder as the feature results in better performance. It is also shown that on the DailyMail/CNN dataset, that our FAD model outperforms BART base model in by 0.2 perplexity score and 0.3-0.5% ROUGE score and matches the S.O.T.A R-Drop model. We claim that the FAD structure has shown great applicability and can be used for other general text generation tasks.

1. Problem Description

Text summarization is the process of distilling the most important information from a text to produce an abridged version for a particular task and user [16]. It has shown great potential in improving people's efficiency at work, as research reveals that "summaries as short as 17% of the full text length speed up decision making twice, with no significant degradation in accuracy" [17]. Nowadays with the textual content of various kinds, e.g. articles, news, social media, etc. flooding in our life, the automatic text summarization task in natural language processing is becoming more and more important.

In our work we focused on the abstractive summarization. Compared to the extractive method, the other mainstream text summarization approach which works by selecting salient information from the text and combining them, the abstractive summarization method aims to generate summaries based on high-level understanding text and with different wording [18]. Given that the abstractive method is more similar to humans' way of thinking that consists of comprehension and cohesion, it is the more popular in nowadays researches and, with the advent of attention mechanism and the powerful pre-trained models, it have become more approachable and achieved better performance especially for corpus of controversial contents [7].

Nevertheless, there are many challenges for abstractive summarization, and one of them is how to improve the coherence and preciseness of the generated summaries. For better illustration, we provide an example 1, where a piece of sport news about the Premier League is passed into a BART-base abstractive summarizer (SOTA model for text summarization). Although the summary generated covers the information of reference summary to a great extent (high ROUGE-Recall), some unimportant factual information is also included ("Wayne Rooney has scored 12 goals ..." is included yet this piece of news report is about the injuries in Manchester United) that makes the summary inconsistent and imprecise . We argue that this phenomenon is largely due to the loss used to train the generator that fails to address the in-text word dependencies and thus propose an new architecture FAD (Section 3). After training with FAD, we managed to achieve higher cohesion in generated summaries, i.e. higher ROUGE-precision/F1 scores compared to the baseline model (Section 4).

The work contribution of our group project is shown in table 1.

2. Related Work

Text Summarization with Pre-trained Model. Neural sequence-to-sequence models with attention mechanism have long been used for abstractive text summarization tasks [23]. Nowadays with the introduction of pre-trained language models such as BERT [10] and GPT [24], text summarization have reached an unprecedented level. Equipped the knowledge of contexts acquired in encoder and decoder during pre-training, summaries generated by the model after fine-turning turn out more precise and in-

Name	Contributions
Zixuan Pan	Model design/implementation/training, proposal draft, progress report Future Plan, presentation/final report Methodology
Muzhe Wu	Model implementation review, inference, diagram, proposal Dataset/Evaluation review, progress report Methodology/Current Result, presentation/final report Problem Description/Related Work
Jiarui Liu	Model implementation review, inference, training log visualization, proposal Related Work review, progress report Data Preprocessing/Current Result, presentation Experiment result

Table 1. project contribution

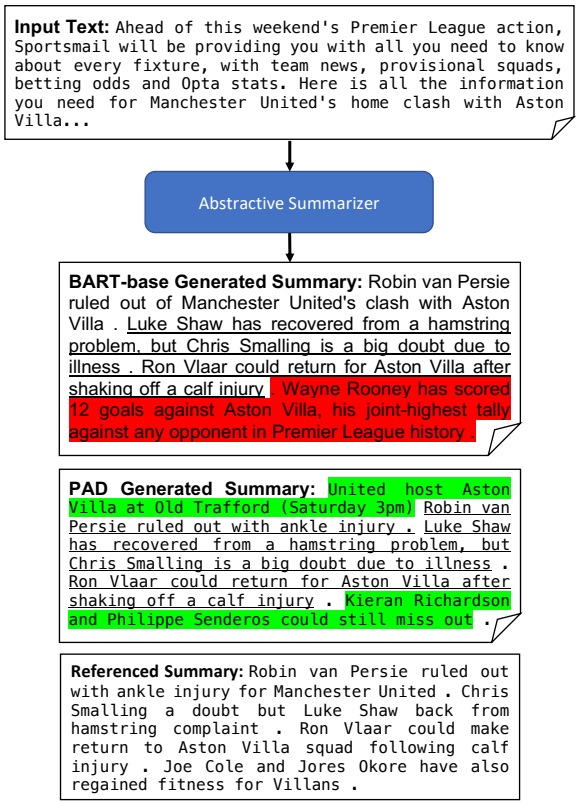

Figure 1. Comparison between summaries generated by bart-based model/our model/reference. Red highlights indicate unimportant factual information, green highlights indicate words that improve coherence and underlines indicate information matching the reference.

