Natural Language Generation with Reinforcement Learning

Thomas Scialom

SCIENCES SORBONNE UNIVERSITÉ

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Spécialité **Informatique**

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Natural Language Generation with Reinforcement Learning

Présentée par

Thomas Scialom

Dirigée par

Dr Benjamin Piwowarski and Dr Sylvain Lamprier

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Présentée et soutenue publiquement le

Devant le jury composé de :

Sara Tonelli *Docteur, Researcher, Fondazione Bruno Kessler (FBK)*	Rapporteur
Benoît Favre *Professeur, Aix-Marseille University - (AMU)*	Rapporteur
Catherine Pelachaud *Director of Research, CNRS ISIR*	Examinateur
Marc Aurelio Ranzato *Docteur, Researcher, DeepMind*	Examinateur
Oriol Vinyals *Docteur, Principal Scientist at Google DeepMind,*	Examinateur
Jacopo Staiano *Docteur, Researcher, ReciTAL*	Directeur de thèse
Benjamin Piwowarski *Chargé de recherche, CNRS, ISIR*	Directeur de thèse
Sylvain Lamprier *Maitre de Conférence, ISIR*	Directeur de thèse

ABSTRACT

Natural Language Generation (NLG) is the subfield of Natural Language Processing, where the task is to produce natural language outputs. Despite the important progress fostered by the application of Deep Learning, generated texts are still inconsistent and contain factual inconsistencies. At the root cause, we argue in this thesis that deep learning models in NLG suffers from inherent flaws in algorithms, which limits their efficiency. At training time, the standard training strategy, Teacher Forcing, induces the so called exposure bias, a mismatch with inference time, where the errors accumulate. Moreover, NLG suffers from a second flaw: its the automatic evaluation does not reflect well human judgement.

In this thesis, we explore how to improve both evaluation and training in NLG toward more reliable systems. In particular, we propose a Question Answering based metric. We show how this metric can be used as a reward in a Reinforcement Learning setup to improve NLG models. Toward this objective, we also explore learned rewards that are the discriminators, and introduce several new algorithms that benefit NLG during training and decoding times. In particular, we propose to combine Monte Carlo Tree Search with Generative Adversarial Networks, resulting in state-of-the-art models.

REMERCIEMENTS

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Natural Language Generation

Natural Language Generation (NLG) is a subfield of Natural Language Processing, whose goal is to produce natural language outputs. NLG enables to automate manual, data-driven processes, with applications including complex report writing, automatic summarization, chatbot, or machine translation. NLG is already widely used daily by several hundred millions of people.[1]

The interest of NLG is deeply rooted in NLP, for which the first experiments consisted of human-computer dialogue systems, such as ELIZA (Weizenbaum 1966) and SHRDLU (Winograd 1980). But it was not until the late 1970s that the first natural language text generation programs were developed, leveraging on Template based approaches i.e. rule based models (McKeown 1980). These methods are limited because 1) they require human expertise, and 2) they cannot be generalized. However, given the task complexity, these methods have long dominated the field (Reiter and Dale 1997), before the recent success of Deep learning.

Deep learning approaches have opened the way for significant achievements in Natural Language Generation. Under the most popular paradigm, sequence to sequence models (Sutskever et al. 2014) are trained with Maximum Likelihood Estimation (MLE) via Teacher Forcing (Williams and Zipser 1989), a strategy that uses ground truth as input.

Despite significant advances, in the last few years, state-of-the-art models were and still are known to be *de*-generated, with outputs containing repetitions and even nonfactual information i.e. hallucination (Holtzman et al. 2019).

Among the culprits is a limitation of Teacher Forcing (Williams and Zipser 1989): the loss is computed at a token level while the aim is to produce complete sequences. Moreover, while a single ground-truth reference is considered correct, several realizations of the same content may exist. Finally, the model is subject to the Exposure Bias (Ranzato et al. 2015a), i.e. a mismatch between training and

[1] https://www.cnet.com/tech/services-and-software/google-translate-now-serves-200-million-people-daily/

inference distributions – in the latter, the model has no access to ground truth for the previously generated tokens. The literature has considered this mismatch responsible for the lower quality observed when generating longer sequences (S. Bengio et al. 2015; Lamb et al. 2016).

To mitigate MLE limitations, many attempts toward the optimisation of a sequence-level loss have been considered (S. Bengio et al. 2015; Ranzato et al. 2015a). Because sequence-level metrics in NLG are not differentiable, e.g. BLEU or ROUGE, Reinforcement Learning (RL) is a natural choice to be applied to text generation tasks (Ranzato et al. 2015a; Paulus et al. 2017), considering those metrics such as the reward.

However, the design of reliable metrics for natural language generation (NLG) systems is very challenging, and is still an open research problem. Novikova et al. (2017b) and Peyrard (2019) recently showed that metrics do not correlate well with human judgments, and argued for the development of new ones.

The core of the problem lies in the fact that for a given document, it often exists more than one unique way to write a correct output, e.g. a summary or a translation. Thus, when only a single reference is given – as is typically the case for large-scale NLG datasets used to train deep learning models, the correlation of standard automatic evaluation metrics with human judgments is low (Louis and Nenkova 2013).

For these reasons, n-gram based metrics, such as ROUGE (C.-Y. Lin 2004), are known to poorly reflect human preference (Louis and Nenkova 2013; Novikova et al. 2017b; Paulus et al. 2017; Bhandari et al. 2020). Finally, it is crucial for an effective summarization to generate texts that are factually consistent with their source documents. However, this aspect is not measured by n-grams based metrics.

When it comes to training a model, a reliable metric is even more important than for evaluation. Hence, when n-gram based metrics are used in a Reinforcement setup to train a model, this yields to poorer generation and higher degradation compared to the MLE counterparts (Paulus et al. 2017). This shows that, we cannot rely on those metrics as RL rewards. To overcome these drawbacks, better rewards are thus necessary (Paulus et al. 2017).

In this thesis, we first propose to optimise metrics and explore the positive impact they have when used as a reward. Although these new metrics allow to improve NLG models, the potential for improvement is capped due to the imperfection of the proposed measures.

To overcome this, (Ziegler et al. 2019) proposed to directly reward systems using human judgment. Although this approach performs very well and approximates the best possible reward, it is obviously not a viable solution in practice, given

the large amount of human annotation to provide for each training. However, it attests that, with perfect rewards, one can achieve excellent levels of performance. In this thesis, we explore a natural alternative, that do not require human judgments: a discriminator reward aiming to distinguish if a text sounds human or is likely generated by a machine. We propose different methods to integrate the discriminator both at training time – corresponding to a Generative Adversarial Networks (Goodfellow et al. 2014) – and at inference time – empowering decoding strategies like Beam Search. Both strategies mitigate the exposure bias, leading to improvement in NLG.

1.2 Contributions

At the core of this thesis, we aim at mitigating the exposure bias. In this Section, we briefly describe the contributions in this direction.

- Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano. "Answers Unite! Unsupervised Metrics for Reinforced Summarization Models." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

This paper corresponds to the Chapter 3. We propose a new metric based on Question Answering that is shown to measure the quality of evaluated texts more accurately than previous works. In addition, our proposed metric has the desirable feature of not requiring any gold-reference to be computed.

Leveraging on this new metric, we propose a reinforcement model for abstractive summarization, and report further improvement with respect to its baseline.

- Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang. "QuestEval: Summarization Asks for Fact-based Evaluation" Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2021.

This paper corresponds to Chapter 4. We extend the Question Answering metric presented in Chapter 3, and report a dramatic improvement in terms of performance when evaluating Summarization. Furthermore, we show how to extend the metric beyond Summarization, on other textual tasks like Text Simplification, as well as multimodal and multilingual setups.

- Scialom, Thomas and Dray, Paul-Alexis and Lamprier, Sylvain and Piwowarski, Benjamin and Staiano, Jacopo. "Discriminative Adversarial Search for Abstractive Summarization". Proceedings of the 37th International Conference on Machine Learning (ICML). 2020.

This paper corresponds to the Chapter 5. We explore two research questions: i) can a discriminator learn to accurately classify sequences? ii) can it help the generation process. We propose a new decoding mechanism that is empowered by a discriminator in addition to the standard generator. Our *Cooperative* decoding strategy enables to mitigate the exposure bias without retraining the generator. Finally, we report improvement over other standard decoding mechanism such as Beam Search.

- Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano. "ColdGANs: Taming Language GANs with Cautious Sampling Strategies". Part of Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020.

This paper corresponds to the Chapter 6. Language GANs are a natural way to mitigate the Exposure Bias. In Language GANs, because of the discrete nature of text, the task is framed under Reinforcement Learning, where the reward is the discriminator prediction for the generated sequence. This setup leads to a difficult and unstable training, usually resulting in poor performance compared to MLE training. In this work, we introduce a new sampling strategies for language GANs that for the first time compared favorably w.r.t. MLE.

- Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano. "To Beam Or Not To Beam: That is a Question of Cooperation for Language GANs". Part of Advances in Neural Information Processing Systems 34 (NeurIPS 2021). 2021.

This paper corresponds to the Chapter 7. We propose a new GAN framework where the discriminator signal is passed to the generator in a novel way, allowing more stability. We do so by sampling via the cooperative decoding strategy presented in Chapter 5. In addition, we introduce a more sophisticated cooperative process based on Monte Carlo Tree Search, which obtains state-of-the-art results.

Chapter 2

NATURAL LANGUAGE GENERATION WITH DEEP LEARNING

abstract

Natural Language Generation (NLG) is a sub-field of Natural Language Processing, where a system goal is to generate text. It encompasses several tasks like Machine Translation (MT), Summarization, or Dialogue. In this Chapter, we present how NLG has been modelized, from Recurrent Neural Networks to the current Transformers. We discuss in depth the standard training and decoding methods. In particular, we highlight an important mismatch between both, known as the exposure bias, which is the primary limitation of NLG systems, that this thesis focus on mitigating. Finally, we present the metrics used to evaluate NLG.

2.1 Natural Language Processing

NLG belongs to the Natural Language Processing (NLP) field, a branch of computer science and artificial intelligence that aims at giving computers the ability to interact with Natural Language.

In addition to NLG, the second important sub-field in NLP is Natural Language Understanding (NLU). NLU deals with machine reading comprehension and encompasses the tasks of classification, relation extraction, name entity recognition or part of speech tagging.

Text are seen like discrete sequences from which we can extract statistics, such as the number of occurrences of the words or n-grams, or other predefined linguistic features. The main method to encode text numerically as vectors, before the Deep Learning dominance was one-hot encoding. In this approach, each element in the vector corresponds to a unique word (token) in the corpus vocabulary. Then, if the token at a particular index exists in the document, that element is marked as 1. Otherwise, it's 0. This can be seen as a Boolean bag-of-words.

Many works have focused producing high quality dense representations. Among those, one-hot encoding is the most emblematic. It requires a representation with a vector dimension equals to the vocabulary space (often 50,000 in English for

instance) and lack of semantic. To mitigate these issues, Tomas Mikolov et al. (2013) proposed Word2vec, allowing a dense representation of the tokens. The token embeddings are created via a two-layer neural network. In this task, the objective is to predict a word given its context, i.e. some window of words. Once training has converged, the encoding in the hidden layer is used as the vector representation for the text.

Once vectorized, different NLU algorithms can be used to represent texts. In particular, standard Machine Learning algorithms such as Support Vector Machine (Noble 2006) or a Naive Bayes classifier, are widely used for different NLU tasks like sentiment analysis. Nowadays, Deep Learning architectures have become the new standard, exploiting dense representations such as Word2vec or its extensions with contextualized representations.

2.1.1 Natural Language Generation

Natural Language Generation (NLG) corresponds to any task requiring to generate text, in contrast to Natural Language Understanding. NLG is mostly conditional with tasks like Machine Translation, Dialogue, or Summarization.

2.1.1.1 Two cases: extract or generate

Some NLG tasks like Summarization or Question Answering can be decomposed in two main modeling families, extractive and abstractive methods. The former consists of coping content from the source, while in the former the model can generate any texts like humans.

Extractive Natural Language Generation is a technique to produce an output based on extracted piece from the source text. In the context of Summarization, it can be compared to copying down the most salient sentences of a text, an intuition developed by Mihalcea and Tarau (2004) who proposed the TextRank algorithm. Inspired by PageRank (Page et al. 1999), TextRank builds a graph of sentences within a text based on their cooccurrences. Then, it assigns an importance score for each sentence based on a random walk on the resulting graph. The most important nodes of the graph are considered as the ones that best describe the text.

In the context of Question Answering, popular datasets have been built to enhance extractive approaches. This is the case for SQuAD (Rajpurkar et al. 2016), where the answer to a question is a span contained in the context.

The main advantage of extractive methods is that the extracted parts are by definition mostly grammatically correct. However, the expressiveness of the model

is strongly limited since models can only generate an output by copying words from their input.

2.1.1.2 Generative Language Modeling

Toward more expressiveness, abstractive methods have been developed, enabling any possible output to be generated by the model. Abstractive summarization techniques tend to mimic the process of 'paraphrasing' from a text. Generated texts using this technique have the potential to look more human-like. However, there is no guarantee regarding the correctness of the produced output, and training fluent models can be challenging, as detailed all along this Thesis.

At the core of generative methods, Language modeling (LM) is the use of various statistical and probabilistic techniques to determine the probability of a given sequence s of N words occurring in a text:

$$P(s) = \prod_{i=1}^{N} p(s_i | \forall s < i, X) \tag{2.1}$$

where X is the input context in conditional NLG, e.g. the text to be summarized or translated. When the text is empty we are in the case of unconditional NLG.

The first statistical LM relies on statistics about N-grams. A probability distribution for a sequence of n is calculated, where n defines the size of the "gram", i.e. the sequence of words being assigned a probability. For example, if $n = 5$, all the sub-sequence of 5 tokens in a text constitutes the n-grams. The model then assigns probabilities using sequences of length n. Hence, n can be thought of as the amount of context the model is considering.

2.1.1.3 Training and Decoding

While n-gram uses discrete representation of Language, deep learning allows for continuous space representations, e.g. leveraging Word2vec embedding. The main paradigm is arguably Seq2Seq models (Sutskever et al. 2014), that we present extensively in Section 2.2.

Constrary to many ML problems, there is an asymmetry in the way NLG models are trained and then used for inference.

Training: Teacher Forcing In the context of Natural Language Generation (NLG), a majority of approaches propose sequence to sequence models trained via maximum likelihood estimation; Teacher Forcing (Williams and Zipser 1989)

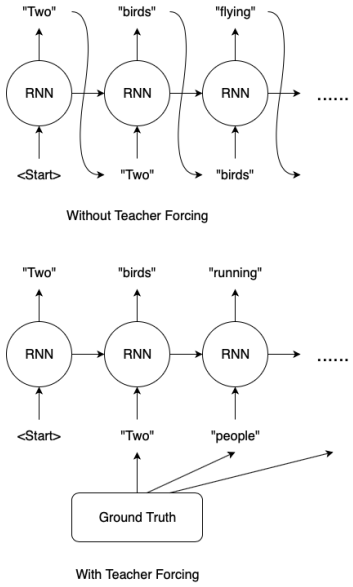


Figure 2.1: Illustration of the Exposure Bias: mismatch between training and simulation conditions. 1) without Teacher Forcing (Top), and 2 with Teacher Forcing (bottom).

strategy is used during training: ground-truth tokens are sequentially fed into the model to predict the next token.

Teacher Forcing uses the ground truth tokens to condition the prediction, since only these tokens from the train set are known. However, using the ground truth imply that the error does not accumulate. This leads to a mismatch at inference time, where the error can accumulate, since the reference is not available, as illustrated in Figure 2.1. We discuss this mismatch more in depth later (Section 2.4.1).

Decoding At decoding time, two different approaches are commonly used in NLG: Sampling and Beam Search. They respectively correspond to two different objectives, sampling from the distribution or generating the most likely one.

Sampling To obtain diverse outputs, it is common to sample tokens from the learned distribution. In particular, this is mandatory when there is no input at all, i.e. *Unconditional NLG*, to introduce stochasticity (e.g. GPT (Radford et al. 2019)). However, the longer the sequence, the more likely to sample a token from the tail of the distribution, causing degeneration (Holtzman et al. 2019). To mitigate this issue, common practices are to lower the Softmax Temperature and/or keeping only the Top K tokens (Fan et al. 2018), or those covering the Top P probability mass (Holtzman et al. 2019).

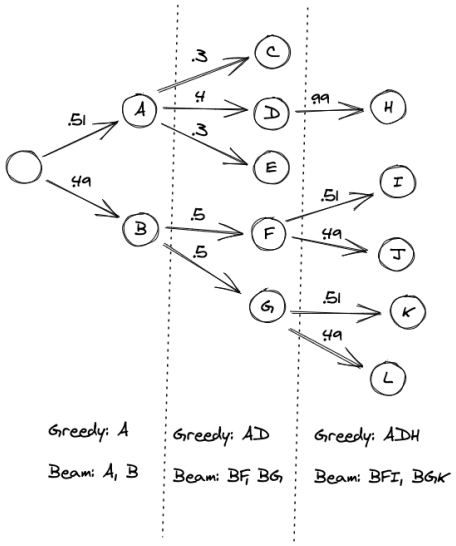


Figure 2.2: Example of a generation using BeamSearch with a width of 2 and 1 (greedy). Image from HuggingFace https://discuss.huggingface.co/t/is-beam-search-always-better-than-greedy-search/2943

Beam Search is the standard algorithm to find the sequence maximising the output probability. It works by maintaining a set of K candidates at each step of the generation. Its usage suits better *conditional* NLG tasks, where we are not interested in the diversity as we would for unconditional generation, but rather in the correctness of the generation.

This decoding algorithm allows to select the sequence with the highest probability, offering more flexibility than a greedy approach, i.e. K=1. 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).

Most models however generate incorrect sentences even with beam search. To deal with these cases, external rules are usually added to further constrain the generation, like the inclusion of a length penalty factor (Y. Wu et al. 2016). Hokamp and Q. Liu (2017) reported improvements when adding simple 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. 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.

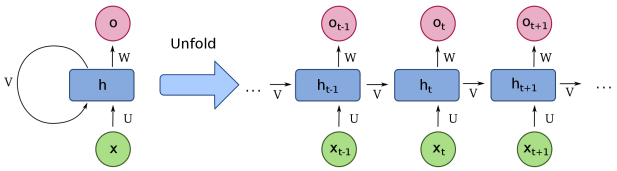


Figure 2.3: Diagram of a RNN. Compressed (left) and unfolded (right) basic recurrent neural network. Illustration taken from Wikipedia https://en.wikipedia.org/wiki/Recurrent_neural_network.

The fact that models require those decoding heuristics to enhance the generation clearly indicates a flaw in the training process. In this thesis, we show some alternative ways by learning implicitly the decoding rules.

2.2 Neural Architectures for NLG

In this section, we focus on NN for NLG which are the current state of the art.

Recurrent Neural Networks Recurrent Neural Networks (RNN) (Rumelhart et al. 1986) are a class of Neural Networks that process each element of a sequence progressively, maintaining a hidden vector representation of the sequence at each step. These latent representations play the role of an internal memory of what has been seen previously, and are used as inputs in the next steps. We present an illustration of their architecture in Figure 2.3.

RNNs were designed to process sequential data in tasks such as spatio-temporal modeling (Xingjian et al. 2015), video prediction (Srivastava et al. 2015), or NLP (Tomas Mikolov et al. 2010; Tomáš Mikolov et al. 2011; Sutskever et al. 2014).

Formerly, RNN is a function parametrized by θ, where $\theta = \{b, W, U\}$ has the form:

$$\boldsymbol{h}^{(t)} = \tanh(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}) \tag{2.2}$$

where $x^{(t)}$ represents the corresponding token, is computed from step t, and $h^{(t-1)}$ is the previous hidden state. That way, each hidden state h_t is updated auto-regressively by the previous one (h_{t-1}) and the current input $x^{(t)}$.

This yields a fully differentiable model, whose parameters θ can approximate the training distribution via Maximum Likelihood Estimation (MLE). The parameters are learned via gradient ascent, specifically with the Backpropagation Through Time (BPTT) algorithm (Williams and Zipser 1995), that consists in unfolding the RNN by successively processing the inputs $x^{(t)}$ and applying Equation

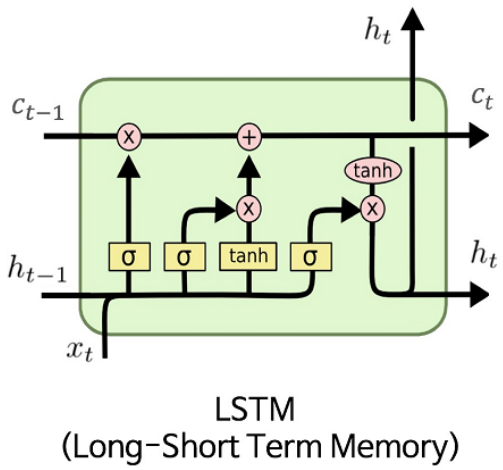


Figure 2.4: Example of an LSTM architecture (illustration taken from Wikipedia https://commons.wikimedia.org/wiki/File:LSTM.png).

2.2. Gradients are then backpropagated through the successive updates of $h^{(t)}$, and accumulated in order to update θ.

We can see from Equation 2.2 that BPTT can lead gradients to *explode or vanish*, as noted by Y. Bengio et al. (1994) and Pascanu et al. (2013). Given the derivation of the objective function with respect to W, this weight matrix is multiplied n times by itself, with n the number of steps processed. Hence, if the greatest eigenvalue is less than 1, gradients will exponentially converge to 0. Conversely, if any eigenvalue is strictly greater to 1, gradients will exponentially diverge to ∞.

LSTM To overcome the aforementioned limitation for RNN, and to better model long range dependencies, Hochreiter and Schmidhuber (1997) proposed the Long Short Term Memory, a more sophisticated implementation of RNN, depicted in Figure 2.4. In LSTM, i_t, f_t, o_t are respectively the input gate, forget gate and output gate at step t, c_t represents the cell state (memory), at time step t, and $\tilde{c_t}$ the candidate for cell state:

$$
\begin{aligned}
i_t &= \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right) \\
f_t &= \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right) \\
o_t &= \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right) \\
\tilde{c_t} &= \tanh\left(w_c\left[h_{t-1}, x_t\right] + b_c\right) \\
c_t &= f_t * c_{t-1} + i_t * \tilde{c_t} \\
h_t &= o_t * \tanh\left(c^t\right)
\end{aligned}
\tag{2.3}
$$

LSTM memory cell refers to one of the two halves of an LSTM layer's hidden state: it's the portion of the hidden state that is only modified by addi-

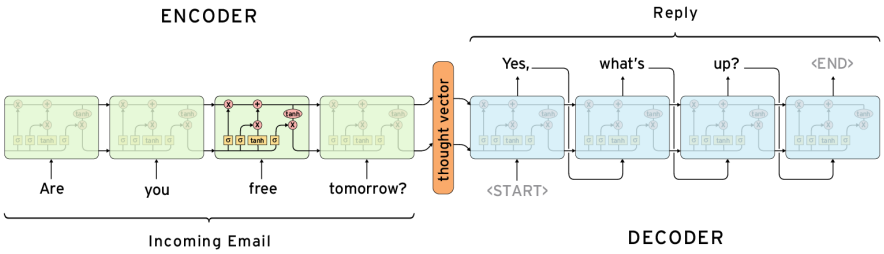


Figure 2.5: Example of a Seq2Seq architecture. (illustration taken from Wikipedia suriyadeepan.github.io).

tion/subtraction, and tends to preserve information for a relatively long time, hence vanishing gradients.

The gates control the passage of information at certain points in the network, by multiplying the current states with an activation sigmoid function, for which values are limited to $]0, 1[$. Hence, a gate lets information pass if the value is close to 1, and blocks it if the value is close to 0. This gate mechanism solves the gradient problems of RNNs, by breaking the multiplicative sequential gradient dependence.

Sequence To Sequence Many tasks in NLG consist in transforming a text into another text. For instance, in Summarization, where the input is a document composed of N tokens $(x_1, ...x_n)$, the goal is to generate a summary composed of T tokens $(y_1, ...y_t)$.

