Discriminative Adversarial Search for Abstractive Summarization

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano

▶ **To cite this version:**

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano. Discriminative Adversarial Search for Abstractive Summarization. 37th International Conference on Machine Learning, Jul 2020, Virtual, Åland Islands. pp.8555-8564. hal-03364422

HAL Id: hal-03364422
https://hal.sorbonne-universite.fr/hal-03364422

Submitted on 4 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discriminative Adversarial Search for Abstractive Summarization

Thomas Scialom [1 2] **Paul-Alexis Dray** [1] **Sylvain Lamprier** [2] **Benjamin Piwowarski** [2 3] **Jacopo Staiano** [1]

Abstract

We introduce a novel approach for sequence decoding, *Discriminative Adversarial Search* (DAS), which has the desirable properties of alleviating the effects of exposure bias without requiring external metrics. Inspired by Generative Adversarial Networks (GANs), wherein a discriminator is used to improve the generator, our method differs from GANs in that the generator parameters are not updated at training time and the discriminator is only used to drive sequence generation at inference time. We investigate the effectiveness of the proposed approach on the task of Abstractive Summarization: the results obtained show that DAS improves over the state-of-the-art methods, with further gains obtained via discriminator retraining. Moreover, we show how DAS can be effective for cross-domain adaptation. Finally, all results reported are obtained without additional rule-based filtering strategies, commonly used by the best performing systems available: this indicates that DAS can effectively be deployed without relying on post-hoc modifications of the generated outputs.

1. Introduction

In the context of Natural Language Generation (NLG), a majority of approaches propose sequence to sequence models trained via maximum likelihood estimation; a Teacher Forcing (Williams & Zipser, 1989) strategy is applied during training: ground-truth tokens are sequentially fed into the model to predict the next token. Conversely, at inference time, ground-truth tokens are not available: the model can only have access to its previous outputs. In the literature (Bengio et al., 2015; Ranzato et al., 2015), such mismatch is referenced to as *exposure bias*: as mistakes accumulate, this can lead to a divergence from the distribution seen at training time, resulting in poor generation outputs.

Several works have focused on alleviating this issue, proposing to optimize a *sequence level metric* such as BLEU or ROUGE: Wiseman & Rush (2016) used beam search optimisation while Ranzato et al. (2015) framed text generation as a reinforcement learning problem, using the chosen metric as reward. Still, these automated metrics suffer from known limitations: Sulem et al. (2018) showed how BLEU metrics do not reflect meaning preservation, while Novikova et al. (2017) pointed out that, for NLG tasks, they do not map well to human judgements.

Similar findings have been reported for ROUGE, in the context of abstractive summarization (Paulus et al., 2017): for the same input, several correct outputs are possible; nonetheless, the generated output is often compared to a single human reference, given the lack of annotated data. Complementary metrics have been proposed to evaluate NLG tasks, based on Question Answering (Scialom et al., 2019) or learned from human evaluation data (Böhm et al., 2019). Arguably, though, the correlation of such metrics to human judgments is still unsatisfactory.

To tackle exposure bias, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) represent a natural alternative to the proposed approaches: rather than learning from a specific metric, the model learns to generate text that a discriminator cannot differentiate from human-produced content. However, the discrete nature of text makes the classifier signal non-differentiable. A solution would be to use reinforcement learning with the classifier prediction as a reward signal. However, due to reward sparsity and mode collapse (Zhou et al., 2020), text GANs failed so far to be competitive with state-of-the-art models trained with teacher forcing on NLG tasks (Caccia et al., 2018; Clark et al., 2019), and are mostly evaluated on synthetic datasets.

Inspired by Generative Adversarial Networks, we propose an alternative approach for sequence decoding: first, a discriminator is trained to distinguish human-produced texts from machine-generated ones. Then, this discriminator is integrated into a beam search: at each decoding step, the generator output probabilities are refined according to the likelihood that the candidate sequence is human-produced. This is equivalent to optimize the search for a custom and dynamic metric, learnt to fit the human examples.

[1]reciTAL, Paris, France [2]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France [3]CNRS, France. Correspondence to: Thomas Scialom <thomas@recital.ai>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

Under the proposed paradigm, the discriminator causes the output sequences to diverge from those originally produced by the generator. These sequences, adversarial to the discriminator, can be used to further fine-tune the discriminator: following the procedure used for GANs, the discriminator can be iteratively trained on the new predictions it has contributed to improve. This effectively creates a positive feedback loop for training the discriminator: until convergence, the generated sequences improve and become harder to distinguish from human-produced text. Additionally, the proposed approach allows to dispense of custom rule-based strategies commonly used at decoding time such as length penalty and n-gram repetition avoidance.

In GANs, the discriminator is used to improve the generator and is dropped at inference time. Our proposed approach differs in that, instead, we do not modify the generator parameters at training time, and use the discriminator at inference time to drive the generation towards human-like textual content.