formative with less repetition [29]. Among these models, BART [14] is commonly acknowledged as the S.O.T.A method (BART-R3F [2] achieved best ROUGE-1/2/L scores of 40.45/20.69/36.56 on Gigaword dataset [20]). BART is designed for text generation tasks and consists of a bidirectional encoder (Bert) and a autoregressive decoder (GPT). To further improve the performance of the BART model, different methods were proposed. For example, MUPPET-BART [1] proposed pre-finetuning, an additional large-scale learning stage between language model pre-training and fine-tuning over massively multi-task dataset. Other methods, suggesting that the BART does not account for interactions between sentence-level and word-level information, include HIBRIDS [6] which takes hierarchical structure of texts into consideration when calculating the attention score and Hie-BART [3] that consist of hierarchical encoder to capture the sentence-word relations of the text. In our case we took BART-base as our backbone model given its fairly good performance (as well as the lack of computing power for pretraining or pre-finetuing). Instead of designing a better generator, we sought to train a better generator and turning to the adversarial training.

Adversarial Training in NLP. Despite its great applicability in vision tasks, the adversarial training usually falls short in natural language generation [5] due to the nature of discreteness of texts (see 3.4 for more discussions). In the light of this fact, different methods were proposed, including designing new evaluation metrics [5,26], modifying the maximum-likelihood objective [8]. In 2020, the model ELECTRA [9] was proposed with a special replaced token detection method, where before puting into the pre-trained discriminator, tokens in generated texts are sampled and used to replace tokens in the ground truth text for falsity detection. Greatly inspired by ELECTRA, we applied the discriminator with replaced token detection mechanism in our base-line BART generator so as to provide another source of rectification of the generator. To further solve the issue of in-text dependency of plain token replacement method, we choose to used the hidden state of the generator as features.

Distribution Adaption as Additional Loss. Previously when calculating classification or autoregressive loss, people neglect all the other logits and only care about the index with label 1. Some recent works have proposed to better utilize the other logits by penalize on distribution adaption. Our work is most similar to [12], where a distribution adaption loss is added to the cross entropy loss and combines a generative model with a discriminative model. However, there are two main differences. On one hand, we hope to

obtain a better generative model while they hope to obtain a better discriminative model. On the other hand, they create the target distribution in a time-delay manner with replay buffer, while ours directly use the reference summaries as target distribution.

2.1. Contribution

Our work has two major contributions:

1. We proposed FAD that achieves S.O.T.A level performance on abstractive summarization and shows promising future for other text generation tasks. We proved that feature alignment as a distribution adaption is useful to generate more concise summaries.

2. We break the discrete nature of adversarial networks in NLP by leveraging the idea that input of a discriminator need not to be words or sentences. Features of hidden layers also work.

3. Methodology

3.1. Dataset

We use CNN/DailyMail version 3.0 as the dataset. It is a widely used English dataset containing more than 300k news articles generated by journalists [19], and version 3.0 supports both abstractive and generative summarization tasks. Apart from the fact that it highly conforms to our task, we decide to use this dataset also because it is used by BART for the abstractive summarization task [14], which saves us much time to compare our experimental results with the baseline model BART base.

Each instance in the dataset contains a string for the id generated by SHA1 hash of the article url, a string for the news article body, and a string for the summary of the article[1]. The mean token count for the summaries and news articles are in Table 2.

Following the pre-processing schema of BART implemented in Fairseq (a sequence modeling toolkit that allows training custom models) [22], we encode the dataset with the GPT-2 Byte-Pair Encoding (BPE) [25], which takes in a set of unique words pre-tokenized by GPT-2 and returns a subword token list. Then, we binarize the generated data using the GPT-2 fairseq dictionary. Generated by some pretrained language models, this data-preprocessing gives a meaningful word embedding.