To deal with such data, Sutskever et al. (2014) proposed the Seq2Seq architecture, composed of two blocks:

1. an encoder, represented by an LSTM, that encodes the input, representing it by its successive hidden states;
2. a decoder, which is an LSTM whose initial state is the final state of the Encoder LSTM, i.e. the context vector of the encoder's final cell is the input of the first cell of the decoder network. Using this initial state, the decoder starts generating the output sequence, and these outputs are also taken into consideration for future outputs, in an auto-regressive process.

We depict the architecture in Figure 2.5. This architecture has been widely adopted by the community fpr several tasks, from Machine Translation (Sutskever et al. 2014) to Summarization (Chopra et al. 2016).

Attention Mechanism While LSTMs help mitigating the flaws of vanilla RNNs, they still lack to deal well with all the past information (Cho et al. 2014). Hence, given the last hidden state of an encoder, h_K, the decoder cannot take into account

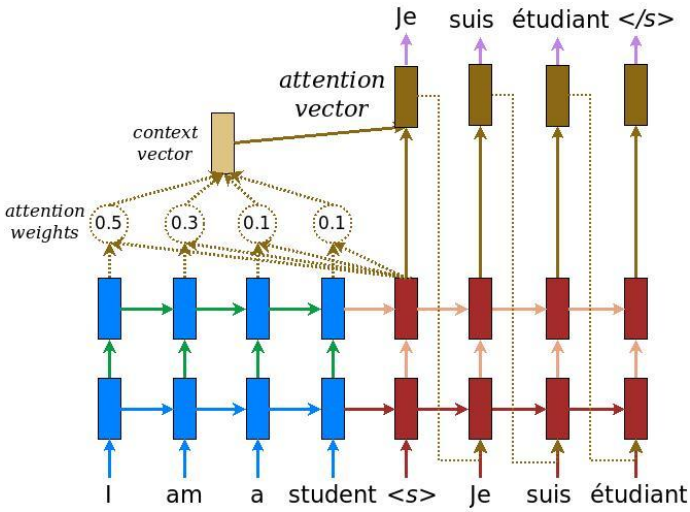


Figure 2.6: Diagram of the Attention Mechanism illustrated with a machine translation example. Illustration taken from Medium https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129.

properly the farthest inputs (e.g. x_1 or x_2). Another problem is that there is no way to give more importance to some of the input words compared to others while translating the sentence.

This bottleneck problem arises with the use of a fixed-length hidden state vector, where the decoder has limited access to the information provided by the input sequence. This is especially problematic for long sequences. To address these issues, Bahdanau et al. (2014) introduced the attention mechanism, where the idea is to not only rely on the last hidden state of the encoder, but instead use a context vector $\boldsymbol{c}^{(t)}$ that dynamically changes at each step i:

$$c^{(i)} = \sum_{j=1}^{N} \alpha_{ij} h_j \tag{2.4}$$

where $h_1, ..., h_N$ are the encoder's hidden states, and α_{ij} is the attention at step i for input j, defined as: defined as:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{N} \exp\left(e_{ik}\right)}, \tag{2.5}$$

with $e_{ij} = a\left(s_{i-1}, h_j\right)$ is the output score of a feedforward neural network described by the function that attempts to capture the alignment between input at j and output at i. We depict the attention mechanism in Figure 2.6.

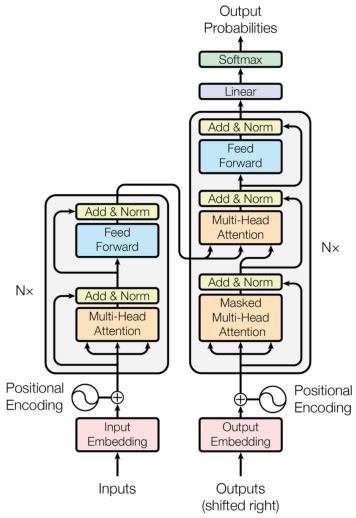


Figure 1: The Transformer - model architecture.

Figure 2.7: Illustration of the Transformer architecture.

Initially applied to Machine Translation (Bahdanau et al. 2014), the attention mechanism has proven to largely improve Seq2Seq architectures in a wide range of tasks such as summarization (Rush et al. 2015) or Image Captioning (You et al. 2016).

Transformers Thanks to the attention mechanism and the Seq2Seq architecture, great progresses have been achieved in NLG. However, based on LSTM, those models still suffer from the vanishing/exploding gradient problem. Moreover, one of their limitation is the computational cost at training time. Due to the auto-regressive nature for the LSTMs, to obtain the hidden state for a token at step t, one need to wait the computation of every previous hidden states $h_1, ...h_{t-1}$.

Vaswani et al. (2017) proposed a new Seq2Seq architecture, namely the Transformer, solely based on the attention mechanism and feed forward layers, that enables parallelisation, allowing to scale training for Seq2Seq compared to its recurrent neural network counterparts. We present the overall architecture in Figure 2.7.

Transformers have rapidly become the de-facto standard in NLG, leading to impressive progress, due to their high scalability and flexibility, resulting in state-of-the-art performance in a wide range of tasks.

For that reason, most of the models presented in this thesis rely on these well-established architectures.

In transformers, this encoder-decoder architecture is composed by stacking a series of so called Transformer blocks on top of each other. Each block is charac-

terized by a self-attention module. Given an input sequence of length n encoded as a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, the output of the self-attention layer is defined as:

$$\boldsymbol{\alpha} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \tag{2.6}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are the Query, Key and Value matrices obtained by a linear transformation of $\mathbf{X}$. More intuitively, the attention matrix, $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top$, provides text-based similarity scores for all pairs of tokens in the sequence, while each row in $\text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)$ represents a distribution that indicates information from input token ($\mathbf{V}$) for the relative importance of context items for building the representation of token ($\mathbf{Q}$).

2.3 Automatic Evaluation

Humans tend to evaluate NLG systems on a variety of dimensions like Fluency, Coherence or Correctness. However, human evaluation is time-consuming, hence automatic metrics are central in measuring progress in NLG. Despite their obvious utility, however, there is no known metric (or set of metrics) that adequately reflects human intuitions across the growing spectrum of NLG-relevant tasks.

2.3.1 Metrics

Numerous metrics have been proposed to evaluate NLG systems. Several surveys have proposed interesting taxonomies (Sai et al. 2020; Kocmi et al. 2021) to group metrics regarding some fundamental distinctions, such as being parametric (i.e. supervised) w.r.t rule-based, acting at the sequence level versus at a token-level, or the token representation being neural based versus n-gram based.

In this section, we start by presenting the n-gram based metrics, which are arguably the most widely employed metrics to compare generative models. We then present more recent metrics leveraging neural representations of sequences.

N-Grams based Metrics

- **BLEU** (Papineni et al. 2002) measures the overlap of n-grams between two texts. It is arguably the most reported metric on Machine Translation, as it is precision oriented.

- **ROUGE** (C.-Y. Lin 2004) stands for Recall-Oriented Understudy for Gisting Evaluation. Similarly to BLEU, is based on the count of overlapping n-grams, but is recall oriented. Therefore, it is the de-facto metric on Summarization. In particular, ROUGE-N is based on the count of overlapping n-grams, while ROUGE-L accounts for the longest common sub-sequences between the candidate and its corresponding reference(s). Formally, ROUGE-N is defined as:

$$ROUGE - N = \frac{\sum_{S \in S_H} \sum_{g_n \in S} C_m(g_n)}{\sum_{S \in S_H} \sum_{g_n \in S} C(g_n)} \tag{2.7}$$

where S_H is the set of human summaries, S is an individual human summary, g_n is an N-gram and $C(g_n)$ is the number of co-occurrences of g_n in the manual summary and generated summary.

- **METEOR** (Lavie and Agarwal 2007) was proposed to address one limitation in BLEU: it provides scores at the sentence level, conversely to BLEU that seeks correlation at the corpus level.

- **BERTScore** N-grams based metrics are limited in measuring similarities in a discrete space, leading to several flaws such as not taking into account synonyms. To mitigate this issue, **zhang2019bertscore** leverage on neural representation in the latent space of BERT to compute the similarity between the tokens.

- **Language Modeling Perplexity** the perplexity of the evaluated text is often reported to measure fluency from a Language Model, e.g using GPT2 (Radford et al. 2019).

Statistics Beyond metrics, it can also be interesting to compare models regarding some statistical properties on the ground truth, using some simple measures:

- **Length:** The number of n-grams in the text;
- **Repetitions:** the number of n-grams that are repeated, normalised by length;
- **Abstractness:** number of n-grams not present in the source text, normalised by the length.

2.3.2 Metrics Limitations

Automated metrics (presented in Section 2.3.1) suffer from known limitations. Sulem et al. (2018) showed how BLEU metrics do not reflect meaning preservation, while Novikova et al. (2017a) pointed out that, for NLG tasks, they do not

map well to human judgements, obtaining very low correlations with human annotators on a wide range of tasks.

Similar findings have been reported for ROUGE, in the context of abstractive summarization (Paulus et al. 2017). The main culprit is that for the same input several correct outputs are possible. Nonetheless, in most datasets, the generated output is often compared to a single human reference, given the lack of annotated data. For the metrics that are usually reported by the community, (e.g. ROUGE for summarization), the correlation with respect to human judgments is arguably still unsatisfactory. In the (Fabbri et al. 2020) study, for measuring factualness, no metric correlates more with humans that only 0.1 (Pearson Coefficient). This makes the automatic evaluation for NLG an open research question.

2.4 The Exposure Bias: A Mismatch Between Training and Inference

2.4.1 Exposure Bias

As presented Section 2.2, the standard algorithm to train NLG models is Teacher Forcing (Williams and Zipser 1989). Ground-truth tokens are sequentially fed into the model to predict the next token. Conversely, at inference time, ground-truth tokens are not available, and the model only has access to its previous outputs. The literature (S. Bengio et al. 2015; Ranzato et al. 2015b) referred to this mismatch as the *exposure bias*. This bias is severe as mistakes accumulate with generated tokens, leading to a divergence between the distribution seen at training time, resulting in poor generation.

2.4.2 Mitigating the Exposure Bias

Several research efforts have tackled the exposure bias caused by Teacher Forcing. Inspired by Venkatraman et al. (2015), S. Bengio et al. (2015) proposed a variation of Teacher Forcing wherein the ground truth tokens are incrementally replaced by the predicted words. Given that only one reference is available, this introduces a new mismatch when computing the loss, this time between the generated tokens used to condition the model, and the target tokens. Conversely, at training time, Lamb et al. (2016) proposed Professor Forcing to overcome this limit, which uses adversarial domain adaptation to encourage the dynamics of the recurrent

network to be the same when training the network and when sampling from the network over multiple time steps.

Metrics optimisation A more principled way to deal with this problem is to optimize sequence level metrics like ROUGE Paulus et al. (2017) and BLEU Ranzato et al. (2015a), instead of optimizing at a token level trough MLE. Since those metrics are non-differentiable, to optimize them one has to rely on Reinforcement Learning. The reward is then directly the metric of interest.

Following the standard RL actor-critic scheme (Konda and Tsitsiklis 1999), with $r(Y)$ the reward function for an output sequence Y, the loss to be *minimized* is defined as:

$$L_{rl} = (r(\widehat{Y}) - r(Y^s)) \sum_{t=0}^{m} \log p(y_t^s | y_0^s, ..., y_{t-1}^s, X) \tag{2.8}$$

where X is the input document to condition the model, and $r(\widehat{Y})$ corresponds to the baseline reward, given the baseline output $\widehat{Y}$, here to lower the variance and allowed faster learning. We can see that minimizing L_{rl} is equivalent to maximizing the conditional likelihood of the sampled sequence Y^s if it obtains a higher reward than the baseline $\widehat{Y}$, thus increasing the reward expectation of our model.

In Mixer, Ranzato et al. (2015a) chose BLEU, the standard metric to evaluate Machine Translation: the reward of the sequence corresponds to the BLEU score. Paulus et al. (2017) applied the same method to Summarization, using this time ROUGE.

As a drawback, all these approaches suffer from the aforementioned limitations of ill-defined reward metrics, such as BLEU or ROUGE (Paulus et al. 2017). While the results improved in terms of ROUGE, human evaluation has shown that the generated summaries were worse in term of fluency than the MLE baseline. The model learns to take advantage of metric biases, while being less correct according to human judgement. One direction we are investigating in this thesis is to develop metrics that reflect better human judgement.

Language GANs Since metrics are not reliable, considering a learned discriminator might be an alternative. In theory, a perfect discriminator would be able to judge if an output corresponds to the data distribution or not. Discriminators could therefore be an interesting alternative reward compared to other metrics.

However, to be effective, we need to train the discriminator D jointly with the generator G, framing the task as a GAN. In this setup, the generator and the discriminator play a min-max game where the former try to generate data as

similar as possible to the real data, while the later tries to distinguish between the generated data and the real data:

$$\min_G \max_D V(D,G) = \min_G \max_D \left(E_{x \sim P_{\text{data}}(x)}[\log D(x)] + E_{z \sim P_g(z)}[\log(1 - D(G(z)))] \right) \tag{2.9}$$

where P_{data} is the probability distribution of the training dataset and P_g of the generator.

Because of the discrete nature of text, continuous data cannot propagate the discriminator signal to the generator, as opposed to images GANs. For that reason, Yu et al. (2016) proposed to use Policy-Gradient reinforcement learning to learn the data generation of discrete samples. In their work, language GANs discriminators are thus the reward function within a Reinforcement Learning approach as follow:

$$L_{rl} = (D(\widehat{Y}) - D(Y^s)) \sum_{t=0}^{m} \log p(y_t^s | y_0^s, ..., y_{t-1}^s, X) \tag{2.10}$$

where the reward function D is a discriminator and X the input context.

Unfortunately, Language GANs are known for their instability, due to the unavoidable sparsity of a discriminator reward. A large body of works have proposed denser rewards: ranking or comparative discriminators (Che et al. 2017; Li et al. 2017; W. Zhou et al. 2020), using a sequential discriminator where the rewards are provided at each time step of the generation (Semeniuta et al. 2018; Masson d'Autume et al. 2019a).

Despite of those efforts, Language GANs in 2020 still under-performed MLE (Caccia et al. 2020). In this thesis, we introduce a new sampling strategies to stabilise the training process, allowing to empirically obtain competitive results in Language GANs over MLE.

Discriminators in NLG Beyond GANs, discriminators have been on a broad range of tasks. Recent works have applied text classifiers as discriminators for different NLG tasks. Kryscinski et al. (2019b) 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%). Clark et al. (2019) proposed to train encoders as discriminators rather than language models, as an efficient alternative to BERT (Devlin et al. 2019). They obtained better performances while improving in terms of training time. 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.

2.4.3 Conclusion

In conclusion, despite important progress this last five years, NLG still suffers from large flaws, from its training to its automatic evaluation.

Regarding training, the standard approach, MLE, introduces a mismatch with the inference time, the Exposure Bias, that causes a degradation the generated texts. Approaches to overcome the Exposure Bias also suffer from their own limitations, and fall short to obtain competitive results with MLE.

Regarding the evaluation, metrics fail to accurately measure the quality of an evaluated text, and are not correlating well with human judgement.

In this thesis, we propose new metrics based on Question Answering allowing to automatically evaluate NLG more accurately. Beyond evaluation, we also show that those metrics can serve to improve the training process as well. To go a step further in mitigating the exposure bias, we also leverage on discriminators to improve the generation process trough GANs.

Chapter 3

QA METRICS AS A BASIS FOR REINFORCEMENT LEARNING IN NLG

abstract

Abstractive summarization approaches based on Reinforcement Learning (RL) have recently been proposed to overcome classical likelihood maximization. RL enables to consider complex, possibly non-differentiable, metrics that globally assess the quality and relevance of the generated outputs. ROUGE, one of the most used metric, is known to be biased toward lexical similarity as well as from suboptimal accounting of fluency and readability. In this Chapter, we explore and propose alternative evaluation measures: the reported human-evaluation analysis shows that the proposed metrics, based on Question Answering, favorably compares to ROUGE – with the additional property of not requiring reference summaries. Training a RL-based model on these metrics leads to improvements (both in terms of human or automated metrics) over current approaches that use ROUGE as a reward.

- Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano. "Answers Unite! Unsupervised Metrics for Reinforced Summarization Models." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

3.1 Introduction

Summarization is one of the popular NLG tasks, where the objective is to generate short but relevant and informative texts given a variable-length inputs.

Recent - at the time of this work - research (Paulus et al. 2017; Pasunuru and Bansal 2018; Arumae and F. Liu 2019) have proposed to use evaluation metrics – and ROUGE in particular – to learn the model parameters through Reinforcement Learning (RL) techniques. This makes the choice of a good evaluation metric even more important. Unfortunately, as discussed in Section 2.3, ROUGE is known to incur several problems: in particular, its poor accounting for fluency and readability of the generated abstracts, as well as its bias towards lexical similarity

(Ng and Abrecht 2015). To emphasize the latter point, since ROUGE evaluates a summary against given human references, summarization models incur the risk of being unfairly penalized: a high quality summary might still have very few tokens/n-grams in common with the reference it is evaluated against.

In this Chapter, we propose to overcome n-gram matching based metrics, such as ROUGE, by developing metrics which are better predictors of the quality of summaries. The contributions of this part can be summarized as follows:

- Extending recent works (Eyal et al. 2019; P. Chen et al. 2018), we introduce new metrics, based on Question Answering, that do not require human annotations.
- We report a quantitative comparison of various summarization metrics, based on correlations with human assessments.
- We leverage the accuracy of the proposed metrics in several reinforcement learning schemes for summarization, including two unsupervised settings: *in-domain* (raw texts from the target documents) and *out-of-domain* (raw texts from another document collection).
- Besides a quantitative evaluation of the generated summarizes, we qualitatively evaluate the performances of the different approaches through human assessment.

Our main results can be summarized as follows:

1. We show that fitting human judgments from carefully chosen measures allows one to successfully train a reinforcement learning-based model, improving over the state-of-the-art (in terms of ROUGE and human assessments).
2. we show that dropping the requirement for human-generated reference summaries, as enabled by the proposed metrics, allows to leverage texts in a self-supervised manner and brings clear benefits in terms of performance.

Section 3.2 reviews related summarization systems and presents our proposed approaches. Section 3.3 introduces the metrics. Section 3.4 presents our experimental results and discussions.

3.2 Summarization Models

Abstractive summarization systems were originally designed as a post-processing of an extractive system – by compressing sentences (Nenkova 2011). A lot of

activity takes place nowadays in designing neural networks sequence to sequence architectures (Sutskever et al. 2014), which allow to consider the problem as a whole rather than a two-step process. Such models reached state-of-the-art performance. To tackle the summarization task, which deals with a long text and possibly out-of-vocabulary tokens, See et al. (2017) proposed to leverage an attention over the input (Bahdanau et al. 2014), as well as a copy mechanism (Vinyals et al. 2015).

One problem of sequence-to-sequence models is that they tend to repeat text in the output. To deal with this problem, (See et al. 2017) use a *coverage mechanism*, and Paulus et al. (2017) introduced *Intra-Decoder Attention* with the same goal of avoiding duplicate information within the output sequences.

More recently, the model proposed by See et al. (2017) was further extended (Gehrmann et al. 2018), with the addition of an attention mask during inference: a pre-trained sequence tagger trained to select which input tokens should be copied and used to filter the copy mechanism. Such a filter, called *Bottom-Up Copy Attention*, was shown to help prevent copying from the source text sequences that are too long, hence resulting in more abstractive summaries. On the CNN/Daily Mail dataset, (Gehrmann et al. 2018) found this two-step process to yield significant improvements in terms of ROUGE – resulting in the current state-of-the-art system. We base our experiments on this model.

To overcome limitations of MLE, approaches based on reinforcement learning have recently been proposed, allowing the models to learn via reward signals, as described in Section 2.2. Ranzato et al. (2015a) used the REINFORCE algorithm (Williams 1992) to train RNNs for several generation tasks, showing improvements over previous supervised approaches. Narayan et al. (2018b) used such an approach in an extractive summarization setting, learning to select the most relevant sentences within the input text in order to construct its summary. (Paulus et al. 2017) combined supervised and reinforcement learning, demonstrating improvements over competing approaches both in terms of ROUGE and on human evaluation. However, the main limit of these works is that they rely on standard summarization metrics which are known to be biased.

3.3 Question Answering as a metric for Summarization

3.3.1 Question-Answering based Metrics

Question-Answering is related to summarization: the first work in this direction (H. Wu et al. 2002) introduced the notion of Answer-Focused Summarization, where answers to relevant questions on the source text are used to build the corresponding summary. Based on the intuition that a good-quality summary should provide the answers to the most relevant questions on a given input text, several works have proposed to adapt Question Answering (QA) for summary quality evaluation.

In that vein, (Pasunuru and Bansal 2018) proposed to measure if answers contain the most *salient* tokens. Then, (Eyal et al. 2019) proposed *APES*, a novel metric for evaluating summarization, based on the hypothesis that the quality of a generated summary is linked to the number of questions (from a set of relevant ones) that can be answered by reading it. In their proposed setup, two components are thus needed: (a) a set of relevant questions for each source document; and (b) a QA system. For each summary to assess, questions are successively generated from a reference summary, by masking each of the named entities present in this reference, following the methodology described in (Hermann et al. 2015). This results in as many triplets $(input, question, answer)$ as named entities present in the reference summary, where *input* denotes the summary to assess, *question* refers to the sentence containing the masked entity and *answer* refers to this masked entity to retrieve. Thus, for each summary to assess, different metrics can be derived to assess the ability of the QA system to retrieve the correct answers. Bellow, we present in particular the F1 score and the QA confidence.

F1 score For each triplet, an F1 score is computed according to the responses retrieved by the considered QA system. This score, commonly used for QA evaluation (Rajpurkar et al. 2016), measures the average overlap between predictions and ground truth answers. For each summary to assess, the final score is the average of the F1 score computed over each triplet. In the following, we denote this metric as $QA_{fscore}(sup)$.

QA confidence Complementary to the F1 score, we propose to also consider the confidence of the QA system in its retrieved answer. This corresponds, for each triplet, to the probability of the true answer according to the QA model.

Confidence scores are averaged for each summary to assess over its associated triplets. In the following, we denote this metric as $QA_{conf}(sup)$.

Besides considering the simple presence of the expected answers in the generated summary, QA-based metrics also account to some extent for readability. They indeed require that the considered QA system, trained on natural language, be able to find the answer in the input to assess, despite the variability of the generated texts.

Extension to the unsupervised setting While being a useful complement to ROUGE, the two QA-based metrics described above still need human-generated summaries, which can be a limitation when training systems on data not annotated. We investigate and propose extending the previously described QA-based approach in an unsupervised setting.

With this aim, we extend the above metrics at *the document level* (more precisely, questions and answers are generated from the source article text rather than from the reference summary), dispensing of the need for human-generated reference summaries. Thus, we propose two unsupervised QA-based metrics, to which we refer to as $QA_{fscore}(unsup)$ and $QA_{conf}(unsup)$. Accounting for both quality and informativeness of a generated summary, those metrics have the appealing property of not requiring reference summaries.

3.3.2 Quantitative Analysis

To evaluate our QA based metrics, we exploit human judgments obtained for 3 types of automatically generated summaries by Paulus et al. (2017) on 100 samples of the CNN/Daily Mail summarization dataset (see detail in section 3.4.1), in terms of readability (how well written the summary is) and relevance (how well does the summary capture the important parts of the article). The summaries are generated by the three different systems proposed in the original work. Those samples have been scored, via Amazon Mechanical Turk, for Readability and Relevance (scores from 1 to 10 for both metrics).

Baseline Metrics We consider the following metrics as baseline to compare our proposed Question-Answering based Metrics: ROUGE, Novelty and Language Modeling, as described in Section 3.3. We also consider TextRank: Automated summarization started with the development of extractive text summarization models. Many unsupervised models, that aim at computing a score between a sentence and document(s) were developed – the score attempting to reflect whether the sentence should be selected for building a summary (Nenkova 2011).