The main contributions of this work can be summarized as:

1. we propose *Discriminative Adversarial Search* (DAS), a novel sequence decoding approach that allows to alleviate the effects of exposure bias and to optimize on the data distribution itself rather than for external metrics;

2. we apply DAS to the abstractive summarization task, showing that even without the self-retraining procedure, our discriminated beam search procedure improves over the state-of-the-art for various metrics;

3. we report further significant improvements when applying discriminator retraining;

4. finally, we show how the proposed approach can be effectively used for domain adaptation.

2. Related Work

2.1. Exposure Bias

Several research efforts have tackled the issue of exposure bias resulting from Teacher Forcing. Inspired by Venkatraman et al. (2015), Bengio et al. (2015) proposed a variation of Teacher Forcing wherein the ground truth tokens are incrementally replaced by the predicted words. Further, Professor Forcing (Lamb et al., 2016) was devised as an adversarial approach in which the model learns to generate without distinction between training and inference time, when it has no more access to the ground truth tokens. Using automated metrics at coarser (sequence) rather than finer (token) granularity to optimize the model, Wiseman & Rush (2016) proposed a beam search variant to optimise the BLEU score in neural translation. Framing NLG as a Reinforcement Learning problem, Ranzato et al. (2015) used the reward as the metric to optimise. Paulus et al. (2017) applied a similar approach in abstractive summarization tasks, using the ROUGE metric as a reward; the authors observed that, despite the successful application of reinforcement, higher ROUGE does not yield better models: other metrics for NLG are needed. Finally, Zhang et al. (2019) proposed to select, among the different beams decoded, the one obtaining the highest BLEU score and then to fine-tune the model on that sequence.

2.2. Discriminators for Text Generation

Recent works have applied text classifiers as discriminators for different NLG tasks. Kryscinski et al. (2019) used them to detect factual consistency in the context of abstractive summarization; Zellers et al. (2019) applied discriminators to detect fake news, in a news generation scenario, reporting high accuracy (over 90%). Recently, Clark et al. (2019) proposed to train encoders as discriminators rather than language models, as an alternative to BERT (Devlin et al., 2019); they obtained better performances while improving in terms of training time. Closest to our work, Chen et al. (2020) leverage on discriminators to improve unconditional text generation following Gabriel et al. (2019) work on summarization. Abstractive summarization systems tend to be too extractive (Kryściński et al., 2018), mainly because of the copy mechanism (Vinyals et al., 2015). To improve the abstractiveness of the generated outputs, Gehrmann et al. (2018) proposed to train a classifier to detect which words from the input could be copied, and applied it as a filter during inference: to some extent, our work can be seen as the generalisation of this approach.

2.3. Text Decoding

Beam search is the de-facto algorithm used to decode generated sequences of text, in NLG taks. This decoding strategy allows to select the sequence with the highest probability, offering more flexibility than a greedy approach. Beam search has contributed to performance improvements of state-of-the-art models for many tasks, such as Neural Machine Translation, Summarization, and Question Generation (Ott et al., 2018; Dong et al., 2019). However, external rules are usually added to further constrain the generation, like the filtering mechanism for copy described above (Gehrmann et al., 2018) or the inclusion of a length penalty factor (Wu et al., 2016). Hokamp & Liu (2017) reported improvements when adding lexical constraints to beam search. Observing that neural models are prone to repetitions, while human-produced summaries contain more than 99% unique 3-grams, Paulus et al. (2017) introduced a rule in the beam forbidding the repetition of 3-grams.

	len_src	len_tgt	abstr. (%)
CNN/DM	810.69	61.04	10.23
TL;DR	248.95	30.71	36.88

Table 1. Statistics of CNN/DM and TL;DR summarization datasets. We report length in tokens for source (len_src) and summaries (len_tgt). Abstractiveness (abstr.) is the percentage of tokens in the target summary, which are not present in the source article.

Whether trained from scratch (Paulus et al., 2017; Gehrmann et al., 2018) or based on pre-trained language models (Dong et al., 2019), the current state-of-the-art results in abstractive summarization have been achieved using length penalty and 3-grams repetition avoidance.

3. Datasets

While the proposed approach is applicable to any Natural Language Generation (NLG) task, we focus on Abstractive Summarization in this study. One of most popular datasets for summarization is the CNN/Daily Mail (CNN/DM) dataset (Hermann et al., 2015; Nallapati et al., 2016). It is composed of news articles paired to multi-sentence summaries. The summaries were written by professional writers and consist of several bullet points corresponding to the important information present in the paired articles. For fair comparison, we used the exact same dataset version as previous works (See et al., 2017; Gehrmann et al., 2018; Dong et al., 2019).[1]

Furthermore, to assess the possible benefits of the proposed approach in a domain adaptation setup, we conduct experiments on TL;DR, a large scale summarization dataset built from social media data (Völske et al., 2017). We choose this dataset for two main reasons: first, its data is relatively out-of-domain if compared to the samples in CNN/DM; second, its characteristics are also quite different: compared to CNN/DM, the TL;DR summaries are twice shorter and three times more abstractive, as detailed in Table 1. The training set is composed of around 3M examples and publicly available,[2] while the test set is kept hidden because of public ongoing leaderboard evaluation. Hence, we randomly sampled 100k examples for training, 5k for validation and 5k for test. For reproducibility purposes, we make the TL;DR split used in this work publicly available.