3.2. Feature Alignment Discriminator

The diagram of our model can be found in 2. An additional Bert-like **discriminator** is attached to the sequence-to-sequence text generator. Unlike previous methods, we use features in the hidden layer as GAN inputs. More specifically, we pass the **feature** of both generated summaries and reference summaries through the same network. Our goal is to obtain a generator that can **align** the distributions of both kinds of summaries.

Here we provided the implementation detail of the forward path of our model (Algorithm 1). For the first stage, we input the original text x to the BART generator and get the generated summaries $hypo$. By directly taking the matching referenced summary as our $target$, we calculated the negative log likelihood loss as $\mathcal{L}_{NLL}$.

For the second stage, we input the reference summaries to the generator and obtained the first hidden state in the BART decoder (shapes: Max Token Size $\times$ Batch size $\times$ Decoder Hidden State Dimension, where Max Token Size represents the maximum number of tokens can be in input x). We then stop the gradient of the label summary, since in vanilla GAN, the real samples are not passed through generator. Similar to training the masked language model and inspired by Electra, here we replace some of the tokens in the referenced summary with fake ones from our generated summaries. The replaced tokens are also first hidden layer features. Random sampling is applied

	Mean Token Count
Summaries	56
News Articles	781

Table 2. CNN/DailyMail dataset average token count

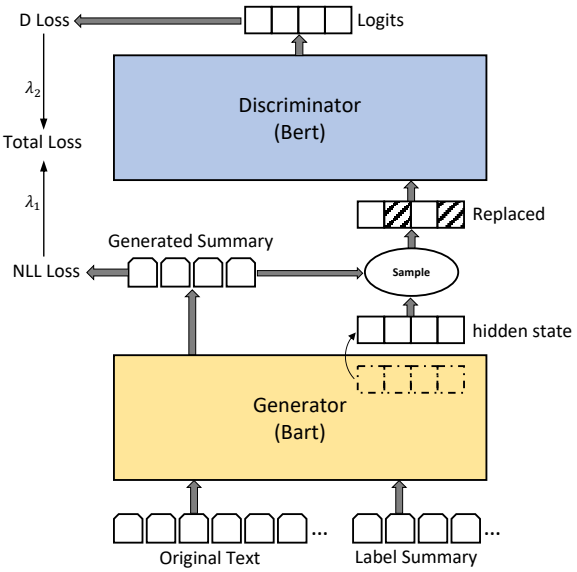

Figure 2. Our sequential model with a BART summary generator and a discriminator with BERT backbone

[1]For some sampled data instances, see https://huggingface.co/datasets/viewer/?dataset=cnn_dailymail

Algorithm 1 model.forward()

Input: x, x_{ref}

▷ First stage

hypo $\leftarrow$ BART(x)
target $\leftarrow$ x_{ref}
$\mathcal{L}_{NLL}$ $\leftarrow$ nll_loss($hypo$,$target$)

▷ Second stage

$h_{x,\text{ref}} \leftarrow$ BART(x_{ref}).detach()
{replace_ids} $\leftarrow$ random_sample(x_{ref}.index(), p_{rep})
$p(x)$ = SoftMax(hypo, dim=-1)
{candidate_ids} $\leftarrow$ (random_sample($hypo$, $p(x)$) == x_{ref}).index()
{replace_ids} = {replace_ids}\{candidate_ids}
$h_{x,\text{ref}}$[{replace_ids}] $\leftarrow$ h_x[{replace_ids}]
logits $\leftarrow$ Discriminator($h_{x,\text{ref}}$)
labels $\leftarrow$ ones_like($logits$)
labels[{replace_ids}, :] $\leftarrow$ 0
$\mathcal{L}_D \leftarrow$ BCEWithLogitsLoss($logits, labels$)

$\mathcal{L}_{total} = \mathcal{L}_{NLL} + \lambda_2 \mathcal{L}_D$

to the x_{ref} indices (first two dimensions of the first hidden state, representing each token) with uniform probability p_{rep} as $\{replace\_ids\}$. Following Electra convention, we exclude those tokens in the generated summary that are correct, another random sampling is applied for indices based on the token probability distribution, and the generated overlapping set $\{candidate\_ids\}$ is subtracted from the $\{replace\_ids\}$. Eventually, the first hidden state of reference after replacement is passed into the discriminator, and the loss of discriminator $\mathcal{L}_D$ is calculated.

The forward() function is called at each iteration and the gradients are backpropagated to update the parameters of the model, before the total loss converges.