	Readability	**Relevance**
Readability	1.0	0.77 **
Relevance	0.77 **	1.0
ROUGE-1 (sup)	0.14 *	0.18 **
ROUGE-2 (sup)	0.12 *	0.18 **
ROUGE-L (sup)	0.13 *	0.18 **
TextRank (unsup)	0.14 *	0.13 **
Novelty (unsup)	-0.13 *	-0.1 *
Bert LM (unsup)	0.21 **	0.08 *
QA_{fscore} (sup)	0.14 *	0.19 **
QA_{conf} (sup)	0.19 **	0.23 **
QA_{fscore} (unsup)	0.08	0.2 **
QA_{conf} (unsup)	0.33 **	0.31 **

Table 3.1: Spearman's ρ for the different metrics w.r.t. Readability and Relevance (*: $p < .05$, **: $p < .005$).

Such scores can thus be used as a proxy of the summary quality. We chose TextRank (Mihalcea and Tarau 2004) – an extractive non-parametric summarization system inspired by PageRank (Page et al. 1999) – since it is well performing for extractive tasks and could be easily adapted for our needs. The algorithm builds a graph of sentences within a text based on their co-occurrences. Then, it assigns an importance score for each sentence based on a random walk on the resulting graph. The most important elements of the graph are considered as the ones that best describe the text. As a derivative usage, we propose to consider these importance scores to assess the quality of abstractive summaries in our study. This metric is referred to as TextRank in the following.

In Table 3.1, we report Spearman's rank correlations on this data, where we compare summaries rankings obtained according to the assessed metrics. Scores render the ability of the various metrics to reproduce human preferences (in terms of readability and relevance). First, we observe that readability and relevance are naturally intertwined: intuitively, an unreadable summary bears very little information, one of the facts that explains the high correlation between *readability* and *relevance*.

From this correlation analysis against human judgments, we observe that, as expected, the Language Model metric captures *readability* better than ROUGE, while falling short on *relevance*.

On the other hand, the results obtained using the proposed QA-based metrics indicate their potential benefits especially under the *unsupervised* setting, with QA_{conf} and QA_{fscore} capturing *readability* and *relevance* better than all the other

reported metrics, including ROUGE. We thus conclude that the proposed metrics, which favorably correlate with *readability* and *relevance* wrt human evaluation, are worth of a deeper experimental investigation.

3.3.3 Learned Metric

Since metrics appear to correlate differently with human assessments, we propose now to assess if they could be combined efficiently. To that aim, we leverage the qualitative data obtained by Paulus et al. (2017) – which compounds to 50 samples evaluated by annotators in terms of *readability* and *relevance* – to *learn* an aggregate metric for evaluation. We use a Ridge regression (with a regularization $\lambda = 1$) to learn to predict the geometric mean of readability and relevance from the metrics defined above. The geometric means was chosen since we want the generated summary to be both readable and relevant.

We randomly sampled 50% of the data to fit the linear model with various subsets of our base metrics. Then, we measured the correlation w.r.t. the expected geometric mean on the remaining 50% data. We performed this procedure 1000 times. Our experiments show that the best performing set of metrics consists of ROUGE-L in conjunction with QA_{conf} and QA_{fscore}, both computed at an article-level, and hence unsupervised.

This learned metric is defined as (with *unsup* versions of QA-based scores):

$$\alpha ROUGE_L + \beta QA_{conf} + \delta QA_{fscore} \tag{3.1}$$

with $\alpha = 0.8576$, $\beta = 2.274$ and $\delta = 0.6413$.

3.4 Question Answering as a Reward to Reinforce

In this Section, we evaluate the effect of substituting the ROUGE reward in the reinforcement-learning model of (Paulus et al. 2017) by our proposed QA-based metrics (section 3.3). We, moreover, study the effect of using metrics that do not necessitate human-generated summaries.

3.4.1 Data Used

Task-specific corpora for building and evaluating summarization models associate a human-generated reference summary with each text provided. We resort

to the CNN/Daily Mail (CNN-DM) dataset (Hermann et al. 2015; Nallapati et al. 2016) for our experiments. It includes 287,113 article/summary pairs for training, 13,368 for validation, and 11,490 for testing. The summary corresponding to each article consists of several bullet points displayed on the respective news outlet webpage. In average, summaries contain 66 tokens ($\sigma = 26$) and 4.9 bullet points. Consistently with See et al. (2017) and Gehrmann et al. (2018), we use the non-anonymized version of the dataset, the same training/validation splits, and perform truncation of source documents and summaries to 400 and 100 tokens, respectively.

To assess the possible benefits of reinforcing over the proposed QG-based metric, which does not require human-generated reference summaries, we employ TL;DR[1], a large-scale dataset for automatic summarization built on social media data, compounding to 4 Million training pairs (Völske et al. 2017b). Both CNN-DM and TL;DR datasets are in English.

3.4.2 Models

For all our experiments, we build on top of the publicly available OpenNMT implementation[2], consistently with Gehrmann et al. (2018) to which we refer to as a baseline. The encoder is composed of a one-layer bi-LSTM with 512 hidden states, and 512 hidden states for the one-layer decoder. The embedding size is set at 128. The model is trained with Adagrad, with an initial learning rate of 0.15, and an initial accumulator value of 0.1. We continue training until convergence; when the validation perplexity does not decrease after an epoch, the learning rate is halved. We use gradient-clipping with a maximum norm of 2.

Gehrmann et al. (2018) showed that increasing the number of hidden states leads to slight improvements in performance, at the cost of increased training time; thus, as reinforcement learning is computationally expensive, we build on top of the smallest model – nonetheless, we include the largest model by Gehrmann et al. (2018) in our discussion of results.

All the experimented reinforcement approaches use the mixed training objectives defined in equation 3.2, with the ML part corresponding to the previously described baseline model pretrained on the CNN-DM dataset. Compared models differ on the considered reward signals. They also differ on their use of additional unsupervised data, either *In-Domain* or *Out-of-Domain*, as discussed below.

[1] https://tldr.webis.de
[2] http://opennmt.net/OpenNMT-py/Summarization.html

3.4.3 Mixed Training Objectives

Paulus et al. (2017) consider a mixed loss L_{ml+rl} combining supervised and reinforcement learning schemes:

$$L_{ml+rl} = \gamma L_{rl} + (1 - \gamma) L_{ml} \tag{3.2}$$

where L_{rl} is the the reinforcement loss (see Equation 2.8), and L_{ml} the maximum likelihood. As ROUGE is the most widely used evaluation metric, Paulus et al. (2017) used ROUGE-L as the reward r for the L_{rl} function and tested the following three different setups:

- *ML*: the model trained with L_{ml} ($\gamma = 0$);
- *RL*: the model trained with L_{rl} ($\gamma = 1$);
- *ML+RL*: the model trained with L_{ml+rl} ($\gamma = 0.9984$).

The human evaluation conducted on the three models shows that *RL* performs worse than *ML*, and *ML+RL* performs best for both *readability* and *relevance*. The authors also conclude that "despite their common use for evaluation, ROUGE scores have their shortcomings and should not be the only metric to optimize on summarization model for long sequences", which is translated in the very high optimal γ.

We show in the following that using a more sensible metric to optimize leads to a better model, and to a lower γ.

3.4.3.1 Reward Signals

The three reward signals used throughout our experiments, are detailed below:

1. **ROUGE**: We use only ROUGE-L as reward signal within the baseline architecture, consistently with Paulus et al. (2017);
2. **QA**$_{learned}$: Conversely, we compute the reward by applying the learned coefficients to the three components of the learned metric, as obtained in Section 3.3.3.
3. **QA**$_{equally}$: We apply the mixed training objective function, using as a reward the three metric components of the learned metric (ROUGE-L, QA$_{conf}$, and QA$_{fscore}$) equally weighted: this corresponds to setting a value of 1 for α, β and δ in Eq. 3.1. This allows to see to which extent learning is sensitive to fitting human assessments.

For (2) and (3), we set γ (Eq. 3.2) to 0.5[3]. This shows that, compared to (Paulus et al. 2017), we do not need to use NLL to avoid the model from generating unreadable summaries.

3.4.3.2 In-Domain vs Out-of-Domain

We experiment with the proposed QA_{conf} and QA_{fscore} metrics in an unsupervised fashion, as they can be computed at article level – *i.e.* without accessing the reference human-generated summaries. We investigate the potential benefits of using this approach both *in-domain* and *out-of-domain*: for the former, we resort to the test set of the CNN-Daily Mail (CNN-DM) dataset; for the latter, we leverage the TL;DR corpus.

As the CNN-Daily Mail is built from mainstream news articles, and the TL;DR data comes from social media sources, we consider the latter as out-of-domain in comparison. From the latter, which includes circa 4 million samples, we randomly draw sample subsets of size comparable with CNN-DM for training, validation and testing splits.

Due to computational costs, we restrict these experiments to the model trained under reinforcement using the $\text{QA}_{learned}$ metric. Under this setup, the model has access at training time both to:

- *supervised* samples for which a reference summary is given (and thus all metrics, including ROUGE and NLL, can be computed as a training objective), coming from the training set of CNN-Daily Mail corpus ;

- *unsupervised* samples, for which no reference is available, thus allowing to only compute QA_{conf}(unsup) and QA_{fscore}(unsup). Three unsupervised settings are considered in the following:

  TL;DR, corresponding to the *out-of-domain* setting where we use articles from the *TL;DR* dataset;

  CNN-DM (VAL), corresponding to an *in-domain* setting where we use texts from the validation set from the CNN/Daily Mail dataset;

  and, *CNN-DM (TEST)* for an *in-domain* setting where we use the articles from the test set (thus containing texts used for evaluation purposes).

While all the data is from the CNN-DM train dataset in the *supervised* setups, for the *unsupervised* setups, we set the proportion of unsupervised data to 50% (either CNN-DM VAL, CNN-DM TEST for *in-domain* or TL;DR for *out-of-domain*

[3]We have run experiments with $\gamma = 0.5$, and $\gamma = 0.9984$ as Paulus et al. (2017); we report here the best performance which was obtained with the former.

	R-1	R-2	R-L	QA_{fscore}	QA_{conf}
See et al. (2017)	39.53	17.28	36.38	-	-
Gehrmann et al. (2018)	41.22	**18.68**	38.34	-	-
ML+RL Paulus et al. (2017)	39.87	15.82	36.90	-	-
RL Paulus et al. (2017)	41.16	15.75	39.08	-	-
Pasunuru and Bansal (2018)	40.43	18.00	37.10	-	-
Y.-C. Chen and Bansal (2018)	40.88	17.80	38.54	-	-
baseline	42.24	17.78	37.44	14.91	40.12
+ ROUGE	**45.62**	16.30	**41.60**	13.64	37.90
+ $QA_{equally}$	43.36	18.06	38.33	16.06	41.01
+ $QA_{learned}$	42.71	17.81	37.94	15.19	41.39
+ $QA_{learned}$ + TL;DR	42.75	17.57	37.88	15.75	41.54
+ $QA_{learned}$ + CNN-DM (VAL)	43.00	17.66	38.23	16.16	41.75
+ $QA_{learned}$ + CNN-DM (TEST)	42.74	17.25	37.96	**16.17**	**42.14**

Table 3.2: Comparison with previous works. On top, we report the results obtained by Gehrmann et al. (2018) using their largest architecture, as well as those by See et al. (2017). Next, we report results recently obtained by reinforcement learning approaches. Finally, we indicate the scores obtained by our baseline – the "small" model by Gehrmann et al. (2018) – and the six reinforced models we build on top of it.

data). Thus, for 50% of the data, the model has access only to the QA_{conf} and QA_{fscore} reward signals, since the ROUGE-L reward can only be computed on *supervised* batches.

Therefore, for all the unsupervised setups, in order to keep consistency in the reward signal throughout the training, we multiply by a factor of 2 the weight associated with ROUGE-L when this reward is computable, and set it to 0 otherwise.

3.4.4 Results

In Table 3.2, we report the results obtained from our experiments in comparison with previously proposed approaches. We observe that, unsurprisingly, reinforcing on ROUGE-L allows to obtain significant improvements over the state-of-the-art, in terms of ROUGE but at the cost of lower QA-based metrics. Conversely, reinforcing on the proposed metric improves consistently all its components (ROUGE-L, QA_{conf} and QA_{fscore}).

However, increasing the reward does not necessarily correlate with better summaries. The human inspection as reported by (Paulus et al. 2017) shows that the

generated summaries reinforced on ROUGE-L are consistently on the low end in terms of readability and relevance.

A closer inspection of the generated summaries revealed that the sequences generated by this model seem to qualitatively degrade as the number of produced tokens grows: they often start with a reasonable sub-sequence, but quickly diverge towards meaningless outputs. This can be explained by the aforementioned drawbacks of ROUGE, which are likely amplified when used both as evaluation and reward: the system might be optimizing for ROUGE, at the price of losing the information captured with the NLL loss by its language model.

We hence conducted a human evaluation for the different setups, reported in Table 3.3, assessing their outputs for *readability* and *relevance* in line with Paulus et al. (2017). We randomly sampled 50 articles from the CNN-DM test set; since the learned metric used in our experiments is derived from the subset manually evaluated in Paulus et al. (2017) we ensured that there was no overlap with it. For each of those 50 articles, three English speakers evaluated the summaries generated by the 7 different systems reported in Table 3.2.

We observe that reinforcing using the proposed metric – which includes QA based metrics, leads to comparable performance in terms of ROUGE w.r.t. state-of-the-art approaches, while clear benefits emerge from the results of the human evaluation: a significant improvement in terms of relevance, particularly when leveraging *in-domain* data in an *unsupervised* setup. Not surprisingly, we observe an improvement for our model when reinforced through the learned metric compared to the one equally weighted. The slightly lower relevance scores observed for the $QA_{learned}$ w.r.t. $QA_{equally}$ are consistent with the lower ROUGE-L and QA_{fscore} reported in Table 3.2. This is explained by the lower coefficients for ROUGE-L and QA_{fscore} (see 3.3.3), and the relatively stronger correlation of those two metrics with *relevance* (see Table 3.1).

Consistently with the figures reported in Table 3.2, the human evaluation results – reported in Tables 3.3 and 3.4 – confirm the progressive improvements of our different proposed models when using unsupervised data closer to the test set documents:

- adding *unsupervised* data from the out-of-domain TL;DR brings a slight improvement using $QA_{learned}$;
- when it comes to the same domain (*i.e.* CNN-DM validation) the improvements increase;
- finally, when unsupervised samples come from the same set as those used for testing, we observe even better results.

	Readability	**Relevance**
human reference	7.27*	7.4**
baseline	7.07	5.82
+ ROUGE	2.14**	5.48**
+ $QA_{equally}$	5.94**	6.34**
+ $QA_{learned}$	6.96	6.21**
+ $QA_{learned}$ + TL;DR	6.60*	6.26**
+ $QA_{learned}$ + CNN-DM (VAL)	6.40*	6.75**
+ $QA_{learned}$ + CNN-DM (TEST)	6.89	6.80**

Table 3.3: Human assessment: two-tailed t-test results are reported for each model compared to the baseline ($* : p < .01$, $** : p < .001$).

	human reference	baseline	+ ROUGE	+ $QA_{equally}$	+ $QA_{learned}$	+ $QA_{learned}$ + TL;DR	+ $QA_{learned}$ + CNN-DM (VAL)	+ $QA_{learned}$ + CNN-DM (TEST)
baseline	* / **	-						
+ ROUGE	** / **	** / **	-					
+ $QA_{equally}$	** / **	** / **	** / **	-				
+ $QA_{learned}$	** / **	- / **	** / **	** / -	-			
+ $QA_{learned}$ + TL;DR	** / **	* / **	** / **	** / -	* / -	-		
+ $QA_{learned}$ + CNN-DM (VAL)	** / **	* / **	** / **	** / *	** / **	- / *	-	
+ $QA_{learned}$ + CNN-DM (TEST)	** / **	- / **	** / **	** / *	- / **	- / *	* / -	-

Table 3.4: Human assessment: two-tailed t-test results are reported for each model pair for Readability / Relevance ($* : p < .01$, $** : p < .001$).

These results show that using the proposed QA-based metrics, that do not depend on reference summaries, allows to leverage raw text data; and, that fine-tuning (without supervision) on the documents to be summarized is beneficial.

To elaborate further, we notice that applying the learned coefficients for 3.1 to the results obtained by models reinforced on $QA_{learned}$ and $QA_{equally}$, see Table 3.2, we obtain very similar scores (namely, 136.43 for $QA_{equally}$ and 136.4 for $QA_{learned}$). However, the qualitative analysis reported in Tables 3.3 and 3.4 shows that while they perform similarly in terms of *relevance*, a significantly lower score for *readability* is obtained using $QA_{equally}$. This can be explained by the stronger weight of ROUGE_L for this setup, a fact which might lead to a degradation of the quality of the output consistently with the observations reported in (Paulus et al. 2017) as well as in our ROUGE experiment.

Another observation from Tables 3.3 and 3.4 is that while $QA_{learned}$ performs significantly better in term of *readability* than $QA_{learned}$ + CNN-DM (VAL), the opposite holds for *relevance*. This could be explained by the setup difference during training: as detailed in section 3.4.3.2, for unsupervised setups (*i.e.* $QA_{learned}$ + CNN-DM (VAL)) only the QA-based metrics are computed for the portion of data for which no reference is available. While testing (TEST) and validation (VAL) splits come the same dataset (CNN-DM), we observe that using the samples from TEST in an unsupervised fashion allows for maintaining comparably high *relevance* compared to $QA_{learned}$ + CNN-DM (VAL), while also obtaining similar *readability* to $QA_{learned}$. This shows the possible benefits that can be obtained by exposing the model to the evaluation data in unsupervised setups. To further study our unsupervised metrics, we performed additional experiments on the TL;DR corpus. We observed more than one absolute point of improvement w.r.t CNN-DM TEST in terms of ROUGE-L, QA_{fscore} (unsup) and QA_{conf} (unsup).

This indicates that the proposed unsupervised metrics allow the model to better transfer to new domains such as TL;DR. These results pave the way for leveraging large numbers of texts, in a self-supervised manner, to train automatic summarization models.

3.5 Conclusions

We have presented the analysis of novel QA-based metrics[4], and have shown promising results when using them as a reward in a RL setup. Crucially, those metrics do not require a human reference, as they can be computed from the text to be summarized. From our experiments this proves particularly beneficial, allowing to leverage both *in-domain* and *out-of-domain* unlabeled data.

The promising results obtained indicate a path towards partially self-supervised training of summarization models, and suggest that progress in automated question generation can bring benefits for automatic summarization.

[4]A python package is available at https://www.github.com/recitalAI/summa-qa.

CHAPTER 4

QUESTEVAL: QUESTIONS TO EVALUATE

abstract

Summarization evaluation remains an open research problem: current metrics such as ROUGE are known to be limited and to correlate poorly with human judgments. To alleviate this issue, we proposed SumQA in the previous Chapter, an evaluation metric which relies on question answering models to assess whether a summary contains all the relevant information in its source document. Though promising, the proposed approach fails to correlate significantly better than ROUGE with human judgments, preventing it to be used as a reliable evaluation metric. It nevertheless provided a better RL reward than ROUGE.

In this Chapter, we extend SumQA and propose a unified framework, named QUESTEVAL. In contrast to established metrics such as ROUGE or BERTScore, QUESTEVAL does not require any ground-truth reference. Nonetheless, QUESTEVAL substantially improves the correlation with human judgments over four evaluation dimensions (consistency, coherence, fluency, and relevance), as shown in extensive experiments. Codes and models are publicly available.[1]

- Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang. "QuestEval: Summarization Asks for Fact-based Evaluation" Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2021.

4.1 Introduction

While we have presented SumQA in the previous chapter, an alternative version has been proposed later by A. Wang et al. (2020).

Although these works have introduced an interesting and novel method to evaluate summarization, with encouraging preliminary results, none of those metrics is found to perform better than ROUGE on a large study conducted by Fabbri et al. (2020): automatic evaluation of summarization systems remains an open research problem (Kryscinski et al. 2019a).

[1] https://github.com/ThomasScialom/QuestEval

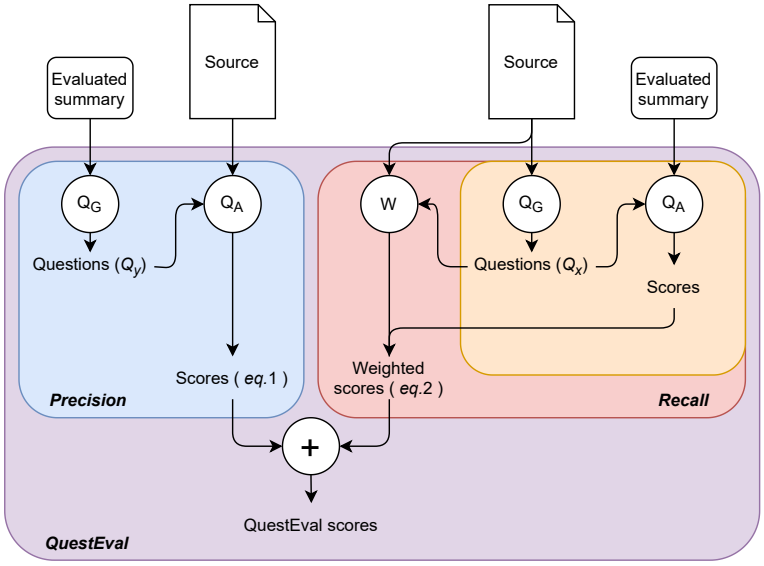


Figure 4.1: Illustration of the QUESTEVAL framework: the blue area corresponds to the precision-oriented framework proposed by A. Wang et al. (2020). The orange area corresponds to the recall-oriented SummaQA. We extend it with a weighter component for an improved recall (red area). The encompassing area corresponds to our proposed unified approach, QUESTEVAL.

Inspired by these works, and motivated to take up the challenge of summarization evaluation, we propose QUESTEVAL, a new *reference-less* metric, which is found to improve the correlation with humans judgments. Our contributions are as follows:

- We show that, by unifying the precision and recall-based QA metrics (as defined in Section 4.3), we obtain a more robust metric;
- We propose a method to learn the saliency of the generated queries, allowing to integrate the notion of information selection;
- We evaluate QUESTEVAL on two corpora containing annotated summaries from CNN/Daily Mail and XSUM datasets. The proposed metric obtains state-of-the-art results in terms of correlation with humans judgments, over all the evaluated dimensions. Notably, QUESTEVAL is effective at measuring factual consistency, a crucial yet challenging aspect for summarization.

4.2 A Question-Answering based Framework

The QuestEval framework we introduced in this Chapter accounts for both factual consistency and relevance of the generated text, without requiring any human reference. QUESTEVAL consists of a Question Generation component Q_G and a Question Answering component Q_A, described in this section and depicted in Figure 4.1.

4.2.1 Question Answering

There has been significant progress on factoid question answering after the experiments conducted in the previous chapter. Models have obtained human-level performance on benchmarks such as SQuAD (Rajpurkar et al. 2016). Leveraging on these advancements, our Q_A component consists of a pretrained T5 model (Raffel et al. 2019), which generates answers from a source document given a question to answer. In the following, we refer to $Q_A(r|T, q)$ as the probability of the answer r to question q on a text T, and $Q_A(T, q)$ as the answer greedily generated from the model.

When a summary is evaluated, there is no guarantee that it contains the answer. Therefore, it is crucial for the QA model to be able to predict when a question is unanswerable. Our Q_A component thus includes the *unanswerable* token, that we denote ϵ, among its possible outputs.

4.2.2 Question Generation

For the QG component, rather than simply masking tokens as described in the previous Chapter, we draw on recent work on neural answer-conditional question generation (Q. Zhou et al. 2017). The component also consists of a T5 model, fine-tuned to maximize the likelihood of human questions, given the corresponding answer and source document.

At test time, given a source document or generated summary, we first select a set of answers from the text to condition the QG model on. Following A. Wang et al. (2020), we consider all the named entities and nouns from the source document as answers. Then, for each selected answer, we generate a question via beam search.[2]

[2]We experimented with nucleus sampling (Holtzman et al. 2019) to increase diversity of the questions, with no success.