4. Discriminative Adversarial Search

The proposed model is composed of a generator G coupled with a sequential discriminator D: at inference time, for every new token generated by G, the score and the label assigned by D is used to refine the probabilities, within a beam search, to select the top candidate sequences.

While the proposed approach is applicable to any Natural Language Generation (NLG) task, we focus on Abstractive Summarization.

Generator Abstractive summarization is usually cast as a sequence to sequence task:

$$P_\gamma(y|x) = \prod_{t=1}^{|y|} P_\gamma(y_t|x, y_{1:t-1})) \tag{1}$$

where x is the input text, y is the summary composed of $y_1 ... y_{|y|}$ tokens and γ represents the parameters of the generator. Under this framework, an abstractive summarizer is thus trained using article (x) and summary (y) pairs (e.g., via log-likelihood maximization).

Discriminator The objective of the discriminator is to label a sequence y as being *human-produced* or *machine-generated*. We use the discriminator to obtain a label at each generation step, rather than only for the entire generated sequence. For simplicity, we cast the problem as sequence to sequence, with a slight modification from our generator: at each generation step, the discriminator, instead of predicting the next token among the entire vocabulary V, outputs the probability that the input summary was generated by a human.

Learning the neural discriminator D_δ, using parameters δ, corresponds to the following logistic regression problem:

$$\frac{1}{|H|} \sum_{(x,y)\in H} \log(D_\delta(x, y)) + \frac{1}{|G|} \sum_{(x,y)\in G} \log(1 - D_\delta(x, y)) \tag{2}$$

where H and G are sets of pairs (x, y) of all texts $x \in X$ to be summarized associated to any sub-sequence y (from start to any token index t), respectively taken from ground truth summaries and generated ones:

$$H = \{(x, y_{1:t}) | x \in X \wedge y \in H(x) \wedge t \leq |y|\}$$

$$G = \{(x, y_{1:t}) | x \in X \wedge y \in G(x) \wedge t \leq |y|\}$$

where $x \in X$ stands as a text from the training set X and $H(x)$ and $G(x)$ respectively correspond to the associated human-written summary and a set of generated summaries for text x.

We refer to D_δ as a sequential discriminator, since it learns to discriminate for any partial sequence (up to the t tokens generated at step t) of any summary y. We cut all the summaries to $T = 140$ tokens if longer, consistently with previous works (Dong et al., 2019).

[1] Publicly available at https://github.com/microsoft/unilm#abstractive-summarization---cnn--daily-mail

[2] https://zenodo.org/record/1168855

4.1. Discriminative Beam Reranking

At inference time, the aim is usually to maximize the probability of the output y according to the generator (Eq. 1). The best candidate sequence is the one that maximizes $P_\gamma(y|x)$. The beam search procedure is a greedy process that iteratively constructs sequences from y_1 to y_n, while maintaining a pool of B best hypotheses generated so far at each step to allow exploration (when $B > 1$). At each step t, the process assigns a score, for every sub-sequence $y_{1:t-1}$ from the pool B, to every candidate new token y_t from the vocabulary V :

$$S_{gen}(\hat{y}) = \log P_\gamma(y_{1:t-1}|x) + \log P_\gamma(y_t|x, y_{1:t-1}) \quad (3)$$

where $\hat{y}$ results from the concatenation of a new token y_t at the end of a sequence $y_{1:t-1}$. The B sequences $\hat{y}$ with best $S_{gen}(\hat{y}, x)$ scores are kept to form the pool of hypotheses at next step. Finally, when all sequences from B are ended sequences (with the end token \$ as the last token $\hat{y}_{-1}$), the one with best S_{gen} score is returned. The beam size B corresponds to a hyper-parameter which enables to control exploration and complexity of the process. It ranges between 1 and 5 in the literature.

Algorithm 1 DAS: a Beam Search algorithm with the proposed discriminator re-ranking mechanism highligted.

Require: B, T, K_{rerank}, α
$C \leftarrow$ {Start-Of-Sentence}
for $t = 1, ..., T$ **do**
  $C \leftarrow \{\hat{y} | (\hat{y}_{1:t-1} \in C \wedge \hat{y}_t \in V) \vee (\hat{y} \in C \wedge \hat{y}_{-1} = \$)\}$
  # Pre-filter K_{rerank} sequences with top S_{gen}
  $C \leftarrow \underset{\tilde{C} \subseteq C, |\tilde{C}| = K_{rerank}}{\arg\max} \sum_{\hat{y} \in \tilde{C}} S_{gen}(\hat{y})$
  # Filter B sequences with top S_{DAS}
  $C \leftarrow \underset{\tilde{C} \subseteq C, |\tilde{C}| = B}{\arg\max} \sum_{\hat{y} \in \tilde{C}} S_{DAS}(\hat{y})$
  if only ended sequences in C **then**
    return C
  end if
end for