3.3. Rationality of Discriminator

Compared to BART, the most significant difference in our model lies in applying a BERT-based discriminator that takes the last hidden state of BART decoder as the input. We propose that with the discriminator, the model can better rectify the word distribution of the generated summary to be in accordance with the word distribution of the original text, thus obtaining summaries of higher coherence.

Current text summarization models like BART are trained based on the Negative Log-Likelihood loss, where for every position in the generated sequence, we calculate and penalize the loss merely based on the probability of the correct (labeled) word token at that position, i.e.

$$\bar{y} = \text{softmax}(f_y) \tag{1}$$

$$\mathcal{L}_{NLL}(\bar{y}) = -\log(y_{ref}) \tag{2}$$

From our perspective, this way of defining loss only maximizes the probability of one certain candidate token, thus very likely to neglect the overall distribution of words in the vocabulary, and further have negative impacts on the coherence of the generated summary. Nevertheless, by introducing the discriminator, we are able to pass the referenced summary synthesized with fake tokens to a pretrained BERT model and obtain an additional D-loss based on cross-entropy:

$$\mathcal{L}_D(h_y) = \mathbb{E}[\log(D(h_{y,real}))] + \mathbb{E}[\log(1 - D(h_{y,fake}))] \tag{3}$$

Unlike the maximum likelihood method of BART, GAN fits a conditional probability $p(y|x)$ directly using a neural network [11]. On one hand, we use the whole feature as GAN's input, so that it is making the overall representation of generated summaries to approach the reference summaries instead of just maximizing the likelihood of one candidate word. On the other hand, due to the self-attention mechanism of the BERT model in the discriminator, the logits (whether each token is fake or not) predicted by the discriminator will also take into account the overall patterns of the summary. It could implicitly urge the generator to generate not only based on single words, but also based on the sentence and paragraph structures. Taking these structures into account could produce more concise summaries.

On the one hand, as we take the first hidden state of the generator h_y as the input to the discriminator rather than each separated candidate word, features representing the whole distribution is passed in. Hence, the gradient back-propagated to the BART generator would contain the knowledge of the overall text, thus more likely to train a generator that emphasizes on the coherence of summarization.

3.4. Comparison with Previous Text GANs

Due to the discrete nature of text, GANs are rarely applied in text generation tasks. Considering a language model using maximum likelihood loss, pred tokens are generated with an index sampling process:

$$\text{gen\_toks} = prob[:, \text{target\_ids}]$$

This process is not differentiable and thus gradient cannot be backpropagated into the generator. To tackle this problem, previous methods include: put generation into a reinforcement learning scenario and add rewards to the generation task [15, 28], approximating the index sampling with a smooth function [13], share the wights of discriminator and generator [9], and only updating the discriminator. However, reinforcement learning and smooth approximation methods makes the model structure different from what used in pretraining, and weight sharing methods discard the generator and only keep the discriminator.

FAD is a simpler way to break the discrete nature by using feature alignment. The idea is based on the fact that inputs from the same distribution would have similar features in hidden layers. We can also view the features as a kind of word embedding with sentence structure awareness. In FAD gradients from the discriminator can be backpropagated and train a more powerful generator.

4. Experiments

4.1. Dataset Splits

Statistics of the dataset separation is shown in Table 3. The separation splits datasets into model inputs and targets, which is the essential pair for seq2seq models, so we set data statistics according to the initial dataset splits without modifications. Previous work shows that CNN/DailyMail dataset has a lower gender bias compared other datasets [4].

Dataset Split	Number of Instances in Split
Train	287113
Validation	13368
Test	11490

Table 3. CNN/DailyMail dataset statistics

4.2. Training process

After preprocessing the data, we trained our model on GreatLakes Server with 2 A40s for around 10 hours (3 epochs). We also trained the fine-tuned the BART base model as the baseline under the same convergence criterion. The descent of the training NLL loss (obtained every 100 iterations) and validation NLL loss (obtained when each epoch ends) is displayed in Figure 3, where FAD is our model that uses the last hidden state of the decoder in the BART generator to generate samples, and FAD-2 is part of our ablation study that uses the first hidden state.