We filter out every question for which the QA model predicts an incorrect answer. Based on this, we denote $Q_G(T)$ the set of question-answer pairs (q, r) for a text T such that $Q_A(T, q) = r$.

4.3 The QuestEval metric

Now that we have specified how to generate questions and answer them, we describe the metric. In the following, D and S are two sequences of tokens with D denoting the source document and S the corresponding evaluated summary.

4.3.1 Precision

A summary is deemed inconsistent w.r.t. its source text if, given a question, the answer differs when conditioned on S or D. Therefore, we define the precision score for the evaluated summary as:

$$Prec(D, S) = \frac{1}{|Q_G(S)|} \sum_{(q,r) \in Q_G(S)} F1(Q_A(D, q), r) \tag{4.1}$$

The F1 score is a standard metric for evaluating factoid question answering models, and measures the overlap between the predicted answer and the corresponding ground truth. It outputs 1 for an exact match between both answers and 0 if there is no common token. This definition of factual consistency corresponds to the frameworks proposed by A. Wang et al. (2020) and Durmus et al. (2020).

4.3.2 Recall

While a summary should contain only factual information (precision), it should also contain the most important information from its source text (recall). Extending SumQA by introducing a query weighter W, we define recall as:

$$Rec(D, S) = \frac{\sum_{q,r \in Q_G(D)} W(q, D)(1 - Q_A(\epsilon | S, q))}{\sum_{q,r \in Q_G(D)} W(q, D)} \tag{4.2}$$

where $Q_G(D)$ is the set of all question-answer pairs for the source text D, and $W(q, D)$ is the weight of query q for text D.

While F1 is accurate, measuring the overlap between the predicted answer and the corresponding ground truth (Rajpurkar et al. 2016). However, an answer

could be correctly expressed in different ways, e.g. "ACL" and "Association for Computational Linguistics". Unfortunately, the F1 score is 0 in this example.

To sidestep this issue, SumQA use the QA confidence of answerability, i.e. $1 - Q_A(\epsilon)$, rather than F1 score. Defining recall this way allows to measure answerability independently of the way the answer is expressed, but does not take into account possible model hallucinations, i.e. the summary could answer the question incorrectly.

Conversely, when we assess factual consistency, it is not enough for a question from the summary to be answerable from the source document. The two answers to this question should *also share the same meaning* to be factually consistent. While using answerability allows for more true positives (e.g. "ACL" in the example above), for precision it is crucial to detect true negatives. This motivates our use of the F1 score in this case, similar to A. Wang et al. (2020).

Query Weighting In SumQA, all questions are considered equally important, i.e. the weight $W(q, D) = 1$ for every query $q \in Q_G(D)$. However, since a summary necessarily has a constrained length, an effective summary should contain the most important information from the source. To account for this, we introduce a question weighter, which is trained to distinguish *important* questions from *anecdotal* ones. We leverage existing summarization datasets to create training data for the weighter: given a source document D, each question $q \in Q_G(D)$ is labeled as *important* if the corresponding human summary contains the answer, as computed by the QA component applied on the summary (i.e. $Q_A(S, q) \neq \epsilon$).

$W(q, D)$ denotes the probability that q is important for D. Note that the question weighter only concerns recall, and therefore is not applied when computing precision.

4.3.3 Unifying Precision and Recall

The final QUESTEVAL score accounts for both the precision and recall by computing their harmonic mean (i.e. the F-Score): $2\frac{Prec \cdot Rec}{Prec + Rec}$. The QUESTEVAL score is thus directly comparable with existing evaluation metrics, such as ROUGE or BLEU, as it lies in the same numerical range.

	#Ref	Consistency	Coherence	Fluency	Relevance	Average
ROUGE-1	11	18.1	20.1	14.9	35.6	22.2
ROUGE-L	11	15.7	15.6	13.8	33.4	19.6
METEOR	11	3.3	2.9	7.1	-0.5	3.2
BLEU	11	17.5	22.	13.7	35.6	22.2
BERTScore-f	11	20.3	18.5	21.6	31.9	23.1
ROUGE-1	1	11.0	9.8	7.5	18.9	11.8
ROUGE-L	1	8.2	7.3	5.7	13.5	8.7
BLEU	1	8.9	3.9	4.0	12.7	7.4
BERTScore-f	1	8.7	9.8	10.6	17.9	11.8
SummaQA	0	8.3	8.0	-2.9	26.2	9.9
QAGS (our impl.)	0	20.4	7.7	16.8	9.1	13.7
QUESTEVAL$_{W=uniform}$	0	43.7	22.9	28.2	37.5	33.1
w/o QA neg sampl.	0	42.5	22.5	27.7	37.2	32.4
QUESTEVAL$_{W=learned}$	0	42.0	**24.0**	28.4	**39.2**	**33.5**
Precision Only	0	**46.5**	14.0	**30.9**	22.2	28.4
Recall Only	0	30.5	22.6	19.2	37.6	27.5
NewsQA	0	40.6	22.8	27.7	38.3	**33.5**

Table 4.1: Summary-level Pearson correlation coefficients for various dimensions between automatic metrics and human judgments on SummEval. The top section corresponds to correlations for metrics computed on 11 reference summaries, as reported in Fabbri et al. (2020). The second section corresponds to these metrics, but given only one reference: each of the 11 available references is used alone, and the correlations averaged. The third section corresponds to the QA-based baselines. The bottom section corresponds to QUESTEVAL and its ablations.

4.4 Experiments

4.4.1 Summarization Datasets

To evaluate QUESTEVAL, we measure its correlation with human judgments on different datasets:

SummEval Released by Fabbri et al. (2020), it is one of the largest human-annotated datasets for summarization. Derived from CNN/Daily Mail (Nallapati et al. 2016), it consists of 12,800 summary level annotations. To ensure diversity, the summaries were generated from 16 different summarization models, including extractive and abstractive architectures. To assess quality, three experts annotated four dimensions: *i) Consistency*: the proportion of facts in the summary

corresponding to facts in the original text; *ii) Coherence*: how well-structured and well-organized is the summary; *iii) Fluency*: how fluent the summary is to read; and, *iv) Relevance*: the ratio between important and excess information in the summary.[3]

QAGS-XSUM A. Wang et al. (2020) released a subset of 239 BART outputs fine-tuned on XSUM (Narayan et al. 2018a).[4] Three annotators measured the consistency of each summary.

4.4.2 Question Answering & Generation

To train our Q_G and Q_A models, we used the SQuAD-v2 (Rajpurkar et al. 2018) factoid question answering dataset: it is composed of (paragraph, question, answer) triplets, and includes unanswerable questions. Note that QG can be seen as the dual task for QA: any QA dataset can be used for QG by switching the generation target from the answer to the question.

Lastly, we found it helpful to train our QA model using additional synthetic unanswerable questions. This is done by considering a shuffled version of the dataset, where each question is randomly assigned to a paragraph from another triplet of the dataset. We consider these additional samples, with flipped contexts, as unanswerable. All experiments in this work, except otherwise specified, use this additional data to improve identification of unanswerable queries.

4.4.3 Results

In Tables 4.1 and 4.2 we report the results for QUESTEVAL, along with several ablations. $W = uniform$ corresponds to setting all questions weights equal. Conversely, $W = learned$ corresponds to the weights learned as detailed in §4.3.2. We report the recall and precision components separately. Finally, for $W = learned$, we also report the results given a QA and QG component trained on NewsQA (Trischler et al. 2016), i.e. a different domain than SQuAD.

In Table 4.1, we observe that, amongst existing metrics, BERTScore achieves the best average Pearson correlation with human judgements (23.1), slightly above ROUGE-1 (22.2) and BLEU (22.2). These correlations are obtained when providing *no less than 11 gold references*, and averaging results. Given a single reference, all these correlations are halved. Most of the large scale datasets provide only

[3] See 4.3 Human Annotations in Fabbri et al. (2020) for more details.
[4] Note that XSUM provides more abstractive summaries than those of CNN/Daily Mail.

Metric	Consistency
ROUGE-1	13.2
ROUGE-L	8.9
METEOR	10.0
BLEU	5.6
BERTScore	2.5
SummaQA	-
QAGS	17.5
QUESTEVAL$_{W=uniform}$	30.4
w/o QA neg sampl.	28.5
QUESTEVAL$_{W=learned}$	29.0
Precision Only	**32.7**
Recall Only	13.9
NewsQA	28.2

Table 4.2: Summary-level Pearson correlation coefficients for Consistency between various automatic metrics and human judgments on QAGS-XSUM. The top section corresponds to correlations for diverse metrics computed on one reference summary. The middle section corresponds to QA-based baselines. The bottom section corresponds to this work.

one reference per example in their test set (e.g. CNN/Daily Mail and XSUM), a fact that highlights the importance of searching for more reference-efficient alternatives.

With regards to sample efficiency, QA-based metrics *do not require any references*. We expect Relevance to be better measured by Recall oriented metrics, and less so for Consistency. This is confirmed in the results, where SummaQA correlates better with Relevance than Consistency (26.2 vs 8.3), and vice versa for QAGS (9.1 vs 20.4). By unifying and extending the two, QUESTEVAL allows to take both dimensions into account, improving the average correlation by 18% (28.4 to 33.5).

The dimension benefiting the most from our question weighter is Relevance (+4%, from 37.5 to 39.2), indicating that our classifier learns *which questions target important information*. We discuss this aspect more in depth in section 4.4.4.

Finally, we do not observe significant differences when using a QA and QG specifically trained on NewsQA. In conclusion, compared to the other metrics, the improvement is remarkable (33.5 vs 11.8 for BERTScore), allowing better evaluations of the systems while not even requiring references.

4.4.4 Discussion

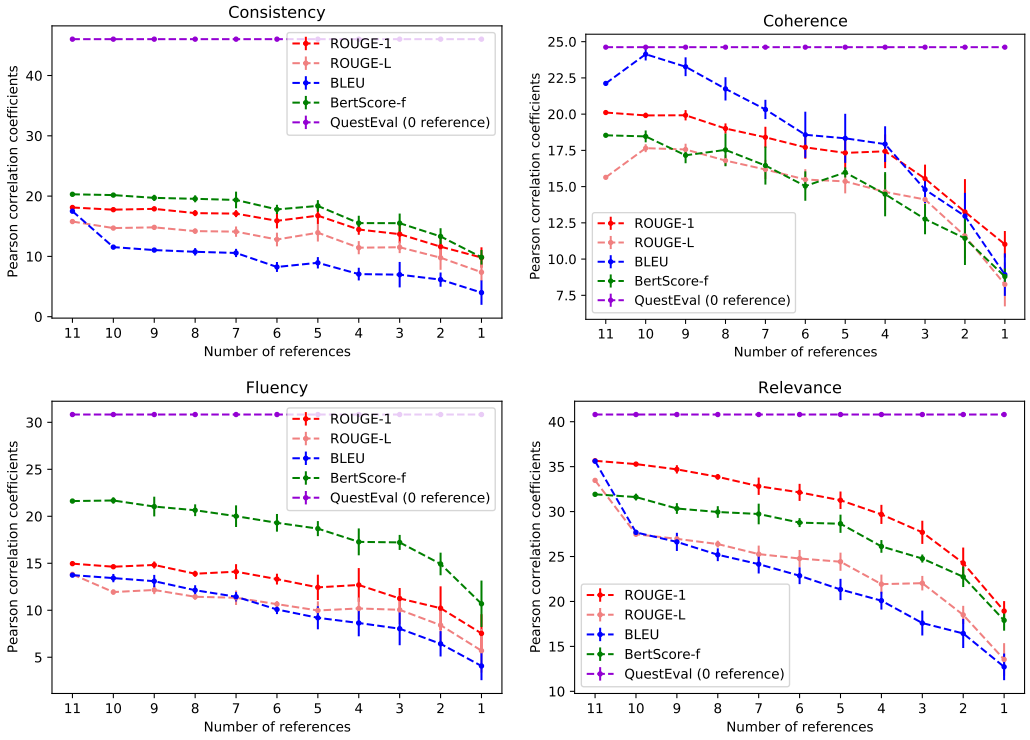


Figure 4.2: Variation of the Pearson correlations between various metrics and humans, versus the number of references available. QUESTEVAL is constant, since it is independent from the references.

important	*answered*	Relevance Corr.
✓	✓	37.6
✓	✗	-33.5
✗	✓	-5.7

Table 4.3: Pearson correlation coefficients between human judgments (for Relevance) and the percentage of *important* and/or *answered* questions, on SummEval data.

Reference-less One of the main limitations for the current metrics is that they require gold references to compute similarity scores. However, many possible summaries are valid for one source document. We argue that the universe of correct outputs is much larger than in other generation tasks such as machine translation. This explains why the correlations with human judgments are largely reduced when they are computed with only one reference instead of 11 (see Table 4.1: BERTScore-f drops from 23.1 to 11.8 in average, and other metrics likewise). Unfortunately, assuming the availability of as many as 11 gold references is not realistic in most scenarios, due to the cost of obtaining reference summaries.

To complement Table 4.1, we report in Figure 4.2 the correlations for the best baselines as we progressively decrease the number of available gold references from 11 to 1. For all four dimensions and all the baselines, we observe that less references result in decreased correlation and increased variance. However, QUESTEVAL *does not require any reference*. Therefore, the improvement over the other metrics grows larger as the number of references used decreases. Furthermore, QUESTEVAL enables the evaluation of systems *even when no gold reference is available*.

Query Weighter There is no unique answer to the question "What makes a good summary?": it depends on the reader's point of view, which makes summarization evaluation challenging. For instance, given a contract, the seller and the buyer could be interested in different information within the same document. To instantiate the weighter W, we learn a specific dataset policy: "what kind of questions are likely answered in the CNN/Daily Mail training summaries?" This is a reasonable heuristic given that editors created the summaries following their specific policy.

To demonstrate the effectiveness of the weighter, we proceed as follows. We first consider that a question $q \in Q_G(D)$, generated on the source document, is *important* if the probability given by the query weighter is above a threshold, i.e. if $W(D, q) > 0.5$. We then say that a question is *answered* if the probability of being

Source Document This is the embarrassing moment a *Buckingham Palace* guard slipped and fell on a manhole cover in front of hundreds of shocked tourists as he took up position in his sentry box. [...] The Guard comprises two detachments, one each for Buckingham Palace and St James's Palace, under the command of the Captain of The Queen's Guard. **Generated Question** Where was the Changing of the Guard held? **Weighter prediction** *Important Question* **Answer Span** Buckingham Palace
Correct Summary The Queen's Guard slipped on a manhole cover during the Changing of the Guard at *Buckingham Palace* last week. [...] **Predicted Answer** Buckingham Palace: ✓
Hallucinated Summary The Queen's Guard slipped on a manhole cover during the Changing of the Guard at *St James's Palace* last week. [...] **Predicted Answer** St James's Palace: ✗
Incomplete Summary The Queen's Guard slipped on a manhole cover during the Changing of the Guard during an embarrassing moment.. [...] **Predicted Answer** Unanswerable: ✗

Table 4.4: Sample output from QUESTEVAL: a generated question, it's predicted importance given a source document; the corresponding predicted answers to the question, for three different summaries.

unanswerable is below a threshold, i.e. $Q_A(\epsilon|S, q) < 0.5$. Therefore, a question can belong to one of four folds, given the two above criteria (*important* and/or *answered*). In Table 4.3, we measure how the percentage of questions belonging to a specific fold correlates with the Relevance dimension for each generated summary on SummEval. We observe that the percentage of questions that are *important* and *answered* correlates positively with Relevance, as opposed to the percentage of questions that are *important* but not *answered*. Finally, the percentage of questions that are *answered* but not *important* does not correlate with Relevance. This indicates that the proposed approach is able to learn what are the questions that should be asked or not.

We emphasize that W is a flexible component of our framework: it can be adapted to specific domains and applications. For instance, one could design a specific W, to focus the evaluation on information about specific entities, such as people or events.

An Explainable Metric One important feature of QUESTEVAL is its explainability. It is straightforward to investigate *1)* what are the important points not answered in the summary and *2)* what are the inconsistencies between the source document and the summary. We illustrate this in Table 4.4, with a source doc-

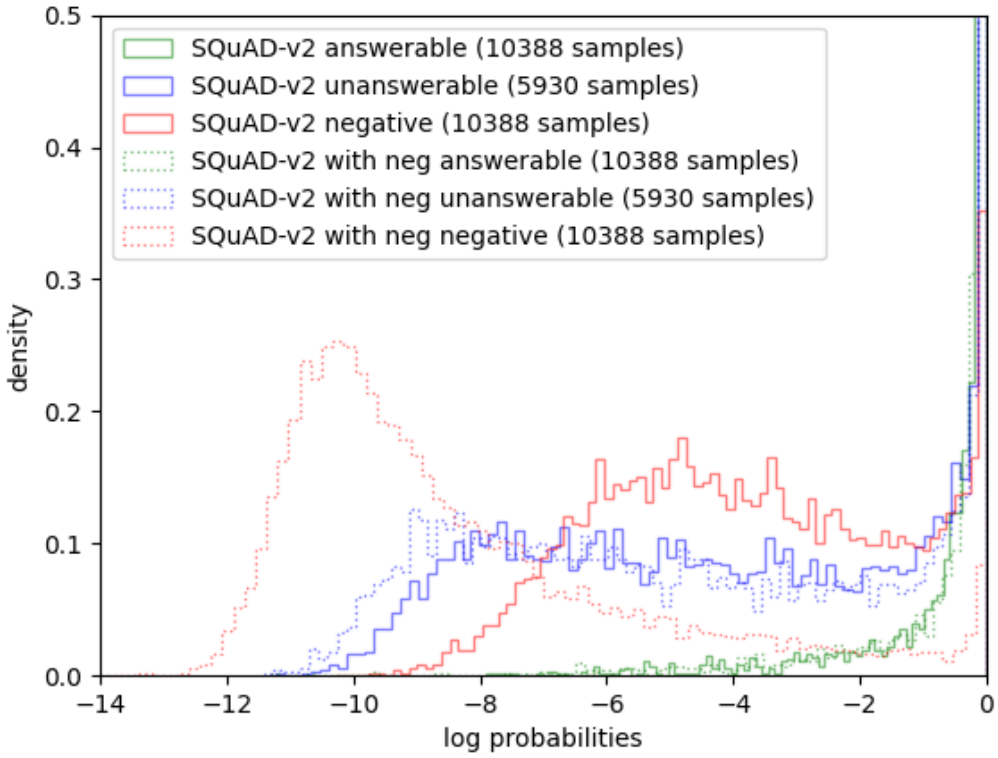


Figure 4.3: Distribution of the log probabilities of answerability – i.e. $\log(1 - Q_A(\epsilon|T, q))$ – for two QA models. 1) solid lines: a model trained on SQuAD-v2 *without* the negative sampled examples. 2) dashed lines: a model trained on SQuAD-v2 *with* the negative sampled examples. The evaluated samples belong to three distinct categories: 1) answerable, 2) unanswerable questions (but present in SQuAD-v2) and 3) the negatively sampled ones (as described in §4.4.1).

ument, from which a question q is generated and answered. According to the weighter W, q is categorized as *important*. Three evaluated summaries are then shown.

The first summary S_{correct} is factually consistent with the source document: the predicted answer $Q_A(S_{\text{correct}}, q)$ corresponds to the source document answer *Buckingham Palace*. The second summary S_{hallu} is inconsistent with the source document: the predicted answer $Q_A(S_{\text{hallu}}, q)$ *does not* correspond to *Buckingham Palace*. Finally, the third summary $S_{\text{incomplete}}$ does not answer the question, i.e. $Q_A(S_{\text{incomplete}}, q) = \epsilon$, and is thus incomplete.

Negative Sampling Effect In Tables 4.1 and 4.2, we observe a decrease of performance when QUESTEVAL uses a QA model trained without negative sam-

pling (see Section 4.4.2), from 33.3 to 32.4 on SummEval and from 30.4 to 28.5 on QAGS-XSUM. In Figure 4.3, we report the distribution of the log probabilities for the two QA models, trained with and without negative sampling. The QA model exposed to the negative sampling during training, learns to separate better the negative sampled questions (for negative, i.e. red lines, the dashed line is more on the left than the solid line).

Indeed, the unanswerable questions of SQuAD-v2 were written adversarially by crowd-workers, to look similar to answerable ones. However, in the context of QUESTEVAL, unanswerable questions are not adversarial: it simply often happens that the summary does not contain the answer. Therefore, QUESTEVAL deals in practice with unanswerable questions that look like those built with negative sampling, rather than adversarial ones. This may explain the improvement of a QUESTEVAL with a QA model trained with negative sampling.

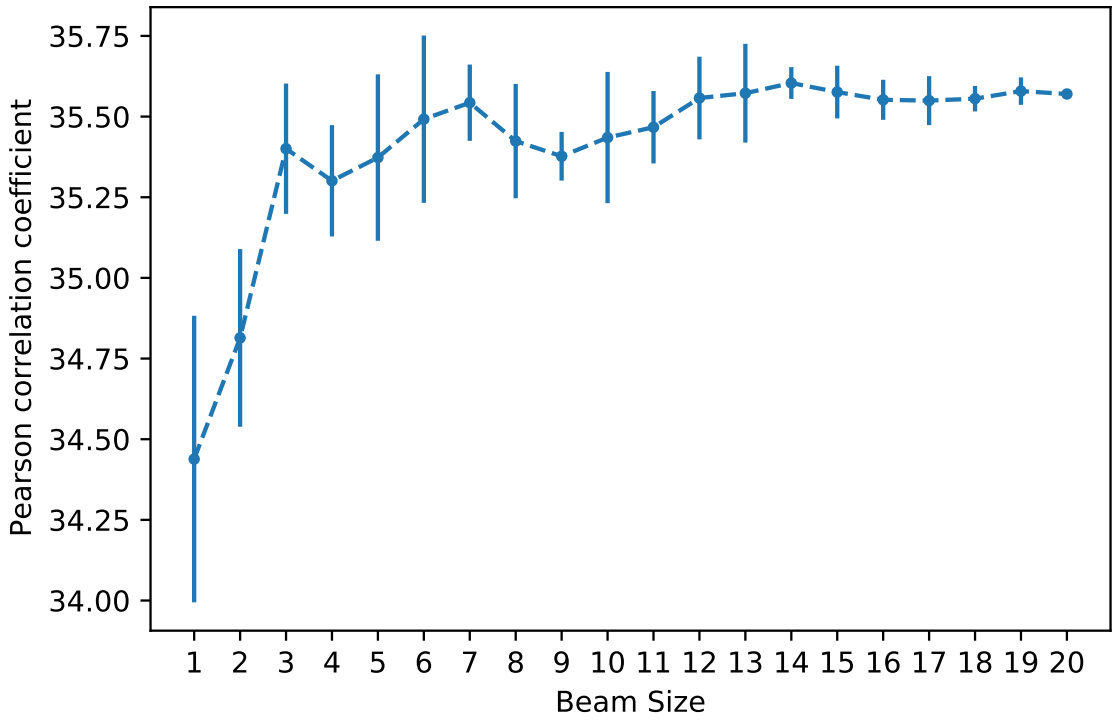


Figure 4.4: Pearson correlation with humans on SummEval w.r.t. the QG beam size.

Computational Complexity Following A. Wang et al. (2020), we generate the questions with $K = 20$ beams during decoding and we keep all the different versions of the questions in the latter steps, which improves correlations. However, the downside of this is the inference time which increases linearly w.r.t the beam size. To be widely adopted, a metric should not only correlate with human

judgment, but also be computationally efficient. In Figure 4.4 we show the variation of the average correlation with respect to the beam size. The improvement from $K = 1$ to $K = 20$ is small (34.4 to 35.6), and the rank order for the different systems remains unchanged. Therefore, we believe that using QUESTEVAL with $K = 1$ is a reasonable choice, allowing for fast computation while preserving satisfying correlation with human judgments.