In our method, we propose a new score S_{DAS} to refine the score S_{gen} during the beam search w.r.t. the log probability of the discriminator, such that:

$$S_{DAS}(\hat{y}) = S_{gen}(\hat{y}) + \alpha \times S_{dis}(\hat{y}) \quad (4)$$

where $S_{dis}(\hat{y}) = log(D_\delta(x, \hat{y})$ is the discriminator log-probability that the sequence $\hat{y}$ is human-written; $\alpha \geq 0$ is used as a weighting factor. While theoretically such scores could be computed for the entire vocabulary, in practice applying the discriminator to all of the $|V| \times B$ candidate sequences $\hat{y}$ at every step t would be too time-consuming. For complexity purposes, we thus limit the re-ranking to the pool of the K_{rerank} sequences with best $S_{gen}(\hat{y}, x)$ score, as detailed in Algorithm 1.

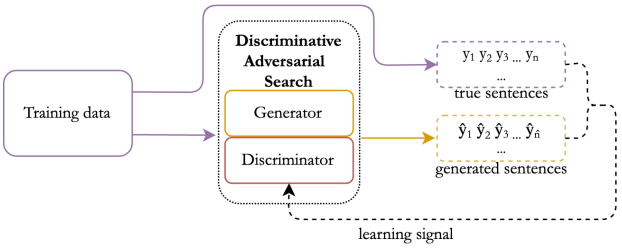

Figure 1. DAS self-training procedure: the generated examples are improved by the discriminator, and then fed back to the discriminator in a self-training loop.

4.2. Retraining the Discriminator

Under the proposed paradigm, as mentioned in Section 1, the discriminator can be fine-tuned using the outputs generated from the re-ranking process, to match the new generation distribution. Inspired by the GAN paradigm, we iteratively retrain the discriminator given the new predictions until convergence. We detail the full procedure in Figure 1, where the discriminator is retrained iteratively following equation 2 at each step. The set of generated summaries G used in equation 2 to train the discriminator at step $t+1$ corresponds to outputs from our DAS process using the discriminator at step t. This allows to consider at each step a discriminator that attempts to correct output distributions from the previous step, in order to incrementally converge to realistic distributions (w.r.t. human summaries), without requiring any retraining of the generator.

5. Experimental Protocol

Generator We build upon the Unified Language Model for natural language understanding and generation (UniLM) proposed by Dong et al. (2019); it is the current state-of-the-art model for summarization.[3] This model can be described as a Transformer (Vaswani et al., 2017) whose weights are first initialised from BERT. However, BERT is an encoder trained with bi-directional self attention: it can be used in Natural Language Understanding (NLU) tasks but not directly for generation (NLG). Dong et al. (2019) proposed to unify it for NLU and NLG: resuming its training, this time with an unidirectional loss; after this step, the model can be directly fine-tuned on any NLG task.

[3]Code and models available at https://github.com/microsoft/unilm#abstractive-summarization---cnn--daily-mail

For our ablation experiments, to save time and computation, we do not use UniLM (345 million parameters). Instead, we follow the approach proposed by the authors (Dong et al., 2019), with the difference that *1)* we start from BERT-base (110 million parameters) and *2)* we do not extend the pre-training. We actually observed little degradation than when starting from UniLM. We refer to this smaller model as *BERT-gen*. For our final results we instead use the exact same UniLM checkpoint made publicly available by Dong et al. (2019) for Abstractive Summarization.

Discriminator As detailed in Section 4, the discriminator model is also based on a sequence to sequence architecture. Thus, we can use again *BERT-gen*, initializing it the same way as the generator. The training data from CNN/DM is used to train the model; for each sample, the discriminator has access to two training examples: the human reference and a generated summary.

Hence, the full training data available for the discriminator amounts to $\sim 600k$ total examples. However, as detailed in the following Section, the discriminator does not need a lot of data to achieve a high accuracy. Therefore, we only used 150k training examples, split into 90% for training, 5% for validation and 5% for test. Unless otherwise specified, this data is only used to train/evaluate the discriminator.

Implementation details All models are implemented in PyText (Aly et al., 2018). For all our experiments we used a single RTX 2080 Ti GPU.

To train the discriminator, we used the Adam optimiser with the recommended parameters for BERT: learning rate of $3e^{-5}$, batch size of 4 and accumulated batch size of 32. We trained it for 5 epochs; each epoch took 100 minutes on 150k samples.

During discriminator retraining, the generator is needed and thus additional memory is required: all else equal, we decreased the batch size to 2. The self-training process takes one epoch to converge, in about 500 minutes: 200 minutes for training the discriminator and 300 minutes to generate the summaries with the search procedure described in Algorithm 1.

Metrics The evaluation of NLG models remains an open research question. Most of the previous works report n-grams based metrics such as ROUGE (Lin, 2004) or BLEU (Papineni et al., 2002). ROUGE-n is a recall oriented metric counting the percentage of n-grams in the gold summaries that are present in the evaluated summary. Conversely, BLEU is precision oriented.