4.3. Parameter Space & Hyperparameter Tuning

Our model's parameters are listed in Table 4, 5, whose size follow the convention of BERT Small and BERT Base separately [10]. Important hyperparameters are given in Table 6. When selecting the replacement ratio, we trained for one epoch each and compare the decline of their losses. A large replacement ratio (0.7) works better than a smaller one (0.4), which probably because we need to replace tokens in the referenced summary by enough number of fake tokens from the generated summary. The loss scale is chosen as $\lambda_1/\lambda_2 = 1/50$, in order to balance their influences on the total loss $\mathcal{L}_{total} = \lambda_1\mathcal{L}_{NLL} + \lambda_2\mathcal{L}_D$. Other hyperparameters are borrowed from previous S.O.T.A to save time [14].

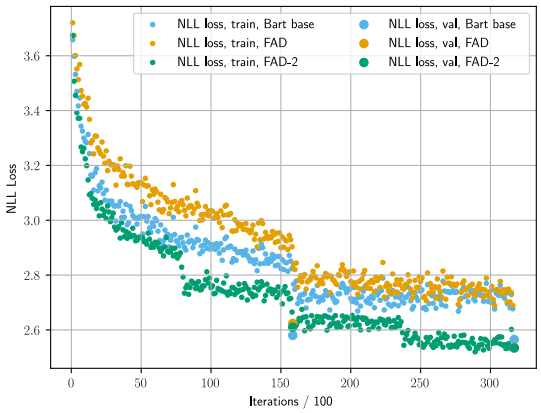

Figure 3. Training and validation loss of BART and our sequential model

Generator (Bart Base)	
Parameter Name	Value
Vocabulary Size	51200
Encoder Hidden State Dimension	768
Encoder Depth	6
Encoder FFN Dimension	3072
Decoder Hidden State Dimension	768
Decoder Depth	6
Decoder FFN Dimension	768
Max Token Size	1024

Table 4. Parameters of generator

Discriminator (Bert Small)	
Parameter Name	Value
Encoder Hidden State Dimension	256
Encoder Depth	12
Encoder FFN Dimension	1024
Max Token Size	512

Table 5. Parameters of discriminator

Parameter Name	Symbol	Value
Learning Rate	lr	3×10^{-4}
Replacement ratio	p_{rep}	0.7
Regularization Strength	α	0.7
Accumulated Gradient Count		16 / GPU
Adam Beta	(β_1, β_2)	(0.9, 0.999)
Adam Weight Decay		0.01
Loss scale	λ_1, λ_2	1, 50

Table 6. Important hyperparameters in our model

4.4. Evaluation Metrics

The evaluation metrics we use include the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score and perplexity. ROUGE is a popular set of evaluation metrics in text summarization, which measures the degree of overlaps between the generated summary and labeled summary. ROUGE-1, ROUGE-2, ROUGE-L refers to overlaps of unigrams, bigrams, and Longest Common Subsequence n-grams, respectively. We choose ROUGE score as it gives a sense of syntactical matches of the generated and labeled summary, and it gives both precision and recall measures instead of BLEU score just giving precisions [21]. The calculation of precision and recall is fairly easy to interpret, as shown in Equations 4 and 5. We also use perplexity to directly measure the ability of our model to minimize the objective function. These metrics are consistent with what BART used to evaluate summarization performance and model perplexity [14].

$$P = \frac{\text{\# n-grams in both labeled and generated summaries}}{\text{\# n-grams in generated summaries}} \quad (4)$$

$$R = \frac{\text{\# n-grams in both labeled and generated summaries}}{\text{\# n-grams in labeled summaries}} \quad (5)$$

4.5. Results

ROUGE scores and perplexities of all models are shown in Table 7. We choose BART base as our baseline model. R-Drop is the current S.O.T.A in abstractive summarization, which introduces the strategy of regularized dropout to BART by minimizing the bidirectional KL divergence [27]. In order to figure out the effect of using feature alignment on our GAN model, we also use the first hidden state (FAD-2) to replace the last hidden state (FAD) of the decoder in our BART generator to generate samples (Algorithm 1). In addition, we test the performance of FAD-2 with R-Drop by tuning some hyperparameters, as well as the performance of FAD by replacing Electra small model with Electra base.

Our first model FAD outperforms BART base on precision by 0.4%-0.6%, and on F1-score by 0.2%-0.3%. Also, the perplexity of our model decreases by 0.07. Compared to our initial FAD model, we also observe that using the first hidden layer (FAD-2) can greatly improve the recall by 0.4-0.6%, and even slightly outperforms the recall of R-Drop.