Computational Complexity We believe it is important to develop effective methods before finding ways to speed them up. Despite being slower than ROUGE, QuestEval correlates much better with human judgments while not needing actual human annotators. The current running time with a single RTX 2080 is 2.53sec per document on average on CNN/DM.

Now that QuestEval effectiveness is confirmed, we plan to focus in future work on speeding up its implementation. Distilled models seems a promising direction.[5]

Moreover, a large space for improvement is possible with implementation tricks, e.g. caching the results, since always the same questions are generated. In particular, what takes most of the time is the generation of the questions on the source document: 1) it is an autoregressive process, and 2) the source document is longer than the summary, hence contains more questions. However, those questions are required to be generated only once and for all, since the source document remains unchanged. By removing it we lower the computation to 1.82sec/doc. We plan to release an optimised version of QuestEval, including caching ability, as well as releasing all the questions on the development and test sets for XSUM & CNN/DM, which will reduce the computation cost for future evaluation.

4.5 QuestEval beyond Summarization

In this Chapter, we presented the QuestEval framework applied to evaluate Summarization. However, while beyond the scope of this thesis, QuestEval could naturally be used as it as a general NLG metric.

In (Scialom et al. 2021), we applied QuestEval to Text Simplification, obtaining state-of-the-art results.

Beyond Text modality, we could even extend the implementation to a multimodal setup, where we would compare not only texts but any kind of objects. We explored such scenarios in (Rebuffel et al. 2021) and (H. Lee et al. 2021), where we applied QuestEval to Table To Text and Image Captioning tasks respectively.

[5] https://efficientqa.github.io/assets/report.pdf

Beyond English, other languages are characterized by different morphological structures. Token-level metrics can be very sensitive to such structures: for instance, they are not suitable to evaluate agglutinative languages like Korean (D. Lee et al. 2020). Conversely, QUESTEVAL could conceptually fit any morphological structure.

Toward this multilingual QUESTEVAL, we need 1) mutlilingual corpus, and 2) mutlingual QA/QG models. To this purpose, in (Scialom et al. 2020), we released MLSUM, the first multilingual Summarization corpus. In (Riabi et al. 2020), we proposed a method to improve multilingual QA and QG models. We hope that these contributions will help to build an efficient multilingual QUESTEVAL.

4.6 Conclusion

We proposed QUESTEVAL, a new *reference-less* framework to evaluate summarization models, which unifies previous QA-based approaches and extends them with question weighting, accounting all in one for factual consistency, relevance and information selection.

Compared to existing metrics, we find that QUESTEVAL correlates dramatically better with human judgments, while at the same time not requiring any reference. This allows for more accurate comparison between systems. Moreover, any progress in question answering and generation can directly be applied within our proposed framework, leading to additional improvements. We make the code available with the hope that it will contribute to further progress in the field.

Leveraging on this public release, we note that in a very recent work, Gunasekara et al. (2021) already proposed to train summarization systems via RL, using QuestEval as a reward. According to their results, it leads to 30% reduction in terms of hallucination.

CHAPTER 5

GUIDED DECODING FOR NLG

abstract

In this Chapter, 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.

- Scialom, Thomas and Dray, Paul-Alexis and Lamprier, Sylvain and Piwowarski, Benjamin and Staiano, Jacopo. "Discriminative Adversarial Search for Abstractive Summarization". Proceedings of the 37th International Conference on Machine Learning (ICML). 2020.

5.1 Introduction

In the previous chapters, we discussed how to alleviating the exposure bias, by optimizing a *sequence level metric*. How well does the optimised metric reflect human judgement is crucial to the success of the resulting model. Our proposed metrics based on Question Answering, SumQA and its extension, QuestEval, perform favorably compared to BLEU or ROUGE. Arguably, though, they do not perfectly reflect human judgments, which limits the potential of their applications as a reward.

To tackle the 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-backpropagable from the discriminator to the generator. A solution is to use reinforcement learning, with the classifier prediction as a reward signal. However, due to reward sparsity and mode collapse (W. 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.

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. In contrast to standard sequence GAN approaches, there is no divergence in our proposed process. Additionally, our 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 presented in this Chapter 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.

5.2 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:

$$P_\gamma(y|x) = \prod_{t=1}^{|y|} P_\gamma(y_t|x, y_{1:t-1}) \tag{5.1}$$

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:

$$\frac{1}{|H|} \sum_{(x,y)\in H} \log(D_\delta(x, y)) + \frac{1}{|G|} \sum_{(x,y)\in G} \log(1 - D_\delta(x, y)) \tag{5.2}$$

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:

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

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

where x^i stands as the i-th text from the training set X and $H(x^i)$ and $G(x^i)$ respectively correspond to the associated human-written summary and the set of every generated summary for text x^i.

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).

5.2.1 Discriminative Beam Reranking

At inference time, the aim is usually to maximize the probability of the output y according to the generator (Eq. 5.1). 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 :

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

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 an hyper-parameter which enables to control exploration and complexity of the process. It ranges between 1 and 5 in the literature.

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:

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

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

Require: B, T, K_{rerank}, α
$C \leftarrow \{\text{Start-Of-Sentence}\}$
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

Figure 5.1: **Algorithm 1** DAS: a Beam Search algorithm with the pro-posed discriminator re-ranking mechanism highlighted.

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

Table 5.1: 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.

the pool of the K_{rerank} sequences with best $S_{gen}(\hat{y}, x)$ score, as detailed in Algorithm 5.1.

5.2.2 Retraining the Discriminator

Under the proposed paradigm, as mentioned in Section 5.1, the discriminator can be fine-tuned using the outputs which were found improved via the re-ranking (see Eq. 5.2.1). Hence, inspired by the GAN, we iteratively retrain the discriminator given the new predictions until convergence. We detail the retraining procedure in Figure 5.2. As illustrated, the learning signal is provided to the discriminator in a positive feedback loop until convergence, enabling to improving the generation process that combines both the Generator and the Discriminator.

5.3 Experimental Protocol

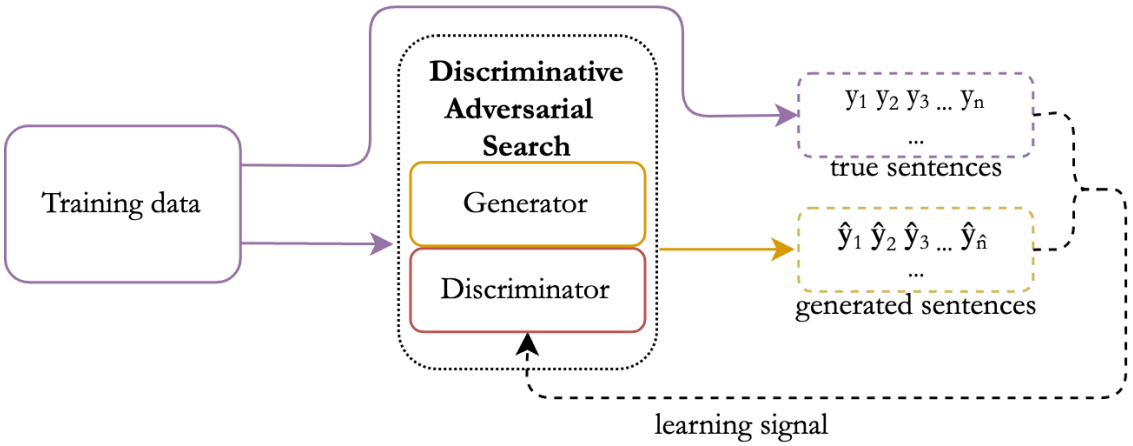


Figure 5.2: DAS self-training procedure: the generated examples are improved by the discriminator, and then fed back to the discriminator in a self-training loop.

Datasets Consistently with the previous Chapters, we use the CNN/Daily Mail (CNN/DM) dataset (Hermann et al. 2015; Nallapati et al. 2016), one of most popular datasets for summarization. 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. 2017a). 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 5.1. 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.

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

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

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

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

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.

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 5.2, 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.

5.4 Preliminary study

High discriminator accuracy is of utmost importance for DAS to improve the decoding search. In Fig. 5.3 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. 5.2). 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. When trained without access to the source article x (orange plot), the discriminator has access to little contextual

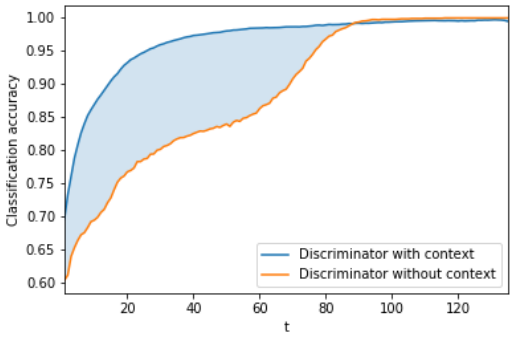


Figure 5.3: Accuracy of two discriminators: one is given access to the source context x while the other is not. The x-axis corresponds to the length of the discriminated sub-sequences.

and semantic information and its accuracy is lower than a discriminator who has access to x. In Fig. 5.3, 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 5.1), 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.

	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 5.2: Scores obtained with varying K_{rerank}

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 5.2 and 5.3 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. 5.2.1). With $K_{rerank} = 1$, no reranking is performed, and the model is equivalent to *BERT-gen*. From Table 5.2, 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.

Table 5.3 reports results for DAS models for $K_{rerank} = 10$ while varying α. α controls the impact of the discriminator predictions when selecting the next token to generate (see Eq. 5.2.1). With $\alpha = 0$, the discriminator is deactivated and only the generator probabilities S_{gen} are used (corresponding to Eq. 5.1): 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 $\alpha = 5$ 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.

5.5 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 5.4.

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

[4]As publicly released by the authors.

	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 5.3: Scores obtained with varying α

	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 5.4: 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 5.5: Results on TL;DR test set for our proposed model in transfer learning scenarios.

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.

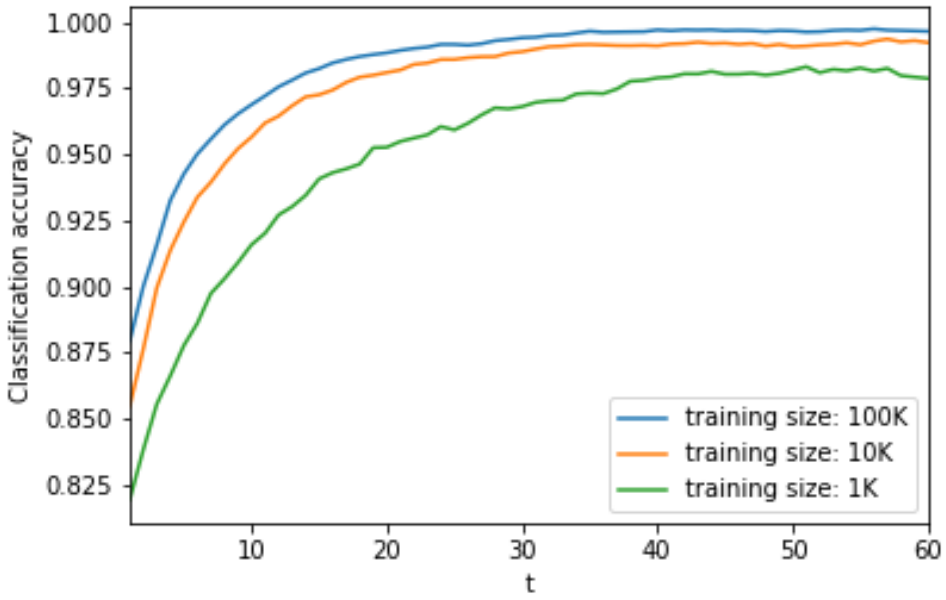


Figure 5.4: Learning curve for discriminators trained on TL;DR on 1k, 10k and 100k examples. The x-axis corresponds to the length of the discriminated sub-sequences.

Domain Adaptation Further, in Fig. 5.5, 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 5.3. 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. 5.4 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

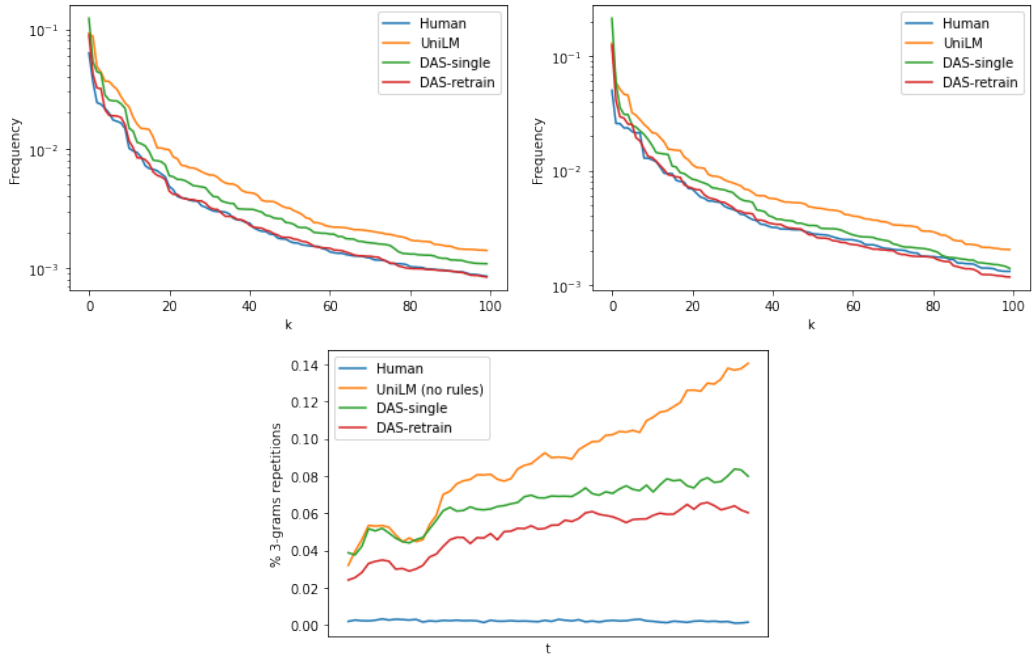


Figure 5.5: Vocabulary frequency for the $k = 100$ most frequent words generated by the models, for CNN/DM (top left) and TL;DR (top right). Distribution of 3-grams repetitions over their position t in the sequence on CNN/DM (bottom center).

the one obtained on CNN/DM (see Fig. 5.3) is due to the relatively lower quality of the *out-of-domain* generator outputs, which makes the job easier for the discriminator. The results on TL;DR[5] (Table 5.5) 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.

Discussion In Fig. 5.5 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

[5]Models participating to the public TL;DR leaderboard[6] 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.

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. 5.5 (top right) – 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.

5.6 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 report extensive ablation analyses, and show the benefits of our approach in a domain-adaptation setup. Importantly, all these improvements are obtained without any costly generator retraining. Our results highlight the effectiveness of discriminators for text generation. Still, one of the main limitation comes from the unscalability of the discriminators that can not output probabilities for all the classes in one pass. In future work, one could mitigate this limitation, a direction explored in the recent GEDI (Krause et al. 2020).

CHAPTER 6

LANGUAGE GANS

abstract

As discussed in the previous chapters, at the root of MLE limitations is the mismatch between training and inference, i.e. the so-called exposure bias. Generative Adversarial Networks (GANs) can mitigate those limitations but the discrete nature of text has hindered their application to language generation: the approaches proposed so far, based on Reinforcement Learning, have been shown to underperform MLE.

For that reason, in the previous chapter, we limited the use of a discriminator at inference time only. In this Chapter, In this Chapter, we go one step further and propose a Language GAN. Departing from previous works, we analyze the exploration step in GANs applied to text generation, and show how classical sampling results in unstable training. We propose to consider alternative exploration strategies in a GAN framework that we name $ColdGANs$, where we force the sampling to be close to the distribution modes to get smoother learning dynamics. For the first time, to the best of our knowledge, the proposed language GANs compare favorably to MLE, and obtain improvements over the state-of-the-art on three generative tasks, namely unconditional text generation, question generation, and abstractive summarization.

- Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano. "ColdGANs: Taming Language GANs with Cautious Sampling Strategies". Part of Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020.

6.1 Introduction

In the previous Chapter, we proposed DAS, a new decoding method that leverages on a discriminator at inference time, without requiring to finetune the generator. While a desirable feature in some scenarios, not retraining the generator is also an intrinsic limitation in DAS: the discriminator is limited to rerank only the sequences close to the generator mode. Arguably, one prefer, when possible, to improve directly the generator itself.

To mitigate MLE limitations, we have shown in Chapter 3 how Reinforcement Learning (RL) can be applied to text generation tasks, considering sequence level metrics as the reward. However, because metrics poorly correlate with human judgments, the potential for such a RL setup is somehow limited. To this end, Ziegler et al. (2019) proposed to directly reward systems using human judgment. Although this approach performs very well and approximates the best possible reward, it is obviously not a viable solution in practice. However, it attests that, with perfect rewards, one can achieve excellent levels of performance.

A natural alternative, not requiring human judgments, is to frame the problem under the Generative Adversarial Network (GAN) paradigm (Goodfellow et al. 2014), which has been used successfully for image generation (Brock et al. 2018). For text, modeled as a sequence of discrete symbols, a naive computation of the gradients is however intractable. Hence, Language GANs are based on gradient estimation via RL-based techniques (Yu et al. 2016).

However, the reward in this case can be extremely sparse (as discussed in Section 6.2.2), yielding to high-variance gradient estimation, which is known to be challenging for optimization (Y. Zhang et al. 2017). Most previous works have focused on this aspect, and proposed denser rewards (Li et al. 2017; Masson d'Autume et al. 2019a). Unfortunately, these attempts to apply GANs to text generation obtained limited success (Caccia et al. 2020) and have been found to underperform MLE (Semeniuta et al. 2018; Tevet et al. 2018; Masson d'Autume et al. 2019a).

Although known to be crucial (Sutton and Barto 2018), *exploration* is surprisingly understudied when RL is applied to text generation. In this work, we propose a new exploration method that aims at sampling more structured rewards and that better suits the GANs' training dynamics, allowing for the first time to successfully train Language GANs. Our main contributions can be summarized as:

1. We study the discriminators' behavior and show that their degree of specialization has important implications on the exploration to stabilize the training process. In particular, we find that reducing the exploration space is essential to successfully train discrete GANs.

2. Based on these observations, we propose $ColdGANs$, a GAN architecture using alternative sampling strategies that force the sampling to remain closer to the distribution modes.

3. Finally, we apply our proposed methods on three tasks. We report positive results compared to previous works, including GANs and MLE-based models.

6.2 Discriminators and Generators Interaction

6.2.1 Generating and discriminating as text to text tasks

Generator Text generation naturally lends itself to autoregressive modeling (Sutskever et al. 2014). The probability to generate a sequence Y composed of N tokens $y_1, ..., y_N$ is given by:

$$p_\theta(Y|X) = \prod_{t=1}^{N} p(y_t|y_1, ..., y_{t-1}, X, \theta) \tag{6.1}$$

where θ are the learnable parameters of the generator and X the input sequence.

Neural networks typically produce class probabilities by using a "softmax" output layer that converts the logit z_i, computed for each token of the vocabulary, into a probability q_i:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{6.2}$$

where T is a "temperature" hyper-parameter, set to 1 unless otherwise specified. The higher the temperature, the more uniform the probability distribution over the vocabulary, resulting in more diversity but also more mistakes (Hinton et al. 2015). In the following, we note as π_θ the distribution defined by the generator with temperature $T = 1$.

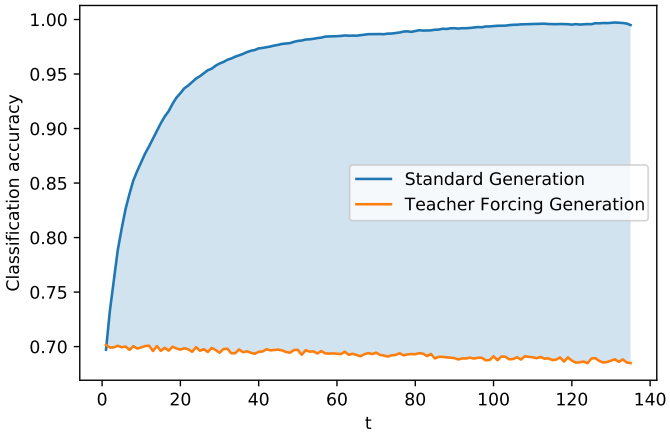
Discriminator In the following, we consider a discriminator D_ϕ learned from sets of human and generated texts for each input X as a logistic regression problem:

$$\frac{1}{|H|} \sum_{(X,Y)\in H} \log(D_\phi(X, Y)) + \frac{1}{|G|} \sum_{(X,Y)\in G} \log(1 - D_\phi(X, Y))$$

where H is a set of pairs of input X associated with a human written text Y from the data distribution, and G is a set of pairs with generated outputs Y.

Text to text tasks Casting any NLP task as a text-to-text problem, T5 (Raffel et al. 2019) demonstrated state-of-the-art results on the established GLUE benchmark (A. Wang et al. 2018) and on its more challenging successor (A. Wang et al. 2019). Accordingly, we employ the same architecture for both discrimination and generation. This allows for fairer comparisons thereafter, as both generator and discriminator have the same architecture, pre-training and capacity.

Figure 6.1: Accuracy of a discriminator model trained under two different generation modes: *Standard* (subject to the exposure bias) and *Teacher Forcing*. The x-axis corresponds to the partial length t of the sequence to discriminate.



6.2.2 Discriminator-Generator Equilibrium

Exposure Bias As mentioned above, a discriminator can easily predict the human or machine nature of a text. One reason for this lies in exposure bias. To quantify this statement, we compare the results for a discriminator when trained under the two following generation strategies: *Standard Generation*, suffering from the exposure bias; and, *Teacher Forcing Generation*, where the ground-truth tokens $y_{i<t}$ are fed to the generator, so not to expose the model to its own prediction, and only y_t is generated by a machine.

We show the results in Fig. 6.1. As expected, the two discriminators have the same score for $t = 0$. We observe that both perform well, and that the Standard Generation discriminator obtains consistently larger improvements, w.r.t. the Teacher Forcing Generation discriminator, as the length of the sequence increases. This could indicate the presence of the exposure bias, for which the errors accumulate over time. Still, the relatively high accuracy observed under Teacher Forcing Generation suggests that additional factors, beyond exposure bias, might be involved: in the following, we show that the extreme specialization of discriminators is among those.

Table 6.1: Probability that a text is human according to various discriminators. $D_{perfect}$ corresponds to a theoretical perfect discriminator with infinite capacity and training data. $D_{T=\gamma}$ corresponds to a discriminator trained on samples generated with a temperature $T = \gamma$. Past $T = 0$ and past $T = 1$ correspond to results on samples obtained with the generator weights resumed from a previous stage of the training, i.e. a checkpoint one epoch before the final state (see Section 6.3, Memory Replay). Note that sampling from the generator with $T = 1$ corresponds to the true distribution of the generator, $T = \infty$ a uniform distribution, and $T = 0$ the argmax in a greedy decoding.

	Evaluated on					
	human	$T = 0$	$T = 1$	$T = \infty$	past $T = 0$	past $T = 1$
$D_{T=0}$	.79	.17	.84	.92	.26	.85
$D_{T=1}$	.79	.76	.23	.09	.75	.31
$D_{T=\infty}$	.92	.92	.91	.08	.92	.91
$D_{T\in\{0,1,\infty\}}$	.69	.24	.24	.09	.32	.36
$D_{perfect}$	1	0	0	0	0	0

Discriminator's No Free Lunch As defined above, the temperature T of the generator is a hyper-parameter which allows to control the randomness of predictions while sampling, by scaling the logits before applying a `softmax`:

$$P_i = \frac{e^{\frac{y_i}{T}}}{\sum_{k=1}^{n} e^{\frac{y_k}{T}}} \tag{6.3}$$

This offers the opportunity to control various sampling strategies from the same generator. Low (close to 0) temperatures provide samples close to the sequence s_θ^{greedy} of a greedy procedure that takes the token with max generator probability π_θ at each step (the output of a beam search with beam size $B = 1$). With high temperatures, the distribution of sequences tends to the uniform distribution. We experiment with different temperature settings for the same generator (trained with MLE), and use the obtained samples to train and test a discriminator. This allows us to evaluate the impact of differences in sampling temperatures, between training and inference, on the discriminator performance. In other words, how a discriminator, trained with samples obtained at a specific temperature, performs when faced with samples generated under different sampling setups.