However, as discussed in Section 1, these metrics do not correlate sufficiently w.r.t human judgments. For summarization, Louis & Nenkova (2013) showed how this issue gets even more relevant when few gold references are given. Unfortunately, the annotation of large scale dataset is not realistic: in practice, all the large scale summarization datasets rely on web-scraping to gather text-summary pairs.

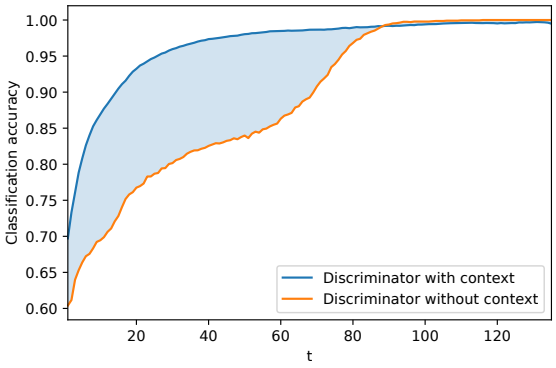

Figure 2. Accuracy of two discriminators: one is given access to the source context x while the other is not. The x-axis corresponds to the length of the discriminated sub-sequences.

For these reasons, See et al. (2017) suggested to systematically compare summarization systems with other metrics such as novelty and the number of repetitions. Following the authors' recommendation, we report the following measures for all our experiments: *i)* Novelty (*nov-n*), as the percentage of novel n-grams w.r.t. the source text, indicating the abstractiveness of a system; *ii)* Repetition (*rep-n*), as the percentage of n-grams that occur more than once in the summary; and *iii)* Length (*len*), as the length in tokens of the summary.

It is important to note that the objective is not to maximize those metrics, but to minimize the difference w.r.t. human-quality summaries. Hence, we report this difference such that for any measure m above, $\Delta m = m_{human} - m_{model}$.

6. Preliminary study

High discriminator accuracy is of utmost importance for DAS to improve the decoding search. In Fig. 2 we plot the discriminator accuracy against the generation step t, with t corresponding to the prediction for the partial sequence of the summary, $y_1, ..., y_t$ (see Eq. 2). As an ablation, we report the accuracy for a discriminator which is not given access to the source article x.

As one would expect, the scores improve with higher t, from 65% for $t = 1$ to 98% for $t = 140$: the longer the sequence $y_1, ..., y_t$ of the evaluated summary, the easier it is to discriminate it. This observed high accuracy indicates the potential benefit of using the discriminator signal to improve the generated summaries.

	K_{rerank}	DAS-single	DAS-retrain
BLEU-1	1 (BERT-gen)	27.70±0.3	27.70±0.3
	5	27.51±0.3	29.70±0.3
	10	29.18±0.3	29.81±0.2
Δ nov-1	1 (BERT-gen)	11.71±0.1	11.71±0.1
	5	11.22±0.1	10.05±0.2
	10	10.83±0.3	9.82±0.1
Δ len	1 (BERT-gen)	-9.84±0.1	-9.84±0.1
	5	-7.24±0.1	-3.05±0.1
	10	-3.14±0.1	-1.42±0.1
Δ rep-3	1 (BERT-gen)	-21.49±1.2	-21.49±1.2
	5	-17.54±0.5	-11.26±0.4
	10	-13.77±0.8	-10.45±0.4

Table 2. Scores obtained with varying K_{rerank}

When trained without access to the source article x (orange plot), the discriminator has access to little contextual and semantic information and its accuracy is lower than a discriminator who has access to x. In Fig. 2, the shaded area between the two curves represents the discrimination performance improvement attributed to using the source article x. It increases for $1 \leq t \leq 40$ and starts shrinking afterwards. After $t = 60$, corresponding to the average length of the human summaries (see Table 1), the performance of the discriminator without context quickly increases, indicating that the generated sequences contain relatively easy-to-spot mistakes. This might be due to the increased difficulty for the generator to produce longer and correct sequences, as errors may accumulate over time.

Impact of K_{rerank} and α To assess the behavior of DAS, we conducted experiments with *BERT-gen* for both the generator and the discriminator using different values for α and K_{rerank}. All models are trained using the same training data from CNN/DM, and the figures reported in Tables 2 and 3 are the evaluation results averaged across 3 runs on three different subsets (of size 1k) randomly sampled from the validation split. We compare *i) BERT-gen*, *i.e.* the model without a discriminator, *ii)* DAS-single, where the discriminator is not self-retrained, and *iii)* DAS-retrain, where the discriminator is iteratively retrained. As previously mentioned, for the repetition, novelty and length measures, we report the difference w.r.t. human summaries: the closer to 0 the better – 0 indicates no difference w.r.t. human.

The parameter K_{rerank} corresponds to the number of explored possibilities by the discriminator (see Sec. 4.1). With $K_{rerank} = 1$, no reranking is performed, and the model is equivalent to *BERT-gen*.