5. Discussion

Outcomes shown above indicate that applying a discriminator to the BART generator could improve the overall text summarization quality. In this section, we will analyze some samples of summaries, and discuss several comparative experiment results.

5.1. Examples

As displayed in Table 11, we select a few summaries that are labeled and generated. We notice that BART might tend to contain redundant sentences, while our model FAD-2 tends to add related information. However, there are also cases that the labeled summary is too condensed that both generated summaries are long and redundant.

5.2. Choice of Feature

Interestingly, we observed that if using last hidden layer as inputs to discriminator, recalls of FAD decrease by 0.1-0.3%, but that of using the first hidden layer features increase by 0.3-0.4% compared with our baseline. The reason is somewhat unknown, but this could be the sign that using the last hidden layer, the generator is trained to emphasize more on the overall coherence of the text by reducing unimportant tokens, which would increase the false negatives. However, using the first hidden layer of the decoder in the BART might extract more possible tokens that could be used in generated summaries, which would lead to a longer summary and slightly reduce the precision. Precision difference of FAD and FAD-2 conforms to the guess.

5.3. Discriminator Size

After substituting the Electra small model with the Electra base model, ROUGE scores and perplexity are approximately the same. The recall and F1-score seem to decline a little bit, and we guess that it may be due to some randomness during the training. Meanwhile, these results are indicating that the size of discriminator is not an important factor for our model performance.

5.4. Strategy for Gradient Stopping during Training

In the training process, we first pass the original text to BART generator (step 1 in Figure), and then the labeled summary (step 2 in Figure). Recall in Section 3, our strategy is to perform backward propagation in Step 1 and detach in Step 2. This is the case that both generator and discriminator learn properly and smoothly. On the other hand, if we do not backward in Step 1 and 2, then weights in the generator will not be updated. However, the observation that discriminator loss will go down indicates that the discriminator learns in the expected manner. Furthermore, if we go backward in both two steps, then the discriminator loss will go down too fast. This is not as expected, since it implies that generator kind of learns to distinguish generated and labeled summary, which ought to be done by discriminator.

Models	ROUGE-1			ROUGE-2			ROUGE-L			PPL
	R	P	F1	R	P	F1	R	P	F1	
BART base	49.651%	38.631%	42.349%	22.923%	17.878%	19.558%	45.878%	35.708%	39.140%	5.84
R-Drop	49.778%	39.420%	42.935%	23.214%	18.415%	20.022%	46.051%	36.487%	39.732%	5.58
FAD	49.970%	39.134%	42.833%	23.268%	18.255%	19.947%	46.252%	36.229%	39.653%	5.65
R-Drop+FAD	49.870%	39.415%	42.963%	23.331%	18.470%	20.099%	46.180%	36.523%	39.800%	5.58

Table 7. Comparison of ROUGE performance and perplexity in our models with the baseline and the S.O.T.A.

Models	ROUGE-1			ROUGE-2			ROUGE-L			PPL
	R	P	F1	R	P	F1	R	P	F1	
FAD (1)	49.372%	39.229%	42.651%	22.897%	18.230%	19.783%	45.747%	36.362%	39.528%	5.77
FAD (2)	49.970%	39.134%	42.833%	23.268%	18.255%	19.947%	46.252%	36.229%	39.653%	5.65

Table 8. Comparison of ROUGE performance and perplexity in using different hidden features. (1) uses the last hidden layer of the decoder in generator, while (2) uses the first hidden layer. By default we refer "FAD" to FAD (2) in other tables showing result.

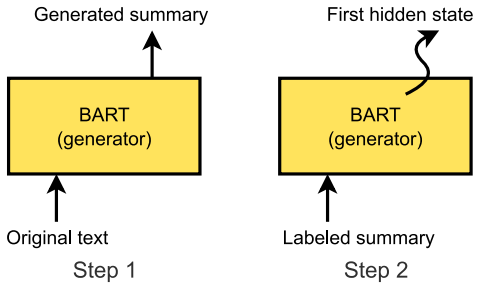

Figure 4. Two steps of forward passes through BART generator in the training process

6. Conclusion

We introduce FAD, a pre-trained model that uses feature alignment discriminator to detect replaced tokens in the abstractive text summarization task. Different from Electra, we use hidden layers of the decoder of the BART generator to generate samples and feed replaced tokens to the BERT discriminator. FAD outperforms BART base in ROUGE measures by 0.3-0.5%, and achieves comparable performance to the current S.O.T.A, R-Drop. Future work should generalize the model to other sequence-to-sequence tasks than the text summarization.