We train and evaluate discriminators on samples generated under temperatures $T = 0$, 1 or ∞, for a conditional generation task (summarization, see Section 6.4.2), which allows to consider various sequence samples even at low temperatures. We report the results in Table 6.1. As expected, in all but one case, discriminators

perform better if trained and evaluated with sequences generated under the same temperature (no mismatch). However, when the training and evaluation samples are generated with different temperatures, we observe that the discriminator fails to distinguish human from generated ones. More precisely, it considers most sentences to be human-generated (around 90%). Conversely, when trained on the different temperatures together ($T \in \{0, 1, \infty\}$), results are more balanced: robust across the various temperatures, but yielding a drop in accuracy, consistently with the well-known accuracy-robustness trade-off (Gilmer et al. 2018; Bubeck et al. 2018). This highlights that individual discriminators are specialized on specific generation pairs (machine/human). Knowing this, it is crucial to orient this specialization on useful areas.

Interestingly, when trained from samples issued from π_θ, the discriminator $D_{T=1}$ is inaccurate at identifying samples close to s_θ^{greedy} as generated ones: $D_{T=1}(s)$ equals 0.76 on average over these samples. This is particularly bad for a discriminator used as a reward signal of a RL process, since such samples lie in the useful area of the output distribution. They correspond to samples close to the modes of the distribution π_θ. Moreover, in many text generation applications, generation strategies such as beam search target these sequences as prediction outputs. A bad reward function at these locations is likely to lead to bad generation performance. Besides, the discriminator trained on samples close to the mode of π_θ (i.e., $D_{T=0}$) is bad for samples from π_θ (i.e., $T = 1$), indicating that one cannot simply use such samples to train the discriminator while considering standard sampling for generator training (as rewards would be very inaccurate).

Implications for Discrete GANs Holtzman et al. (2019) report that for $T = 1$, sampling from the tail of the distribution is expected to happen within the first three steps of decoding and with a probability superior to 99.96% within 20 steps. Such unstructured exploration causes a large variance which grows with the number of time steps, and perturbs actions too frequently (Rückstieß et al. 2008; Kober and Peters 2009). A less random exploration would thus yield to better structured sequences and lower variance, closer to the distribution learned by the discriminator, and would likely enable better training dynamics between the discriminator and the generator.

6.3 Taming Language GANs with Cautious Sampling Strategies

Based on the findings above, we seek sampling strategies that allow both the discriminator to train on useful samples, and the generator to be trained from reliable

rewards from the discriminator, within a policy gradient RL scheme where we are interested at maximizing $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[D_\phi(\tau)]$, according to generator parameters θ. The discriminator is updated at the end of each training epoch, via gradient ascent on human-machine pairs, with new artificial sequences resulting from the generator distribution. In order to introduce cautious sampling that focuses more on modes of distributions, note that it would be useless to consider the policy gradient $\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta^{T=\gamma}}[D_\phi(\tau)] = \mathbb{E}_{\tau\sim\pi_\theta^{T=\gamma}}[D_\phi(\tau)\nabla_\theta \log \pi_\theta^{T=\gamma}(\tau)]$ of a generator distribution with modified temperature $T = \gamma$, as it would, compared to $T = 1$, only imply rescaling the network outputs without altering the learning process.

Instead, we propose to employ Importance Sampling for defining our cautious sampling strategies for text GANs, based on the fact that, for any distribution $P, Q : \mathcal{X} \to [0, 1]$ such that $Q(x) > 0$ whenever $P(x) > 0$, and any function $f : \mathcal{X} \to \mathbb{R}$, we have $\mathbb{E}_{x\sim P(x)}[f(x)] = \mathbb{E}_{x\sim Q(x)}[\frac{P(x)}{Q(x)} f(x)]$. In our case, this yields the following unbiased policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\hat{\pi}_\theta}\left[\frac{\pi_\theta(\tau)}{\hat{\pi}_\theta(\tau)} D_\phi(\tau) \sum_{t=1}^{|\tau|-1} \nabla_\theta \log \pi_\theta\left(\tau_t|\tau_{1:t-1}\right)\right] \tag{6.4}$$

where $\tau_t \in V$ is the t-th token from sequence τ and $\tau_{1:t-1}$ the subsequence of its $t-1$ first tokens, π_θ the generator probability and $\hat{\pi}_\theta$ a modified sampling distribution, which enables the generation of any possible sequence of tokens given the vocabulary V.

In this work, we focus on the exploration stage; therefore, conversely to previous works, we can choose the most sober form of reward: 1 if $D_\phi(\tau)$ predicted human, and 0 otherwise. We show that a sparse reward is not a limitation if the sampling strategy is close to the modes of the distribution – provided the initial solution is a good enough bootstrap (which, according to our experiments, is the case). Note that D_ϕ is trained with samples from $\hat{\pi}_\theta$ to avoid any mismatch with the generator training samples, which would be problematic otherwise (as pointed out in Section 6.2.2).

ColdGANs exploration The temperature T plays a major role in moderating exploration. Indeed, being a scaling factor applied to the generator outputs, it directly defines the degree of diversity of the generated sequences. The default exploration is obtained by recursively sampling a sequence of tokens from the model distribution with $T = 1$. The higher T, the more random the sampled sequences, regardless of the model's policy. Conversely, lower temperatures reduce the exploration, with $T \to 0$ ultimately equivalent to the `argmax` function. Therefore, we consider a distribution $\hat{\pi}_\theta = \pi_\theta^T$ with lower (*colder*) temperatures $T \in]0, 1[$.

This allows to explore sequences composed of tokens less likely to be sampled from $\hat{\pi}_\theta$ tail. Note that for $T > 0$, $\hat{\pi}_\theta > 0$ whenever $\pi_\theta > 0$.

ColdGANs$_{nucleus}$ In addition, we consider a more sophisticated technique: *nucleus sampling* (Holtzman et al. 2019). This decoding method has been shown to produce higher quality texts than previous sampling strategies, including those temperature-based. Sampling from the nucleus of tokens containing the vast majority of the probability mass, the approach dynamically truncates the unreliable tail of the probability distribution and hence is an instance of a *cautious* generative process. However, with nucleus sampling, many sequences τ get $\hat{\pi}_\theta(\tau) = 0$ while $\pi_\theta(\tau) > 0$, invalidating the IS. To avoid this, we propose to use a mixture combining low temperatures and nucleus policies:

$$\hat{\pi}_\theta(\tau) = \epsilon \pi_\theta^{nucleus}(\tau) + (1 - \epsilon)\pi_\theta^{T=\gamma}(\tau) \tag{6.5}$$

where ϵ is a hyper-parameter, $\pi_\theta^{nucleus}$ is the probability under nucleus and $\pi_\theta^{T=\gamma}$ the probability rescaled for temperature γ, as described in the previous paragraph.

Importance Weight Clipping The importance weights can become large, causing instability. Adapting (Z. Wang et al. 2016) (see paragraph 3.2 of their paper for more details), we truncate the importance weights and add a correction term in the computation of $\nabla_\theta J(\theta)$:

$$\mathbb{E}_{\tau\sim\hat{\pi}_\theta}\left[\min(c, w(\tau))D_\phi(\tau)\nabla \log \pi_\theta(\tau)\right] + \mathbb{E}_{\tau\sim\pi_\theta}\left[\max\left(0, \frac{w(\tau) - c}{w(\tau)}\right) D_\phi(\tau)\nabla \log \pi_\theta(\tau)\right]$$

where $w(\tau) = \frac{\pi_\theta(\tau)}{\hat{\pi}_\theta(\tau)}$. In the first term of Eq. 6.3, by clipping the importance weight, the variance of the gradient estimate is bounded. The second term of the equation ensures that our estimate is unbiased, by re-sampling another sequence from the true policy π_θ. In our experiments, we set $c = 5$. Note that, contrary to off-policy RL, for which such a IS clipping was proposed (Z. Wang et al. 2016), in our case clipping is very rare: it only occurs for sequences whose probability from the generator is much higher than the one from the sampling distribution, which is designed for sampling close to the mode of π_θ. However, if this happens, this clipping ensures that the corresponding gradient does not explode.

Memory Replay In Table 6.1, we observed that the performance of the discriminators is lower when evaluated on samples generated from the previous checkpoint of the same model (i.e., evaluated on past T). We connect this to the failure mode in GANs observed by Metz et al. (2016), where the generator and the discriminator oscillate during training, rather than converging to a fixed point.

In lifelong learning literature (Masson d'Autume et al. 2019b), it has been shown that 1% of experience replay is sufficient to avoid catastrophic forgetting. Inspired by this work, we construct a memory buffer which contains samples generated in the last K training steps, and replace 1% of the discriminator training examples with samples from the buffer. This allows the discriminator to remain accurate on the samples from the previous state of the generator, hence preventing such failure loop during training.

6.4 Experiments

Due to the computational cost of T5-large (11B parameters), we used T5-small (60M parameters). For all our experiments, we used the validation sets for hyperparameter selection. In more detail, we evaluated our approach with several learning rates,[1] reporting results for a value of 2e-5. From the best performing $ColdGAN$ configuration, we perform ablations to assess the impact of Memory Replay and Importance Weight Clipping. Finally, we experimented with BART (Lewis et al. 2019) instead of T5.[2]

6.4.1 Unconditional Language Generation

Most previous works for language GANs have been evaluated on unconditional language generation benchmarks. In this task, no input is provided and the goal is to generate both meaningful and diverse texts. Consistently with (Masson d'Autume et al. 2019a), we measure these two aspects using, respectively, BLEU (Papineni et al. 2002) and self-BLEU (Zhu et al. 2018) metrics.[3] To obtain a finer comparison between models, Caccia et al. (2020) proposed to draw the curve of (negative) BLEU vs self-BLEU, by sampling with various temperatures at inference. This allows to measure the trade-off between quality and diversity. Following (Che et al. 2017; K. Lin et al. 2017; Semeniuta et al. 2018; Guo et al. 2018; Caccia et al. 2020; Masson d'Autume et al. 2019a), we used the EMNLP2017 news dataset.[4] We report $ColdGANs$ results in Figure 6.2 (left). Notice that previous works did not use self-supervised pretrained models, while we did (with T5): this explains the improvement of our MLE baseline over theirs (MLE ScratchGAN). As one cannot directly compare our performances with those reported from previous

[1] 2e-6, 8e-6, 2e-5, 8e-5, 2e-4.

[2] BART has comparable performance to T5-large, but with 20x fewer parameters.

[3] Implemented in https://github.com/deepmind/deepmind-research/tree/master/scratchgan

[4] http://www.statmt.org/wmt17/

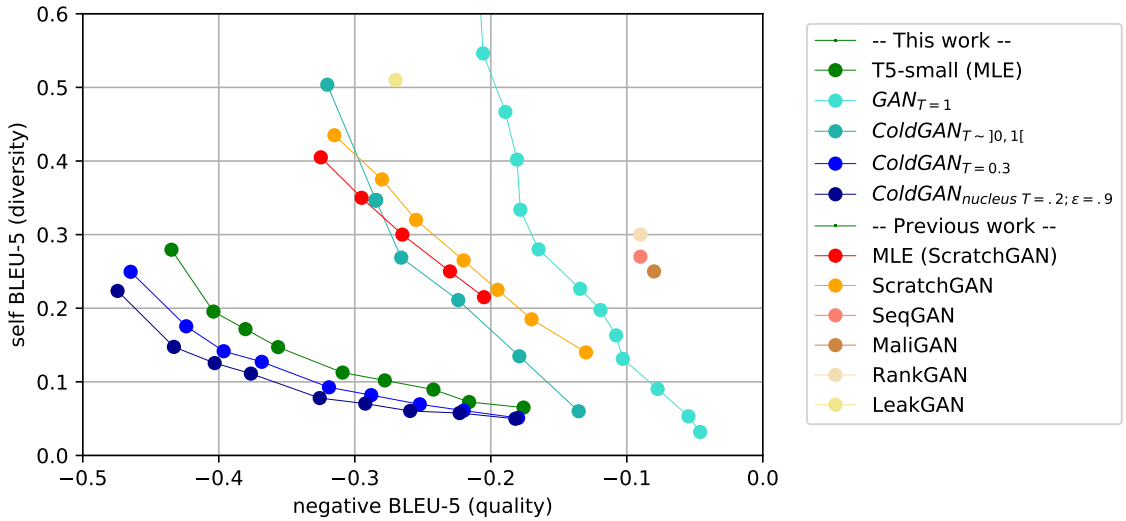


Figure 6.2: Results on the EMNLP 2017 News dataset (for all metrics, lower is better). Scores for previous works are taken from (Masson d'Autume et al. 2019a).

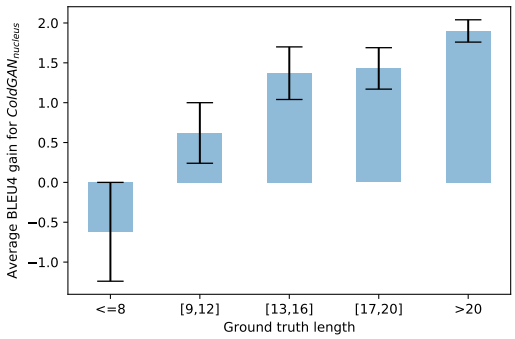


Figure 6.3: Relative BLEU-4 gains obtained with $ColdGANs$ over MLE, grouped by ground truth sequence length, on QG.

works, we study the performance variations from the corresponding MLE baseline. Consistently with previous works (Semeniuta et al. 2018; Tevet et al. 2018; Masson d'Autume et al. 2019a), we observe that the model, under the default exploration (i.e. $GAN_{T=1}$), performs strictly worse than MLE. As a baseline, we experimented $ColdGAN_{T\sim]0,1[}$, where during the training the temperature is randomly sampled between 0 and 1 for each sequence. While it performs better than $GAN_{T=1}$, it still does not compare favorably w.r.t. MLE. We report the results for a T5 model trained with the ScratchGAN protocol, and found it did not compare favorably w.r.t. T5 (MLE). Conversely, both $ColdGAN_{T=0.3}$ and $ColdGAN_{nucleus}$ obtain better results than MLE for the entire curve. To our knowledge, this is the first time that *MLE falls short* (Caccia et al. 2020; Masson d'Autume et al. 2019a) w.r.t. GAN-based approaches for this task.

6.4.2 Conditional Language Generation

We evaluate $ColdGANs$ on two popular tasks where text inputs are given for conditioning the generation, namely Question Generation and Text Summarization. These are highly competitive benchmarks, with recent state-of-the-art results achieved by MLE based on pre-trained transformers (Vaswani et al. 2017). Answer-aware Question Generation (QG) (Q. Zhou et al. 2017) is the task wherein, given a text and a target answer, the goal is to generate a relevant question. Following previous works (Dong et al. 2019; Du et al. 2017), we used the SQuAD dataset (Rajpurkar et al. 2016). Automatic Summarization aims to produce concise and fluent summaries given a longer text. Consistently with previous Chapters (see 3.4.1), we used the popular CNN/DM dataset (Nallapati et al. 2016), a corpus containing news articles and the corresponding abstractive summaries. For conditional text generation tasks, output sequences are commonly evaluated using BLEU (for e.g. Machine Translation, Question Generation) or ROUGE (for e.g. Summarization) metrics. In contrast to the unconditioned scenario, the diversity is linked to the variety of the inputs, and it is common practice to decode through beam search at inference.

Results For both tasks, we used data and evaluation metrics released by Dong et al. (2019).[5] The results shown in Table 6.2 are consistent across the two tasks: again, we observe that exploring under the default temperature yields to poor performances, while $ColdGANs$ compare favorably to MLE. The best performance is achieved with the experiment emphasizing the $ColdGAN_{nucleus}$ exploration the most, with $\epsilon = .9$ and $T = .2$. Over 10 independent training runs, we also observed

[5] https://github.com/microsoft/unilm/tree/master/unilm-v1

Table 6.2: Results on Question Generation (QG) and Abstractive Summarization (Summ.) tasks.

		QG (SQuAD)		Summ. (CNN/DM)		
	#params	BLEU-1	BLEU-4	ROUGE-1	ROUGE-L	BLEU-4
SemQG S. Zhang and Bansal 2019			18.37			
BertSumAbs Y. Liu and Lapata 2019	340M			41.72	38.76	
UniLM Dong et al. 2019	340M		22.78	43.33	40.41	
PEGASUS J. Zhang et al. 2019	568M			44.17	**41.11**	
T5-large (MLE) Raffel et al. 2019	11B			43.52	40.69	
T5-small (MLE) Raffel et al. 2019	60M	47.72	19.65	42.34	40.37	15.94
" ($GAN_{T=1}$)	60M	46.44	18.84	38.98	36.42	13.23
" ($ColdGAN_{T=.2}$)	60M	47.94	20.23	42.58	40.74	16.04
" ($ColdGAN_{nucleus\ T=1;\epsilon=.1}$)	60M	46.82	18.97	39.05	38.01	14.04
" ($ColdGAN_{nucleus\ T=1;\epsilon=.9}$)	60M	47.83	20.85	42.31	40.44	16.21
" ($ColdGAN_{nucleus\ T=.2;\epsilon=.9}$)	60M	48.50	20.55	42.54	40.61	16.86
w/o Memory Replay	60M	48.93	20.52	42.34	40.44	16.72
w/o IS Weight Clipping	60M	48.21	20.14	42.23	40.35	16.72
BART (MLE) Lewis et al. 2019	400M	53.13	22.68	44.16	40.90	17,87
" ($ColdGAN_{nucleus\ T=.2;\epsilon=.9}$)	400M	**53.73**	**23.05**	**44.46**	**41.12**	**18.17**

	Fluency	Relevance	Answerability
Human	3.66	4.31	4.22
BART (MLE)	3.80*	4.43	4.11
$ColdGAN$	**4.36****	4.45	4.01

Table 6.3: Human evaluation on QG. $ColdGAN$ corresponds to BART trained with $ColdGAN_{nucleus}$ $T = .2; \epsilon = .9$. Two-tailed t-test results are reported for each model compared to Human (*: $p < .01$, **: $p < .001$).

very stable results for this model, with a standard deviation of the average BLEU-4 lower than .09 on the test set. Finally, we applied this last $ColdGANs$ setup to BART (Lewis et al. 2019), achieving a new state-of-the-art on both QG with 23.05 BLEU-4 and summarization with 41.12 ROUGE-L.

Mitigating the Exposure Bias In Figure 6.3 we report the relative gain obtained, in terms of BLEU-4 for T5-small, for the best configuration (i.e. $ColdGAN_{nucleus}$, $\epsilon = 0.9$) w.r.t. the corresponding MLE baseline. The x-axis gives the length of considered ground truth target sequences. We observe that the longer the target sequence, the more $ColdGAN$ outperforms MLE. This might indicate that $ColdGANs$ can successfully mitigate exposure bias.

Human Evaluation As discussed in Section 6.1, automatic metrics are known to suffer from key limitations. Therefore, we additionally conducted a human evaluation on the QG task. Three professional English speakers were asked to

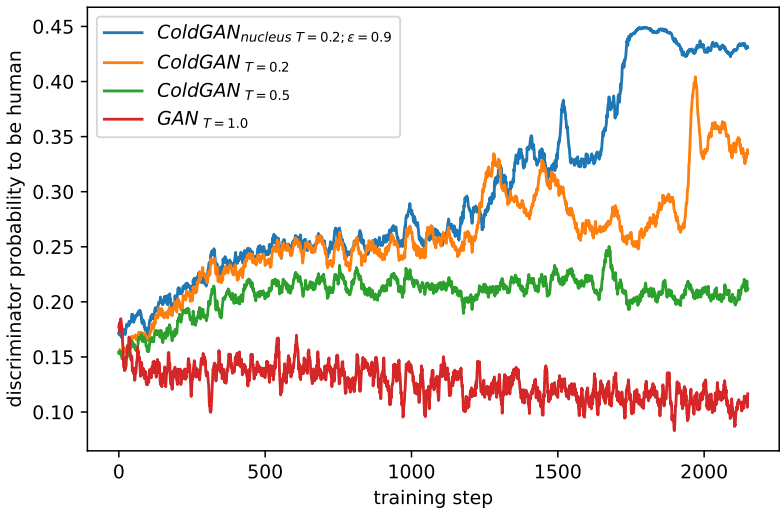


Figure 6.4: Probability that the generated text is human according to D_ϕ on CNN/DM.

judge, on a 1-to-5 Likert scale, to what extent the generated questions were: well-posed and natural (*Fluency*), relevant to their context (*Relevance*), and answerable, by looking at their context and answer (*Answerability*). The results in Table 6.3 show, surprisingly, both MLE-BART and $ColdGAN$-BART outperform the ground truth for Fluency. A similar result was reported by Yoon et al. (2020) (refer to Table 2 in their paper). A plausible explanation is that humans are more inclined to use informal language and make grammar mistakes. For instance the human question *"About how many yellow cabs operate in New York?"* sounds slightly less formal than the one, generated by $ColdGAN$, *"How many yellow taxicabs are in Manhattan ?"*. Compared to MLE, $ColdGAN$ enables to significantly improve in term of fluency, while remaining competitive on other metrics, consistently with our experiments on exposure bias.

Adversarial training curves Figure 6.4 shows the evolution (during training and for different setups) of the probability of the generated text to be human, according to the discriminator. Consistently with Table 6.2, $ColdGAN_{nucleus}$ appears to be the most adverse to the discriminator. Conversely, the regular GAN ($T = 1$) is less and less adversarial, and comparatively more perturbed.

6.5 Conclusion

We proposed $ColdGANs$, a novel approach able to *tame* the exploration in Language GANs, allowing to obtain performance improvements on both conditional

and unconditional text generation, w.r.t to MLE-based training. Our proposed IS method makes it compatible with advanced sampling methods, such as nucleus, or other future decoding methods. In the future, we plan to combine $ColdGANs$ with orthogonal approaches proposed by previous works, such as denser rewards. This opened several research directions to exploit complex sampling strategies, such as MCTS process explored in the next Chapter.

CHAPTER 7

COOPERATIVE GANS

abstract

As discussed in the previous Chapter, due to the discrete nature of words, language GANs require to be optimized from rewards provided by discriminator networks, via reinforcement learning methods. This is a much harder setting than for continuous tasks, which enjoy gradient flows from discriminators to generators, usually leading to dramatic learning instabilities. However, we claim that this can be solved by making discriminator and generator networks cooperate to produce output sequences during training. These cooperative outputs, inherently built to obtain higher discrimination scores, not only provide denser rewards for training, but also form a more compact artificial set for discriminator training, hence improving its accuracy and stability. In this Chapter, we show that our *SelfGAN* framework, built on this cooperative principle, outperforms Teacher Forcing and obtains state-of-the-art results on two challenging tasks, Summarization and Question Generation.

- Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano. "To Beam Or Not To Beam: That is a Question of Cooperation for Language GANs". Part of Advances in Neural Information Processing Systems 34 (NeurIPS 2021). 2021.

7.1 Introduction

In order to improve NLG, and to overcome Teacher Forcing limitations, a consensus has emerged as discussed previously: a sequence level objective should be introduced. A body of work has proposed to use Reinforcement Learning (RL) with NLG metrics like BLEU (Y. Wu et al. 2016) or ROUGE (Paulus et al. 2017) or our proposed SumQA and QuestEval (see Chapters 3 and 4). However, NLG metrics are not reflecting perfectly human judgement, which explains why the resulting models tend to be qualitatively worse than their MLE baselines (Caccia et al. 2020). To move toward less biased metrics, a natural alternative is to evaluate the output with a learned discriminator as explored in Chapters 5 and 6. Indeed, an ideal discriminator would not be biased w.r.t. to its training set, and could therefore be considered as a perfect metric that matches human consensus.