	α	DAS-single	DAS-retrain
BLEU-1	0 (BERT-gen)	27.70±0.3	27.70±0.3
	0.5	27.51±0.3	29.70±0.3
	1	28.38±0.3	29.25±0.2
	5	24.26±0.4	27.47±0.4
Δ nov-1	0 (BERT-gen)	11.71±0.1	11.71±0.1
	0.5	11.22±0.1	10.05±0.2
	1	10.70±0.2	9.33±0.1
	5	7.98±0.2	6.57±0.2
Δ len	0 (BERT-gen)	-9.84±0.1	-9.84±0.1
	0.5	-7.24±0.1	-3.05±0.1
	1	-4.11±0.1	-3.10±0.1
	5	-7.11±0.1	-3.85±0.1
Δ rep-3	0 (BERT-gen)	-21.49±1.2	-21.49±1.2
	0.5	-17.54±0.5	-11.26±0.4
	1	-12.85±0.4	-8.93±0.4
	5	-2.19±0.3	-5.49±0.3

Table 3. Scores obtained with varying α

In Table 2, for which we set $\alpha = 0.5$, we observe that both increasing K_{rerank} and retraining the discriminator help to better fit the human distribution (*i.e.* lower Δ): compared to *BERT-gen*, DAS models generate more novel words, are shorter and less repetitive, show improvements over the base architecture, and also obtain performance gains in terms of BLEU.

Further, we report in Table 3 results for DAS models with a fixed $K_{rerank} = 10$, while varying α. α controls the impact of the discriminator predictions when selecting the next token to generate (see Eq. 4).

With $\alpha = 0$, the discriminator is deactivated and only the generator probabilities S_{gen} are used (corresponding to Eq. 1): the model is effectively equivalent to *BERT-gen*. Consistently with the results obtained for varying K_{rerank}, we observe: DAS-retrain > DAS-single > *BERT-gen* for $\alpha \neq 5$. However, when $\alpha = 5$, BLEU scores decrease. This could indicate that a limit was reached: the higher the α, the more the discriminator influences the selection of the next word.

With $\alpha = 5$, the generated sequences are too far from the generator top-p probabilities, selected tokens at step t do not lead to useful sequences in the best K_{rerank} candidates at the following steps. The generation process struggles to represent sequences too far from what was seen during training.

	Δ len	Δ nov-1	Δ nov-3	Δ rep-1	Δ rep-3	R1	RL	B1
See et al.	-	-	-	-	-	36.38	34.24	-
Gehrmann et al.	-	-	-	-	-	41.22	38.34	-
Kryściński et al.	-	10.10	32.84	-	-	40.19	37.52	-
UniLM	-40.37	8.35	7.98	-27.99	**0.12**	43.08	40.34	34.24
UniLM (no rules)	-45.57	8.58	7.98	-31.41	-6.88	42.98	40.54	34.46
DAS-single	-29.75	**6.05**	2.80	-28.21	-4.60	42.90	40.05	35.69
DAS-retrain	**-16.81**	6.69	**2.59**	**-25.21**	-2.40	**44.05**	**40.58**	**35.94**

Table 4. Results on CNN/DM test set for the previous works as well as our proposed models.

	Δ len	Δ nov-1	Δ nov-3	Δ rep-1	Δ rep-3	R1	RL	B1
UniLM	-12.11	27.16	5.49	-6.87	**0.19**	18.66	**15.49**	16.91
UniLM (no rules)	-13.11	30.16	5.69	-7.87	-3.77	18.76	14.49	17.14
DAS-single	-10.76	19.68	4.58	-10.81	-5.05	18.19	13.30	15.41
DAS-retrain	**-2.72**	**19.05**	**1.01**	**-3.42**	-1.33	**19.76**	14.92	**17.59**

Table 5. Results on TL;DR test set for our proposed model in transfer learning scenarios.

7. Results and discussion

In our preliminary study, the best performing DAS configuration was found with $K_{rerank} = 10, \alpha = 1$. We apply this configuration in our main experiments, for fair comparison using the state-of-the-art UniLM model checkpoint.[4] Results on the CNN/DM test set are reported in Table 4. Confirming our preliminary study, DAS favorably compares to previous works, for all the metrics. Compared to UniLM, we can observe that both DAS-single and DAS-retrain are closer to the target data distribution: they allow to significantly reduce the gap with human-produced summaries over all metrics. The length of the summaries are 16.81 tokens in average longer than the human, as opposed to 40.37 tokens of difference for UniLM and 45.57 without the length penalty. DAS-retrain is also more abstractive, averaging only 2.59 novel 3-grams less than the human summaries, as opposed to 7.98 for UniLM. Notably, the proposed approach also outperforms Kryściński et al. (2018) in terms of novelty, while their model was trained with novelty as a reward in a reinforcement learning setup. UniLM applies a 3-grams repetition avoidance rule, which is why this model generates even less 3-grams repetitions than human summaries. Without this post-hoc rule, DAS-retrain generation is less repetitive compared to UniLM. Incidentally, our approach also outperforms the previous works and achieves, to the best of our knowledge, a new state-of-the-art for ROUGE.