References

[1] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. arXiv preprint arXiv:2101.11038, 2021. 2

[2] Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. arXiv preprint arXiv:2008.03156, 2020. 2

[3] Kazuki Akiyama, Akihiro Tamura, and Takashi Ninomiya. Hie-bart: Document summarization with hierarchical bart. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 159–165, 2021. 2

[4] Shikha Bordia and Samuel R. Bowman. Identifying and reducing gender bias in word-level language models. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 7–15, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 5

[5] Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. arXiv preprint arXiv:1811.02549, 2018. 2

[6] Shuyang Cao and Lu Wang. Hibrids: Attention with hierarchical biases for structure-aware long document summarization. arXiv preprint arXiv:2203.10741, 2022. 2

[7] Giuseppe Carenini and Jackie Chi Kit Cheung. Extractive vs. nlg-based abstractive summarization of evaluative text: The effect of corpus controversiality. In Proceedings of the Fifth International Natural Language Generation Conference, pages 33–41, 2008. 1

[8] Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks. arXiv preprint arXiv:1702.07983, 2017. 2

[9] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders

Models	ROUGE-1			ROUGE-2			ROUGE-L			PPL
	R	P	F1	R	P	F1	R	P	F1	
Electra (1)+FAD	49.970%	39.134%	42.833%	23.268%	18.255%	19.947%	46.252%	36.229%	39.653%	5.65
Electra (2)+FAD	49.779%	39.181%	42.780%	23.140%	18.238%	19.883%	46.066%	36.263%	39.593%	5.64

Table 9. Comparison of ROUGE performance and perplexity in using models with different scaling. (1) uses Electra small model, while (2) uses Electra base model.

as discriminators rather than generators. arXiv preprint arXiv:2003.10555, 2020. 2, 4

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 1, 5

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 4

[12] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. arXiv preprint arXiv:1912.03263, 2019. 2

[13] Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. arXiv preprint arXiv:1611.04051, 2016. 4

[14] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019. 2, 3, 5, 6

[15] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547, 2017. 4

[16] Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth M Sundheim. The tipster summac text summarization evaluation. In Ninth Conference of the European Chapter of the Association for Computational Linguistics, pages 77–85, 1999. 1

[17] Sandesh Mhatre, Samata Pradhan, Shraddha Shetty, Sayali Patil, and Varsha Bhole. Automatic text summarization. 1

[18] Nikita Munot and Sharvari S Govilkar. Comparative study of text summarization methods. International Journal of Computer Applications, 102(12), 2014. 1

[19] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 280–290, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. 3

[20] Courtney Napoles, Matthew R Gormley, and Benjamin Van Durme. Annotated gigaword. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), pages 95–100, 2012. 2

[21] Ani Nenkova. Summarization evaluation for text and speech: issues and approaches. In Ninth International Conference on Spoken Language Processing, 2006. 6

[22] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of NAACL-HLT 2019: Demonstrations, 2019. 3

[23] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304, 2017. 1

[24] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 1

[25] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. 3

[26] Guy Tevet, Gavriel Habib, Vered Shwartz, and Jonathan Berant. Evaluating text gans as language models. arXiv preprint arXiv:1810.12686, 2018. 2

[27] Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. R-drop: regularized dropout for neural networks. Advances in Neural Information Processing Systems, 34, 2021. 6

[28] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017. 4

[29] Haoyu Zhang, Jianjun Xu, and Ji Wang. Pretraining-based natural language generation for text summarization. arXiv preprint arXiv:1902.09243, 2019. 2