In light of this observation, two concurrent approaches have been explored: i) at training time, using *Generative Adversarial Networks* (e.g. ColdGAN in Chapter 6); and ii) at inference time, via cooperative decoding (e.g. DAS in Chapter 5), where a discriminator guides the search algorithm, such that *the generator and the discriminator cooperate* to select the generated tokens. These approaches pursue the same objective: producing texts more similar to what a human writes.

Both methodologies suffer from specific limitations. Cooperative decoding algorithms rely on a discriminator that re-ranks a *limited* set of candidates selected by the generator. Hence, cooperative decoding algorithms are limited by the generator ability to rank relevant tokens in a good enough position. On the other hand, language GANs are learned via reinforcement learning due to the discrete nature of text. This makes them particularly unstable to train, and usually fall short compared to Teacher Forcing (Caccia et al. 2020). In standard Language GANs, the discriminator provides a reward for the entire sequence, which can be difficult to exploit by the generator due to its sparsity (Masson d'Autume et al. 2019a).

In this Chapter, we propose *SelfGAN*, a framework to learn language GANs in a *Self*-training process where the signal from the discriminator is passed to the generator in a completely differently. We consider cooperative algorithms as a way to infuse the discriminator signal. We start from a simple observation: outputs obtained via cooperative decoding are more human-like, compared to their generator-only counterparts. Inspired by recent knowledge distillation approaches, we propose to consider *cooperative outputs* as targets in a Teacher Forcing training process: cooperative decoding stands as a teacher we attempt to imitate through the generator network. Just like a standard GAN, both the generator and the discriminator are trained at each step. While the generator improves, it becomes adversarial to the discriminator, which benefits from the cooperative generation. The discriminator, now trained on improved sequences, also contributes to improve the cooperative generation, and so forth. Note that in *SelfGAN*s the discriminator is only used to drive the cooperative generation and never to provide a reward signal like in standard Language GANs.

SelfGAN can be implemented with any cooperative decoding algorithm. Current cooperative approaches, e.g. DAS and (Deng et al. 2020) rely on "myopic" algorithms like Beam Search or Sampling that generate the tokens left-to-right. The model has to always predict the next word, and can never look back and revise past choices. In some cases, despite all the candidates being judged to likely not be human by the discriminator, the model is locked in a dead-end. This behavior is quite unnatural for humans – who often proofread their texts. We refer to this phenomenon as the *left-to-right curse*.

To address this *left-to-right curse*, we introduce *Coop-MCTS*, a new decoding algorithm based on Monte Carlo Tree Search (MCTS) (Coulom 2006; Kocsis and Szepesvári 2006). We compare *Coop-MCTS* to state-of-the-art cooperative decoding algorithms in two scenarios: i) at inference time, as the decoding algorithm; and ii) during training, as the cooperative algorithm in *SelfGAN*. In both scenarios, we show that the respective resulting outputs are more likely to look like human texts and improve all the automatic metrics.

All in all, our contributions can be summarized as follows:

1. We propose a new training framework based on cooperative decoding named **SelfGAN** , wherein the generated sequences are used as ground truth;
2. We improve cooperative decoding with a new decoding algorithm with *Coop-MCTS*, offering a solution to the left-to-right limitation of current search methods;
3. We show that combining *SelfGAN* and *Coop-MCTS* compare favorably to prior state-of-the-art results on two challenging tasks, Summarization and Question Generation.

7.2 SelfGAN

Algorithm 7.1 *SelfGAN*

Input: a generator gen, a discriminator $discr$, and a cooperative decoding method $decod_{coop}$
for n epochs **do**
  for X, S_{ref} in training set **do** ▷ *Start Training*
    $S_{coop} \leftarrow decod_{coop}(X, gen, discr)$
    gen.train(srcs=X, tgts=S_{coop}) ▷ *Standard maximum likelihood but with* S_{coop} as the target, and not S_{ref}
    discr.train(srcs=X, human_exs= S_{ref}, machine_exs=S_{coop})
  end for
end for

Inspired from Expert Iteration algorithm (Anthony et al. 2017), the idea in our *SelfGAN* approach is to transfer the sparse signal of the discriminator, classically used as rewards for a RL procedure, to the sampling mechanism of sequences that have to be favored through MLE. In that way, *SelfGAN* starts from a pre-trained generator, that we fine-tune using sequences S_{coop} provided by a cooperative decoding process $decod_{coop}$ for each condition in the training set X. This process, detailed in the next section, uses both the generator and a discriminator

network to output human-like sequences S_{coop}, for which we improve the generator likelihood via classical maximization: $\max_\theta \sum_{(x,s)\in(X,S_{coop})} log\pi_\theta(s|x)$, where $\pi_\theta(s|x) = \prod_{t=1}^{|s|} \pi_\theta(s_t|s_{0:t-1}, x)$ stands for the generator probability of sequence s given the conditioning input x, with π_θ as previously implemented as a neural architecture with a softmax output function.

At each iteration of the training procedure, the discriminator network is optimized as a binary classifier on i) the human references and ii) the machine generated via the cooperative sequences:

$$\frac{1}{|H|} \sum_{(x,s_{ref})\in H} \log(D(x, s_{ref})) + \frac{1}{|G|} \sum_{(x,s_{coop})\in G} \log(1 - D(x, s_{coop}))$$

where x is the source input, H is the set of pairs associating x with a human written text s_{ref} from the data distribution, and G is a set of pairs with generated outputs s_{coop}. As in previous Chapter, $D(x, s)$ stands for the probability, provided by the discriminator network, that sequence s is a human reference for condition x. In order to effectively guide the cooperative process at each step, the discriminator needs to be sequential: consistently with DAS, we use a left-to-right mask during training, allowing discriminator predictions for unfinished sequences.

Please note that, by construction of the cooperative decoding process, we have with high probability at each iteration $D(x, s_{coop}) >= D(x, s_{gen})$ for any condition $x \in X$, with s_{coop} a cooperative decoded sequence for x and s_{gen} a sequence directly sampled from the generator according to $\pi_\theta(s|x)$. Based on this observation, and provided that the discriminator is sufficiently trained at each step, the generator is trained such that the probability of predicting human-like sequences is maximized. This process i) allows us to consider a sequence level metric, and ii) offers more stability compared to Reinforcement Learning, as we observe in our experiments (see section 7.5). Note also that, contrary to RL approaches which have to find a good balance between discriminator and generator capacities, our approach does not suffer from Vanishing Gradient (Arjovsky and Bottou 2017), since discrimination is only used for decoding, in a cooperative process for generator training. We depict the *SelfGAN* in Algorithm 7.1.

7.3 MCTS for Decoding

7.3.1 Coop-MCTS: Cooperative Decoding beyond the *Left-To-Right Curse*

It can happen that all sequence candidates are judged by the discriminator to be machine-like rather than human-like. In such case, the cooperative decoding is stuck in a *dead end*; such limitation is unsatisfactory. Neither $\text{DAS}_{\text{local}}$ or $\text{DAS}_{\text{global}}$ have the ability to revise their previous decisions.

To cope with those limitations of myopic decoding strategies, we propose to consider an adaptation of MCTS for NLG. Just like in the context of games (Silver et al. 2017), we consider a policy network π, the generator, that outputs a probability over all the possible actions (tokens) at each step of the sequence. The discriminator D corresponds to the value network. In MCTS, the trajectories are explored to build a tree following three steps:

1. **Selection** starting from the root, children nodes tokens ω are selected among the vocabulary $\mathcal{V}$ recursively w.r.t. the `PUCT` algorithm (Rosin 2011; Silver et al. 2017):
$$\omega = \arg\max_{\omega \in \mathcal{V}} \left(Q(s,\omega) + c_{puct} \pi_\tau(\omega \mid s) \frac{\sqrt{\sum_b N(s,b)}}{1 + N(s,\omega)} \right) \tag{7.1}$$
where Q is the value of taking action ω in state s: in NLG, this corresponds to selecting a token among the vocabulary at step i given the source context and the sub-sequence $\omega_0, ..., \omega_{i-1}$. c_{puct} is a constant, τ the temperature that scales the Softmax, and $N(s,\omega)$ the number of times the token ω has been chosen in state s. We stop the loop when a node s_o has not been expanded yet, i.e. the discriminator D has not calculated its value.
2. **Extension** Given the selected node, we calculate the distribution probability from the generator $\pi(\omega \mid s_o)$. We apply nucleus sampling (Holtzman et al. 2019) to filter out the less likely tokens and reduce the number of actions. The remaining tokens constitute the children nodes, associated to their corresponding probability. At the same time, we calculate the value of the current state $D(s_o)$ that allows to compute the backup step.
3. **Backup** we update Q for all the nodes that led to s_o such that $Q \leftarrow \max(Q, D(s_o))$. Note that we choose to use the max instead of the average for the following reason: the value network, i.e. a discriminator, becomes more accurate as the candidate sequence grows (see Figure 5.3 , hence if a long sequence is judged human by the discriminator, any of its sub-sequences should be considered human-like as well. In contrast, a long sequence can be machine-like despite

starting in a very human-like manner: the beginning sub-sequence should keep its human-like score.

These three steps are computed for a restricted number of simulations. Then, the next token corresponds to the root child with the most visit counts. The process continues step by step to generate the next token, until reaching either the special token End Of Sentence, or the maximum length.

7.4 Experiments

7.4.1 Datasets

To measure the effectiveness of *SelfGAN*, we experiment on two standard conditional NLG tasks: Question Generation (QG) and Summarization, consistently with the previous Chapters (see 3.4.1):

- **Question Generation:** we used the SQuAD dataset (Rajpurkar et al. 2016), consisting of 100K triplets of Wikipedia paragraphs, factual questions, and their answers.
- **Summarization:** we used the CNN/Daily Mail dataset (CNNDM) (Nallapati et al. 2016), consisting of 300K news articles, paired with their corresponding summaries. The summaries are formed of multiple sentences, making the amount of tokens to generate much larger than for Question Generation.

7.4.2 Models Reported

MLE the first baseline we consider is a standard model trained via teacher forcing. As for all our experiments, we initialised the seq2seq with T5 (Raffel et al. 2019).

ColdGAN we consider as a second baseline the current state-of-the art for language GANs, ColdGAN, described in Chapter 6. The authors proposed to lower the temperature when sampling the sequences during training, with the objective of stabilizing the training process.

SelfGAN can be based on any cooperative decoding algorithm. To train *SelfGAN*, we therefore experiment the three different cooperative algorithms described in Section 7.3 ($\text{DAS}_{\text{Local}}$, $\text{DAS}_{\text{Global}}$, and *Coop-MCTS*) and report the results for the corresponding *SelfGAN*: $\textit{SelfGAN}_{\textit{DAS-Local}}$, $\textit{SelfGAN}_{\textit{DAS-Global}}$, and $\textit{SelfGAN}_{\textit{Coop-MCTS}}$.

Decoding Method at inference time For each model, any decoding method can be applied at inference time, independently from the training scheme. Therefore,

for all the models described above, we report the results given each decoding method previously described (Section 7.3): Beam Search, DAS_{Local}, DAS_{Global}, and *Coop-MCTS*.

To the best of our knowledge, GANs and Cooperative decoding have never been directly compared before this work. A fortiori, this is the first time that a GAN model is tested with a Cooperative decoding method at inference. We investigate possible distillation effects in Section 7.5.

7.4.3 Metrics

To compare the different models, we report two type of metrics: *n-gram based* and *discriminator*.

N-gram based We report the standard BLEU (Papineni et al. 2002) and ROUGE (C.-Y. Lin 2004). Both measure an overlap of n-grams between the reference and the evaluated text. They differ in that BLEU is precision oriented while ROUGE is rather recall oriented.

Discriminators Both BLEU and ROUGE suffer from the aforementioned limitations. We therefore propose to consider discriminators for model evaluation. Intuitively, they measure how model outputs are similar to what a human would have written. We consider two different discriminators:

- **Base** is a discriminator trained on the MLE baseline outputs generated via beam search. It allows to measure the corresponding improvement from the MLE baseline. Note that it corresponds to the initial discriminator in all the GANs experiments, and the discriminator used in the cooperative search for the MLE baseline.

- **Base+** Since the *Base* discriminator plays a role in all our experiments (except MLE+Beam Search), it is possible that a model that makes use of this *Base* obtains better *Base* results, despite bringing new biases and *de*-generation behaviors. For this reason, we also report *Base+*, a discriminator fine-tuned on all the different model outputs together. *Base+* is never used by any model at training or inference time. It is thus more robust toward an undesirable adversarial generation mode, while still being comparable for the different experiments. We argue that a higher *Base+* score indicates a real improvement beyond potential bias.

Generator	Question Generation					Summarization				
Decoder	**B4**	**R1**	**RL**	**Base**	**Base+**	**B4**	**R1**	**RL**	**Base**	**Base+**
MLE										
BeamSearch	19,7	45,2	41,1	15%	15%	15,9	42,3	40,4	9%	8%
DAS_{local}	19,9	45,2	41,1	28%	19%	16,6	43,8	40,9	17%	11%
DAS_{global}	20,0	45,2	41,2	20%	17%	16,2	44,1	41,9	12%	9%
Coop-MCTS	19.8	45,3	41,5	33%	21%	16,3	42,5	40,6	20%	12%
ColdGAN										
BeamSearch	19.9	45,2	41,4	26%	17.9%	16,3	42,8	40,7	15%	10%
DAS_{local}	19,8	45,3	41,1	31%	20%	15,9	42,5	42,0	19%	11%
DAS_{global}	20,2	45,6	41,5	26%	18%	16,6	44,6	41,2	16%	10%
Coop-MCTS	19,9	45,4	41,2	39%	22%	15,9	44,2	41,2	23%	12%
SelfGAN$_{DAS_{loc}}$										
BeamSearch	20,2	45,4	41,6	27%	21%	16,9	44,2	42,5	16%	11%
DAS_{local}	20,5	45,5	41,7	30%	23%	16,9	44,4	41,9	18%	13%
DAS_{global}	20,1	45,4	41,7	33%	20%	16,6	44,0	42,3	19%	11%
Coop-MCTS	20,4	45,5	41,8	39%	23%	16,4	43,8	42,8	23%	13%
SelfGAN$_{DAS_{glob}}$										
BeamSearch	20,4	45,5	41,7	24%	19%	16,9	43,0	41,5	14%	11%
DAS_{local}	19,9	45,4	41,3	32%	22%	15,9	42,7	40,6	18%	12%
DAS_{global}	20,7	45,6	41,9	29%	20%	17,0	43,7	42,6	17%	11%
Coop-MCTS	20,0	45,3	41,4	40%	24%	16,1	43,4	42,3	23%	13%
SelfGAN$_{Coop\text{-}MCTS}$										
BeamSearch	20,5	46,6	42,6	34%	21%	17,0	42,8	41,5	20%	13%
DAS_{local}	20,6	46,7	41,7	42%	24%	16,6	43,7	42,8	25%	13%
DAS_{global}	20,5	46,6	41,7	39%	21%	16,5	42,8	40,9	23%	12%
Coop-MCTS	**21,1**	**48,9**	**44,7**	40%	**26%**	**17,5**	**43,5**	**42,3**	23%	**15%**

Table 7.1: Results of our experiments on QG (left) and Summarization (right). For each generator, we report the results with the four different decoders. The reported metrics correspond to BLEU4 (B4), ROUGE-1 (R1), ROUGE-L (RL) and the discriminators Base and Base+ as described in Section 7.4.3. For Base and Base+ the scores correspond to the probability of being human, so higher is better for all the metrics. For *SelfGAN*$_{MCTS}$, we experimented with 5 different seeds and the standard deviation is always inferior to 0.1 for BLEU4 and ROUGE, and inferior to 0.5% for Base and Base+.

7.5 Results and discussion

7.5.1 Conditional Text Generation

In Table 7.1, we report the results for all the previously trained generators, with the different decoding algorithms presented in Section 7.3. By 'model', in the following, we refer to the couple composed by a trained generator and a decoding algorithm.

We report BLEU4, ROUGE-1, ROUGE-L along with scores for the discriminators *Base* and *Base+*, computed as the percentage of outputs considered as human by a given discriminator model. *Base* was only trained on *MLE+Beam Search* outputs. As expected, by further training on the outputs generated by all the different models, *Base+* has a higher accuracy, which consistently results in lower scores compared to *Base*.

We start by focusing on the MLE results to compare the different decoding mechanisms. We observe that all the cooperative searches outperform Beam Search. Regarding Base and Base+ metrics, $\text{DAS}_{\text{Local}}$ compares favorably to $\text{DAS}_{\text{Global}}$. We hypothesize that invoking the discriminator to rank at each step can have more impact than using it only once on fully decoded sequences. Finally, our proposed *Coop-MCTS* obtains the best results by a large margin.

Regarding the different GANs, we first compare them given the default decoding mechanism, i.e. Beam Search. The three versions of *SelfGAN* compare favorably to MLE and ColdGAN on both n-gram based metrics and discriminators metrics. Among *SelfGAN*s, $\textit{SelfGAN}_{\text{Coop-MCTS}}$ obtains the best results: given a Beam Search decoding, it obtains the best BLEU, ROUGE-1 and ROUGE-L on the two tasks (respectively 17.2; 44.3; 40.6 on QG and 12.3; 38.6; 36.7 on Summarization). The performance in term of Base and Base+ for $\textit{SelfGAN}_{\text{Coop-MCTS}}$ is even more important in comparison to the other models (34.1%; 21.9% on QG and 20.2%; 12.7% on Summarization).

Both GAN at training time and Cooperative decoding at inference time pursue the same objective: to obtain better outputs that look like human texts. Would a generator trained via GAN, coupled with a Cooperative Decoding mechanism for inference result into a cumulative improvement from the two methods? First, on both ColdGAN and three *SelfGAN*s, we can observe that adding a Cooperative Decoding method allows to gain significant improvement on Base and Base+. In particular, it is interesting to note that for *SelfGAN* an additional pattern seems to emerge: using the same cooperative decoding algorithm both during training and inference seems to provide additional gains. The best performance is achieved with the generator $\textit{SelfGAN}_{\text{Coop-MCTS}}$ paired with the decoding *Coop-MCTS*. Com-

Model	T=0.5	T=1	T=2
MLE+Sample	0.42;0.29	0.31;0.11	0.18;0.07
ColdGAN+Sample	0.47;0.21	0.33;0.08	0.22;0.06
MLE+Coop-MCTS	0.45;0.22	0.34;0.10	0.21;0.06
SelfGANCoop-MCTS+Coop-MCTS	0.48;0.20	0.37;0.09	0.24;0.05

Table 7.2: Results on Unconditional Text Generation for samples realized at three different temperatures, in terms of BLEU Vs Self-BLEU (higher better;lower better).

pared to MLE via Beam Search, it obtains a final improvement superior to 1 point in term of ROUGE and BLEU. The relative improvement for Base+ is significant: from 15.2% to 26.2% on QG and from 8.6% to 15.3% on Summarization. This corresponds to almost twice more outputs that sound human according to the Discriminator metric.

7.5.2 Unconditional Text Generation

We follow the ColdGAN setup: we compared our proposed approaches on the EMNLP2017 News dataset. The evaluation takes into account both the quality and the diversity. Consistently with previous works (e.g. ColdGAN, ScratchGAN, LeakGAN), we use the following metrics:i) BLEU-5 for measuring the quality (higher better); ii) Self-BLEU-5 for measuring the diversity (lower better).

To obtain a finer comparison between models, Caccia et al. (2020) proposed to draw the curve of BLEU vs self-BLEU, by sampling with various temperatures at inference.

We denote as the standard method to generate text in this setup, as used in all the previous works we are comparing to. It is a simple left to right decoding where, at each step, a token is sampled among the Softmax probabilities scaled by the temperature.

In our Coop-MCTS, the probability of a token is given by its visit counts during the simulations. In Conditional generation, we select at each step the token with the maximum number of counts. In Unconditional generation, we sample from the tokens counts distribution.

Overall, the results are consistent with the experiments on Conditional Generation: the MLE generator decoded with our proposed MCTS (3rd row) obtains:

1. significantly better slightly lower results than ColdGAN decoded with Sample(2nd row)
2. results than when the same MLE decoded with Sample(1st row);

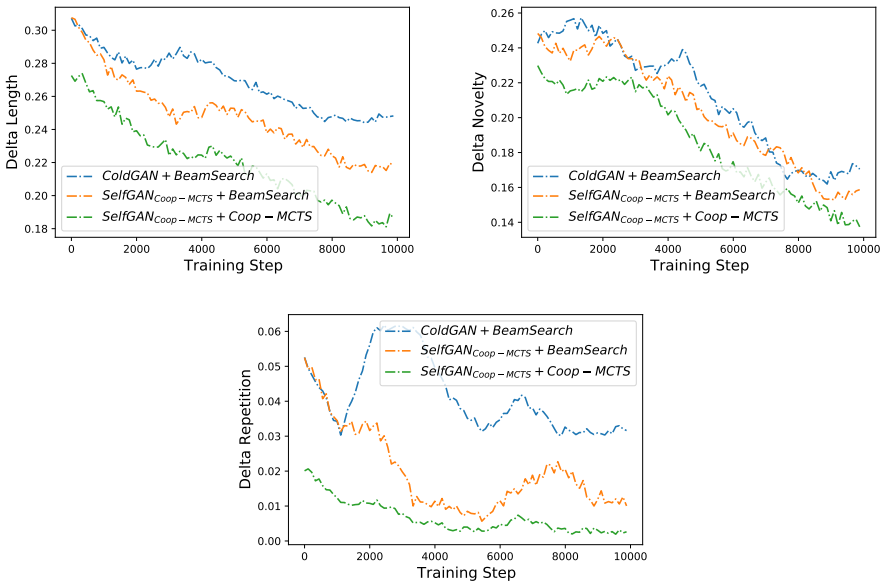


Figure 7.1: Average difference for Summarization between human references and model outputs for the Length (Left), the Novelty (Middle), and the 3-grams repetitions (Right) during training. The closer to 0 the less differences w.r.t. gold-references.

3. SelfGAN decoded with MCTS (4th row) obtains the best results.

7.5.3 Discussion

Human-like features during training In NLG, various rules are often integrated into the Beam Search to improve the quality of the outputs, for instance a length penalty (See et al. 2017) or an interdiction for 3-grams repetitions (Paulus et al. 2017; Dong et al. 2019). Such a need to hard code these rules indicates a discrepancy between the human output characteristics and what the model has learned. In particular, in Chapter 5, we reported the difference between DAS and the human reference for: i) **Length**: the average number of tokens per output; ii) **Novelty**: percentage of tokens in the output that were not present in the source text; iii) **N-gram repetition**: percentage of N-grams that occur more than once in the output.

To measure how *SelfGAN* learns these features by itself, we report in Figure 7.1 the evolution of these statistics during training: we observe that *SelfGAN* constantly reach statistics more similar to human references than ColdGAN.

Human Evaluation We conduct a human evaluation to measure the models performances beyond automatic metrics. We limit the evaluation to three genera-

Generator	Decoder	**Consistency**	**Coherence**	**Fluency**	**Relevance**
MLE	BeamSearch	3.9	3.1	4.1	3.2
MLE	Coop-MCTS	3.4**	3.5**	3.8	3.6**
ColdGan	BeamSearch	3.8	3.3	4.2	3.5
ColdGan	Coop-MCTS	3.4**	3.6**	4.0	3.7**
$\text{SelfGAN}_{\text{Coop-MCTS}}$	BeamSearch	**4.0**	3.5**	**4.3***	3.9**
$\text{SelfGAN}_{\text{Coop-MCTS}}$	Coop-MCTS	3.9	**3.9****	4.0	**4.2****

Table 7.3: Human Evaluation on Summarization. Two tailed t-test results are reported for each model compared to MLE+BeamSearch (*: $p < .01$, **: $p < .001$).

tors (MLE, ColdGAN, and $\textit{SelfGAN}_{\text{Coop-MCTS)}}$ and two decoding methods (Beam Search and *Coop-MCTS*), for a total of 6 different models. Three professional English speakers rated 300 sampled summaries and followed the same protocol from Fabbri et al. (2020). Four dimensions are evaluated on a Likert scale from 1 to 5 (the higher the better):

1. **Consistency:** the proportion of facts in the summary correct w.r.t. the source text;
2. **Coherence:** how well-structured and well-organized is the summary;
3. **Fluency:** how fluent the summary is to read;
4. **Relevance:** the ratio between important and excess information in the summary.