Domain Adaptation Further, in Fig. 5, we explore a domain adaptation scenario, applying DAS-retrain on a second dataset, TL;DR. This dataset is built off social media data, as opposed to news articles as in CNN/DM, and differs from the latter in several respects, as described in Section 3. In this scenario, we keep the previously used generator (*i.e.* the UniLM checkpoint trained on CNN/DM), and only train the discriminator on a subset of TL;DR training samples. This setup has practical applications in scenarios where limited data is available: indeed, learning to generate is harder than to discriminate and requires a large amount of examples (Gehrmann et al., 2018). A discriminator can be trained with relatively few samples: in Fig. 3 we show the learning curves for discriminators trained from scratch on TL;DR training subsets of varying size. The samples are balanced: a training set size of 10k means that 5k gold summaries are used, along with 5k generated ones. We observe that only 1k examples allow the discriminator to obtain an accuracy of 82.5% at step $t = 1$. This score, higher in comparison to the one obtained on CNN/DM (see Fig. 2) is due to the relatively lower quality of the *out-of-domain* generator outputs, which makes the job easier for the discriminator.

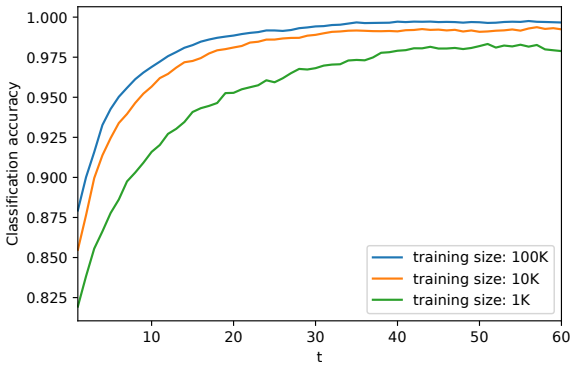

Figure 3. Learning curve for discriminators trained on TL;DR on 1k, 10k and 100k examples. The x-axis corresponds to the length of the discriminated sub-sequences.

[4] As publicly released by the authors.

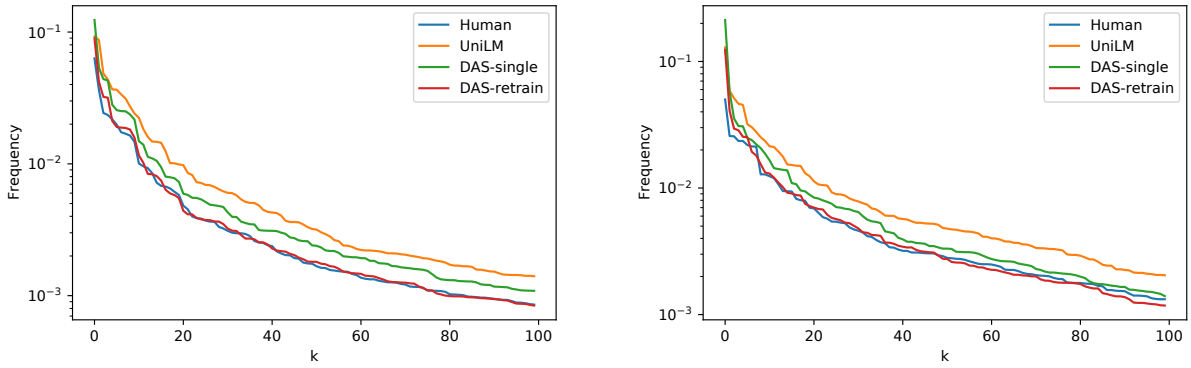

Figure 4. Vocabulary frequency for the $k = 100$ most frequent words generated by the models, for CNN/DM (left) and TL;DR (right).

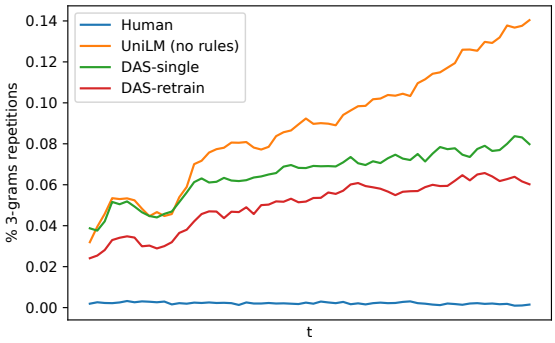

Figure 5. Distribution of 3-grams repetitions over their position t in the sequence (CNN/DM data).

The results on TL;DR[5] (Table 5) show larger improvements of DAS-retrain over UniLM, than on CNN/DM. Due to the high accuracy of the discriminator, the summaries generated are only 2.72 tokens shorter than the human ones as opposed to 12.11. They also contain more novelty and less repetitions. In terms of ROUGE and BLEU, DAS-retrain also compares favorably with the exception of ROUGE-L. This might be due to the shorter length of DAS-retrain summaries as compared to UniLM: ROUGE is a recall-oriented metric and ROUGE-L is computed for the longest common sub-sequence w.r.t. the ground truth.