Samples	Labeled Summary	BART Base Generated Summary	FAD-2 Generated Summary
1	Robin van Persie ruled out with ankle injury for Manchester United . Chris Smalling a doubt but Luke Shaw back from hamstring complaint . Ron Vlaar could make return to Aston Villa squad following calf injury . Joe Cole and Jores Okore have also regained fitness for Villans .	Robin van Persie ruled out of Manchester United's clash with Aston Villa . Luke Shaw has recovered from a hamstring problem, but Chris Smalling is a big doubt due to illness . Ron Vlaar could return for Aston Villa after shaking off a calf injury . ~~Wayne Rooney has scored 12 goals against Aston Villa, his joint-highest tally against any opponent in Premier League history .~~	United host Aston Villa at Old Trafford (Saturday 3pm) Robin van Persie ruled out with ankle injury . Luke Shaw has recovered from a hamstring problem, but Chris Smalling is a big doubt due to illness . Ron Vlaar could return for Aston Villa after shaking off a calf injury . **Kieran Richardson and Philippe Senderos could still miss out .**
2	Andrew Getty's death appears to be from natural causes, police say, citing coroner's early assessment . In a petition for a restraining order, Getty had written he had a serious medical condition. Police say this is not a criminal matter at this time .	Andrew Getty, 47, was found on his side near a bathroom in his home . He had "several health issues," police say . The coroner's preliminary assessment is there was no foul play involved in the death . ~~Gordon Getty is one of three living sons of oil baron J. Paul Getty .~~	Andrew Getty appears to have died of natural causes, a police spokesman says . The coroner's preliminary assessment is there was no foul play involved in the death . Andrew Getty had "several health issues," an autopsy will be conducted . **His parents, Ann and Gordon Getty, released a statement confirming their son's death .**
3	Once a super typhoon, Maysak is now a tropical storm with 70 mph winds . It could still cause flooding, landslides and other problems in the Philippines .	tropical storm Maysak is a tropical storm . It's expected to make landfall Sunday morning on the southeastern coast of Isabela province . Authorities have taken preemptive steps to keep people safe . **The storm is centered 200 miles southwest of Aurora province as of 5 p.m.** ~~(5 a.m.)~~	tropical storm Maysak approaches the Asian island nation . **It's now classified as a tropical storm, according to the Philippine national weather service .** It boasts steady winds of more than 70 mph (115 kph) and gusts up to 90 mph . Authorities have taken preemptive steps to keep people safe .
4	David Lynch says he won't be directing new episodes of Twin Peaks . Showtime "saddened" over decision, which involved a dispute over money .	Film director David Lynch has confirmed he will no longer direct the revival of "Twin Peaks" ~~The offbeat TV series featured a quirky FBI agent who went to the Pacific Northwest town of Twin Peaks to investigate the mysterious murder of a high school girl .~~ Showtime Network, which will air the nine-episode comeback, released a statement saying they were "saddened" by the decision .	David Lynch has confirmed he will no longer direct the revival of "Twin Peaks" The offbeat TV series was set to return in 2016 . **Lynch broke the news in a series of tweets, saying the show's third season will continue without him .** Showtime Network released a statement saying they were "saddened" by Lynch's decision .

Table 10. Examples of some generated summaries and labelled summaries. Based on our understanding, the sentence stroked out contains long and redundant information, and the bold text contains useful and concentrate information.

Samples	Labeled Summary	BART Base Generated Summary	FAD-2 Generated Summary
5	Here are six of CNN's best videos of the week . Clips include a look at Mike Tyson's abandoned mansion .	Mike Tyson lived in a gaudy, abandoned mansion in Ohio in the 1980s and '90s . Alisyn Camerota looked into allegations that yoga creator Bikram Choudhury sexually assaulted six former students . Penn Jillette was part of a panel speaking to CNN's Don Lemon about the controversial Indiana religious freedom law .	Some of the best videos of the week include a trip to a former heavyweight champ's gaudy, abandoned mansion . Alisyn Camerota looks into allegations that Bikram yoga creator bikram Choudhury sexually assaulted six former students . Penn Jillette was part of a panel speaking to CNN's Don Lemon about the controversial Indiana religious freedom law .
6	The singer had been off the scene for a while . She says she was bedridden for months . Lavigne was sometimes too weak to shower .	Avril Lavigne has revealed she was bedridden for five months after contracting Lyme disease . The singer believes she was bitten by a tick last spring . Lavigne is releasing a new single this month to support the 2015 Special Olympics . She says being ill was a "wake-up call" that has given her a new perspective .	Avril Lavigne says she was bedridden for five months after contracting Lyme disease . The Canadian singer believes she was bitten by a tick last spring . Lavigne is releasing a new single this month to support the 2015 Special Olympics . "I really just want to enjoy life from here on out," she says .

Table 11. Continued from previous page