[5]Models participating to the public TL;DR leaderboard (https://tldr.webis.de/) are omitted here, since they are trained on TL;DR data, and evaluated on a hidden test set. Nonetheless, assuming that the distribution of our sampled test set is similar to that of the official test set, we observe that our approach obtains comparable performance to the state-of-the-art, under a domain-adaptation setup and using only 1k training examples – exclusively for the discriminator – over an available training set of 3M examples.

Discussion In Fig. 4 we report the frequency distributions for the different models and the human summaries. We observe that DAS-retrain comes closer to the human distribution, followed by DAS-single and significantly outperforming UniLM. This shows the benefit of DAS at inference time, to produce relatively more human-like summaries. Further, the distribution of 3-grams repetition across their relative position in the sequence – Fig. 5 – shows how the gap between UniLM and Human increases more than that between DAS-retrain and human, indicating that our approach contributes to reduce the exposure bias effect. Rather than exclusively targeting exposure bias (as in Scheduled Sampling or Professor Forcing), or relying on automatic metrics as in reinforcement learning approaches, we optimize towards a discriminator instead of discrete metrics: besides reducing the exposure bias issue, this allows improvements over the other aspects captured by a discriminator.

8. Conclusion

We introduced a novel sequence decoding approach, which directly optimizes on the data distribution rather than on external metrics.

Applied to Abstractive Summarization, the distribution of the generated sequences are found to be closer to that of human-written summaries over several measures, while also obtaining improvements over the state-of-the-art.

We reported extensive ablation analyses, and showed the benefits of our approach in a domain-adaptation setup. Importantly, all these improvements are obtained without any costly generator retraining. In future work, we plan to apply DAS to other tasks such as machine translation and dialogue systems.

References

Aly, A., Lakhotia, K., Zhao, S., Mohit, M., Oguz, B., Arora, A., Gupta, S., Dewan, C., Nelson-Lindall, S., and Shah, R. Pytext: A seamless path from nlp research to production. *arXiv preprint arXiv:1812.08729*, 2018.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 1171–1179, 2015.

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 Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3101–3111, 2019.

Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., and Charlin, L. Language gans falling short. *arXiv preprint arXiv:1811.02549*, 2018.

Chen, X., Cai, P., Jin, P., Wang, H., Dai, X., and Chen, J. A discriminator improves unconditional text generation without updating the generator. *arXiv preprint arXiv:2004.02135*, 2020.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pp. 13042–13054, 2019.

Gabriel, S., Bosselut, A., Holtzman, A., Lo, K., Celikyilmaz, A., and Choi, Y. Cooperative generator-discriminator networks for abstractive summarization with narrative flow. *arXiv preprint arXiv:1907.01272*, 2019.

Gehrmann, S., Deng, Y., and Rush, A. M. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

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.

Hokamp, C. and Liu, Q. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1535–1546, 2017.

Kryściński, W., Paulus, R., Xiong, C., and Socher, R. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1808–1817, 2018.

Kryscinski, W., McCann, B., Xiong, C., and Socher, R. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*, 2019.

Lamb, A. M., Goyal, A. G. A. P., Zhang, Y., Zhang, S., Courville, A. C., and Bengio, Y. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pp. 4601–4609, 2016.

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.

Louis, A. and Nenkova, A. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300, 2013.

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

Novikova, J., Dušek, O., Cercas Curry, A., and Rieser, V. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2241–2252, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1238. URL https://www.aclweb.org/anthology/D17-1238.

Ott, M., Auli, M., Grangier, D., and Ranzato, M. Analyzing uncertainty in neural machine translation. *arXiv preprint arXiv:1803.00047*, 2018.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Paulus, R., Xiong, C., and Socher, R. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Scialom, T., Lamprier, S., Piwowarski, B., and Staiano, J. Answers unite! unsupervised metrics for reinforced summarization models. 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. 3237–3247, 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.

Sulem, E., Abend, O., and Rappoport, A. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 738–744, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1081. URL https://www.aclweb.org/anthology/D18-1081.

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.

Venkatraman, A., Hebert, M., and Bagnell, J. A. Improving multi-step prediction of learned time series models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Vinyals, O., Fortunato, M., and Jaitly, N. Pointer networks. In *Advances in neural information processing systems*, pp. 2692–2700, 2015.

Völske, M., Potthast, M., Syed, S., and Stein, B. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4508. URL https://www.aclweb.org/anthology/W17-4508.

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

Wiseman, S. and Rush, A. M. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1296–1306, 2016.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pp. 9051–9062, 2019.

Zhang, W., Feng, Y., Meng, F., You, D., and Liu, Q. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*, 2019.

Zhou, W., Ge, T., Xu, K., Wei, F., and Zhou, M. Self-adversarial learning with comparative discrimination for text generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1l8L6EtDS.