From Table 7.3 we observe significantly better results for $\textit{SelfGAN}_{\text{Coop-MCTS}}$ w.r.t. both MLE and ColdGAN. While *Coop-MCTS* decoding appears overall beneficial in terms of Coherence and Relevance, but scores lower on Consistency and Fluency, its combination with $\textit{SelfGAN}_{\text{Coop-MCTS}}$ allows to obtain significant improvements on the former two dimensions while still maintaining comparable scores on the latter.

Analysis To further understand the benefits of selfGAN, we propose to analyze the evolution of the generator and discriminator networks through the learning process. In figure 7.2 (left), we first plot the average magnitude (L2 norm) of the discriminator gradients w.r.t. its parameters. We observe that $ColdGAN$ induces important instabilities for its discriminator over time, with a highly fluctuating gradient magnitude. Conversely, thanks to its cooperative decoding process, *SelfGAN* produces sequences that form a more compact set for discriminator training, a variance of gradient magnitude twice lower than $ColdGAN$, for a comparable magnitude in average. This discriminator stability is a first explanation for the improvements of the proposed approach.

In a second plot, given on the right of Figure 7.2, we report the collinearity of generator gradients for the generated samples from the model with those for

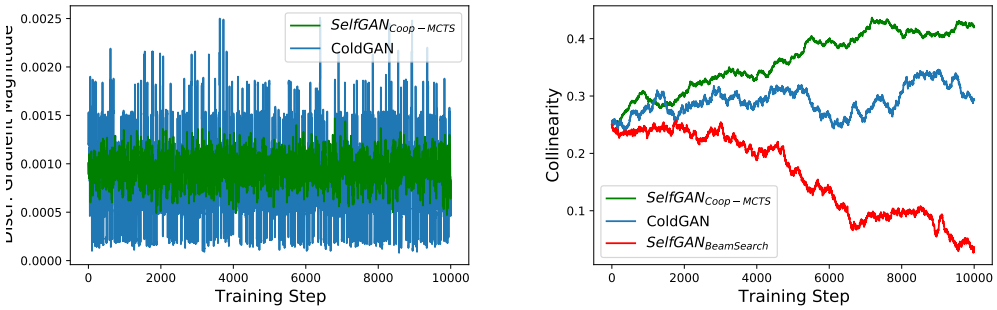


Figure 7.2: Left: Moving Average of the magnitude of the *discriminators* gradients during training. Right: collinearity of the *generators* gradients between the sampled texts and their corresponding human reference for *SelfGAN*$_{\text{Coop-MCTS}}$, ColdGAN and *SelfGAN*$_{\text{BeamSearch}}$. Both on Summarization.

the corresponding human references. Higher values indicate sampling strategies that induce a useful gradient flow for the generator. For ablation purposes, we first report values for a "*SelfGAN*$_{BeamSearch}$" approach, where we used a standard Beam Search to generate the training examples: note that it has no discriminator, hence it is not a GAN anymore. We can observe its divergence, as opposed to *SelfGAN*$_{\text{Coop-MCTS}}$, which emphasizes the importance of the cooperative decoding for producing the example used to train the model. For *SelfGAN*$_{\text{Coop-MCTS}}$ and *ColdGAN*, the gradients become more co-linear with human references through time, indicating a convergence of the process towards the human distribution. We observe that *SelfGAN*$_{\text{Coop-MCTS}}$ produces more useful sequences for achieving this convergence.

Coop-MCTS as an alternative to the dead-end search When analysing the behavior for the *Coop-MCTS* decoding, we observed in different examples that it provides an effective mean to revise generations that eventually ended up to be unlikely. To illustrate this, we report in Table 7.4 the different MCTS steps for an ambiguous example: the conditioned answer, *Super Bowl*, occurs at different places of the the input. Therefore, the model has to decide which specific mention of *Super Bowl* to focus on: at step 17, it considers its current generation as a dead end and decides to start on new node (*How*). The final output is a question that arguably sounds better than the initial one.

7.6 Conclusion

In this Chapter we propose *SelfGAN*, a new framework to train Generative Adversarial Networks based on a cooperative decoding search. To overcome the left-to-right curse that limits standard search algorithms, we propose *Coop-MCTS*. We

Conditioned Answer: *Super Bowl*
Context:

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24â€“10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals

Step 01: *What*
⋮
Step 16: *What was the name of the game that would have been known as "Super Bowl*
Step 17: *How*
⋮
Step 46: *How is called the American football game that determines the NFL champion?*

Table 7.4: Progressive results obtained by our *Coop-MCTS* decoding method on Question Generation during a simulation. Until the 16th step, the generation is left-to-right. Then, the cooperation mechanism kicks in, allowing the model to safely abort this beam, by restarting a new question with *How*. We report the cross-attention weights on the input context for step 16 (red) and 17 (blue).

conducted extensive experiments on two challenging tasks: Summarization and Question Generation, obtaining state-of-the-art performance for *SelfGAN* both in terms of automatic metrics and within a human evaluation. As the stability of the discriminator looks to be crucial for language GANs, we plan for future works to still focus on increasing it through the definition of dynamic regularization mechanisms. Finally, we will explore how reference-less metrics, e.g. QuestEval (presented in Chapter 4), can be combined to help the exploration during the decoding.

Chapter 8

CONCLUSION AND PERSPECTIVES

Fluent and reliable Natural Language Generation can have significant societal impacts. On the one hand, we envision several applications beneficial for business, research, or education: from automatic summarization of news articles, scientific papers or books, to efficient information access; from automatic and personalized student evaluation tests through question generation, to responsive conversational interfaces. On the other hand, malicious actors can use the same technology to build tools detrimental to society, e.g. for creation and propagation of misleading (fake) news as discussed in (Radford et al. 2019), impersonation, and deceit. Nonetheless, keeping this research open and under public scrutiny is arguably one of the best ways to defend against such actors (Zellers et al. 2019).

8.1 Beyond A Unique Reference

In this thesis, we have explored different axes to improve NLG, both on the generative and evaluation sides. One common trait among all these efforts is the need to work beyond a unique gold-reference.

In NLG, given an input, there are arguably many different possible outputs. For both humans and machines, given the same input, it is highly likely that several different – yet, all correct – sequences can be produced. In particular, the probability to write the same exact sequence as the only gold-reference available is very low. Should this behavior be penalized? Obviously not. And yet, this is what happens under the standard NLG practices:

1) at training time with Teacher Forcing, any generated token that is different from the target increases the loss. The model can therefore be exposed to contradictory information, which might limit its effectiveness. Note that this issue does not apply to a discriminator, as only two output categories (machine or human) are possible.

2) at inference time models are evaluated with automatic metrics fed with only one gold-reference, since most large-scale datasets provide a single human example in their test set.

To address this limitation, and in contrast to other metrics, our QG/QA based metric $QuestEval$ compares the evaluated text directly to the source, not the reference.

Regarding training, in the last chapter we proposed SelfGAN, which offers a theoretical solution to this multi-reference limitation. Lets denote S_{human} the universe of possible correct outputs, where $s_{ref} \in S_{human}$. Then, given a perfect discriminator (optimal to distinguish real data distribution from a different distribution), and an infinite computational capacity, we have $s_{coop} \in S_{human}$. Indeed, given an infinite computational capacity, all the possible sequences can be explored. A perfect discriminator classifies a sequence s as human only if $s \in S_{human}$. It results that $s_{coop} \in S_{human}$: the sequence generated via a cooperative mechanism is guaranteed to be indistinguishable from any human output, just like the reference.

In addition, since the generator probability is also taken into account in a cooperative decoding, we have $s_{coop} = argmax(P_\pi(S_{human}))$. We note that this is guaranteed only if all possible sequences are explored via an infinite computation. If we stop searching when one sequence is accepted by the decoder, it is pseudo-guaranteed since a Beam Search is only an approximation of the argmax.

s_{coop} is the sequence among all the human sequences that maximise the likelihood according to the generator π. Therefore, if the generator outputs a human-level sequence (i.e. $s \in S_{human}$), is will actually correspond to s_{coop}. It results that considering s_{coop} as the gold-reference in Teacher Forcing, the generator will not be subject to an undesirable loss.

In conclusion, SelfGAN can be interpreted as a generalisation of Teacher Forcing that takes into account the multiple possible references and trains the model on the reference the highest to its likelihood.

8.2 Future Directions

8.2.1 Metrics

In this thesis we have paved the way for a new kind of metrics based on Question Answering and Generation. We have shown how this method can be broadly adapted to various NLG tasks, systematically comparing favorably over the others metrics regarding human judgement.

Nonetheless, Question-based metrics have interesting properties by design. In particular, they provide a direct and fine-grained explanation w.r.t. the output hallucinations. This could open new ways toward automatic proofreading mechanisms, or being exploited within human-machine interactions. We note that such new directions are only made possible because our proposed metrics are reference-less, hence usable even when no ground-truth reference is available.

At the root of the reference-less property, lies the intuition that any objects can be compared by asking and answering questions. In this regard, QuestEval can be seen as a general similarity function, offering a way to compare the generated text with any object, in particular the source, and not necessarily a text. In contrast, metrics like ROUGE, BLEU or BERTScore operate at a token level. Hence, they require a certain alignment between the two compared texts. This prevents from using the source input, and limits to compare the evaluated text to a ground truth reference. Moreover, the NLG community is progressively moving toward more complex and longer documents. As several realizations of a correct generation are possible, the number of potentially correct gold-references exponentially rises w.r.t. the length of the input sequence. In this context, QuestEval becomes particularly interesting, as it loosens the dependence on a specific realization of the source text.

8.2.2 RL

Toward an unbiased reward, we have explored learned discriminators. First at inference time, as a way to select the words likely to sounds more human-like during the decoding process. Then at training time in a GAN. For the later, we have proposed different methods to stabilise the training process trough complex sampling strategies, such as MCTS. Overall, our empirical experiments obtain state-of-the-art results, under both automatic metrics and human evaluations.

Moving forward, investigating theoretical properties for our proposed approaches would be an important step. In particular, it would be interesting to introduce theoretical guarantees regarding discrete GANs, based on cooperative sampling strategies.

In addition, future works could study how our cooperative mechanisms might apply in the context of approaches based on density ratio estimators, such as the ones recently proposed in (**lu2019cot**; **song2020improving**). Hybrid approaches, based on ratio estimators between current densities and expected ones also constitute a promising research perspective.

BIBLIOGRAPHY

Anthony, Thomas, Zheng Tian, and David Barber (2017). "Thinking fast and slow with deep learning and tree search". In: *Advances in Neural Information Processing Systems* 30 (cit. on p. 81).

Arjovsky, Martin and Léon Bottou (2017). "Towards principled methods for training generative adversarial networks". In: *arXiv preprint arXiv:1701.04862* (cit. on p. 82).

Arumae, Kristjan and Fei Liu (June 2019). "Guiding Extractive Summarization with Question-Answering Rewards". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2566–2577. URL: https://www.aclweb.org/anthology/N19-1264 (cit. on p. 21).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (cit. on pp. 13, 14, 23).

Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer (2015). "Scheduled sampling for sequence prediction with recurrent neural networks". In: *Advances in Neural Information Processing Systems*, pp. 1171–1179 (cit. on pp. 2, 17).

Bengio, Yoshua, Patrice Simard, and Paolo Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2, pp. 157–166 (cit. on p. 11).

Bhandari, Manik, Pranav Gour, Atabak Ashfaq, and Pengfei Liu (Dec. 2020). "Metrics also Disagree in the Low Scoring Range: Revisiting Summarization Evaluation Metrics". In: *Proceedings of COLING 2020, the 30th International Conference on Computational Linguistics: Technical Papers*. The COLING 2020 Organizing Committee (cit. on p. 2).

Brock, Andrew, Jeff Donahue, and Karen Simonyan (2018). "Large scale gan training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096* (cit. on p. 66).

Bubeck, Sébastien, Eric Price, and Ilya Razenshteyn (2018). "Adversarial examples from computational constraints". In: *arXiv preprint arXiv:1805.10204* (cit. on p. 70).

Caccia, Massimo, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin (2018). "Language gans falling short". In: *arXiv preprint arXiv:1811.02549* (cit. on p. 52).

Caccia, Massimo, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin (2020). "Language GANs Falling Short". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=BJgza6VtPB (cit. on pp. 19, 66, 73, 75, 79, 80, 88).

Che, Tong, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio (2017). "Maximum-likelihood augmented discrete generative adversarial networks". In: *arXiv preprint arXiv:1702.07983* (cit. on pp. 19, 73).

Chen, Ping, Fei Wu, Tong Wang, and Wei Ding (2018). "A Semantic QA-Based Approach for Text Summarization Evaluation". In: *Thirty-Second AAAI Conference on Artificial Intelligence* (cit. on p. 22).

Chen, Yen-Chun and Mohit Bansal (July 2018). "Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting". In: pp. 675–686. URL: https://www.aclweb.org/anthology/P18-1063 (cit. on p. 31).

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (cit. on p. 12).

Chopra, Sumit, Michael Auli, and Alexander M Rush (2016). "Abstractive sentence summarization with attentive recurrent neural networks". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 93–98 (cit. on p. 12).

Clark, Kevin, Minh-Thang Luong, Quoc V Le, and Christopher D Manning (2019). "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators". In: *International Conference on Learning Representations* (cit. on pp. 19, 52).

Coulom, Rémi (2006). "Efficient selectivity and backup operators in Monte-Carlo tree search". In: *International conference on computers and games*. Springer, pp. 72–83 (cit. on p. 81).

Deng, Yuntian, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato (2020). "Residual energy-based models for text generation". In: *arXiv preprint arXiv:2004.11714* (cit. on p. 80).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "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 (cit. on p. 19).

Dong, Li, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon (2019). "Unified language model pre-training for natural language understanding and generation". In: *Advances in*

Neural Information Processing Systems, pp. 13042–13054 (cit. on pp. 9, 54, 56, 57, 75, 76, 89).

Du, Xinya, Junru Shao, and Claire Cardie (2017). "Learning to Ask: Neural Question Generation for Reading Comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1342–1352 (cit. on p. 75).

Durmus, Esin, He He, and Mona Diab (2020). "FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization". In: *arXiv preprint arXiv:2005.03754* (cit. on p. 38).

Eyal, Matan, Tal Baumel, and Michael Elhadad (June 2019). "Question Answering as an Automatic Evaluation Metric for News Article Summarization". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics (cit. on pp. 22, 24).

Fabbri, Alexander R, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev (2020). "SummEval: Re-evaluating Summarization Evaluation". In: *arXiv preprint arXiv:2007.12626* (cit. on pp. 17, 35, 40, 41, 90).

Fan, Angela, Mike Lewis, and Yann Dauphin (2018). "Hierarchical neural story generation". In: *arXiv preprint arXiv:1805.04833* (cit. on p. 8).

Gehrmann, Sebastian, Yuntian Deng, and Alexander Rush (2018). "Bottom-Up Abstractive Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109 (cit. on pp. 9, 19, 23, 28, 31, 56, 60, 61).

Gilmer, Justin, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow (2018). "Adversarial Spheres". In: (cit. on p. 70).

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680 (cit. on pp. 3, 51, 66).

Gunasekara, Chulaka, Guy Feigenblat, Benjamin Sznajder, Ranit Aharonov, and Sachindra Joshi (Nov. 2021). "Using Question Answering Rewards to Improve Abstractive Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 518–526. URL: https://aclanthology.org/2021.findings-emnlp.47 (cit. on p. 49).

Guo, Jiaxian, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang (2018). "Long text generation via adversarial training with leaked information". In: *Thirty-Second AAAI Conference on Artificial Intelligence* (cit. on p. 73).

Hermann, Karl Moritz, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom (2015). "Teaching machines to read and comprehend". In: *Advances in neural information processing systems*, pp. 1693–1701 (cit. on pp. 24, 28, 56).

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (cit. on p. 67).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780 (cit. on p. 11).

Hokamp, Chris and Qun Liu (2017). "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 (cit. on p. 9).

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi (2019). "The curious case of neural text degeneration". In: *arXiv preprint arXiv:1904.09751* (cit. on pp. 1, 8, 37, 70, 72, 83).

Kober, Jens and Jan R Peters (2009). "Policy search for motor primitives in robotics". In: *Advances in neural information processing systems*, pp. 849–856 (cit. on p. 70).

Kocmi, Tom, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes (2021). "To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation". In: *arXiv preprint arXiv:2107.10821* (cit. on p. 15).

Kocsis, Levente and Csaba Szepesvári (2006). "Bandit based monte-carlo planning". In: *European conference on machine learning*. Springer, pp. 282–293 (cit. on p. 81).

Konda, Vijay and John Tsitsiklis (1999). "Actor-critic algorithms". In: *Advances in neural information processing systems* 12 (cit. on p. 18).

Krause, Ben, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani (2020). "Gedi: Generative discriminator guided sequence generation". In: *arXiv preprint arXiv:2009.06367* (cit. on p. 63).

Kryscinski, Wojciech, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher (Nov. 2019a). "Neural Text Summarization: A Critical Evaluation". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 540–551. URL: https://www.aclweb.org/anthology/D19-1051 (cit. on p. 35).

Kryscinski, Wojciech, Bryan McCann, Caiming Xiong, and Richard Socher (2019b). "Evaluating the factual consistency of abstractive text summarization". In: *arXiv preprint arXiv:1910.12840* (cit. on p. 19).

Kryściński, Wojciech, Romain Paulus, Caiming Xiong, and Richard Socher (Oct. 2018). "Improving Abstraction in Text Summarization". In: pp. 1808–1817. URL: https://www.aclweb.org/anthology/D18-1207 (cit. on pp. 19, 60).

Lamb, Alex M, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio (2016). "Professor forcing: A new algorithm for training recurrent networks". In: *Advances In Neural Information Processing Systems*, pp. 4601–4609 (cit. on pp. 2, 17).

Lavie, Alon and Abhaya Agarwal (2007). "METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments". In: *Proceedings of the second workshop on statistical machine translation*, pp. 228–231 (cit. on p. 16).

Lee, Dongyub, Myeongcheol Shin, Taesun Whang, Seungwoo Cho, Byeongil Ko, Daniel Lee, Eunggyun Kim, and Jaechoon Jo (Dec. 2020). "Reference and Document Aware Semantic Evaluation Methods for Korean Language Summarization". In: *Proceedings of COLING 2020, the 30th International Conference on Computational Linguistics: Technical Papers*. The COLING 2020 Organizing Committee (cit. on p. 49).

Lee, Hwanhee, Thomas Scialom, Seunghyun Yoon, Franck Dernoncourt, and Kyomin Jung (2021). "QACE: Asking questions to evaluate an image caption". In: *arXiv preprint arXiv:2108.12560* (cit. on p. 48).

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer (2019). "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension". In: *arXiv preprint arXiv:1910.13461* (cit. on pp. 73, 76).

Li, Jiwei, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky (2017). "Adversarial Learning for Neural Dialogue Generation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2157–2169 (cit. on pp. 19, 66).

Lin, C-Y (2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Proc. of Workshop on Text Summarization Branches Out, Post Conference Workshop of ACL 2004*, pp. 74–81 (cit. on pp. 2, 16, 85).

Lin, Kevin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun (2017). "Adversarial ranking for language generation". In: *Advances in Neural Information Processing Systems*, pp. 3155–3165 (cit. on p. 73).

Liu, Yang and Mirella Lapata (2019). "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. 3721–3731 (cit. on p. 76).

Louis, Annie and Ani Nenkova (2013). "Automatically assessing machine summary content without a gold standard". In: *Computational Linguistics* 39.2, pp. 267–300 (cit. on p. 2).

Masson d'Autume, Cyprien de, Shakir Mohamed, Mihaela Rosca, and Jack Rae (2019a). "Training language gans from scratch". In: *Advances in Neural Information Processing Systems*, pp. 4302–4313 (cit. on pp. 19, 66, 73–75, 80).

Masson d'Autume, Cyprien de, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama (2019b). "Episodic Memory in Lifelong Language Learning". In: *Advances in Neural Information Processing Systems*, pp. 13122–13131 (cit. on p. 73).

McKeown, Kathleen R (1980). "Generating Relevant Explanations: Natural Language Responses to Questions about Database Structure." In: *AAAI*, pp. 306–309 (cit. on p. 1).

Metz, Luke, Ben Poole, David Pfau, and Jascha Sohl-Dickstein (2016). "Unrolled generative adversarial networks". In: *arXiv preprint arXiv:1611.02163* (cit. on p. 72).

Mihalcea, Rada and Paul Tarau (2004). "Textrank: Bringing order into text". In: *Proceedings of the 2004 conference on empirical methods in natural language processing* (cit. on pp. 6, 26).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (cit. on p. 6).

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur (2010). "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. Makuhari, pp. 1045–1048 (cit. on p. 10).

Mikolov, Tomáš, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur (2011). "Extensions of recurrent neural network language model". In: *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 5528–5531 (cit. on p. 10).

Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. (2016). "Abstractive text summarization using sequence-to-sequence rnns and beyond". In: *arXiv preprint arXiv:1602.06023*, pp. 280–290 (cit. on pp. 28, 40, 56, 75, 84).

Narayan, Shashi, Shay B Cohen, and Mirella Lapata (2018a). "Don't Give Me the Details, Just the Summary!" In: *Topic-aware Convolutional Neural Networks for Extreme Summarization. In* (cit. on p. 41).

Narayan, Shashi, Shay B. Cohen, and Mirella Lapata (June 2018b). "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)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1747–1759. URL: https://www.aclweb.org/anthology/N18-1158 (cit. on p. 23).

Nenkova, Ani (2011). *Automatic Summarization*. Foundations and Trends in Information Retrieval. Now (cit. on pp. 22, 25).

Ng, Jun-Ping and Viktoria Abrecht (2015). "Better Summarization Evaluation with Word Embeddings for ROUGE". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1925–1930 (cit. on p. 22).

Noble, William S (2006). "What is a support vector machine?" In: *Nature biotechnology* 24.12, pp. 1565–1567 (cit. on p. 6).

Novikova, Jekaterina, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser (Sept. 2017a). "Why We Need New Evaluation Metrics for NLG". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2241–2252. URL: https://www.aclweb.org/anthology/D17-1238 (cit. on p. 16).

Novikova, Jekaterina, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser (2017b). "Why we need new evaluation metrics for NLG". In: *arXiv preprint arXiv:1707.06875* (cit. on p. 2).

Ott, Myle, Michael Auli, David Grangier, and Marc'Aurelio Ranzato (2018). "Analyzing uncertainty in neural machine translation". In: *arXiv preprint arXiv:1803.00047* (cit. on p. 9).

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd (1999). *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab (cit. on pp. 6, 26).

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pp. 311–318 (cit. on pp. 15, 73, 85).

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks". In: *International conference on machine learning*. PMLR, pp. 1310–1318 (cit. on p. 11).

Pasunuru, Ramakanth and Mohit Bansal (June 2018). "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)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 646–653. URL: https://www.aclweb.org/anthology/N18-2102 (cit. on pp. 21, 24, 31).

Paulus, Romain, Caiming Xiong, and Richard Socher (2017). "A deep reinforced model for abstractive summarization". In: *arXiv preprint arXiv:1705.04304* (cit. on pp. 2, 9, 17, 18, 21, 23, 25, 27, 29–33, 79, 89).

Peyrard, Maxime (2019). "Studying summarization evaluation metrics in the appropriate scoring range". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5093–5100 (cit. on p. 2).

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9 (cit. on pp. 8, 16, 93).

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu (2019). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *arXiv preprint arXiv:1910.10683* (cit. on pp. 37, 67, 76, 84).

Rajpurkar, Pranav, Robin Jia, and Percy Liang (2018). "Know what you don't know: Unanswerable questions for SQuAD". In: *arXiv preprint arXiv:1806.03822* (cit. on p. 41).

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (Nov. 2016). "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250*, pp. 2383–2392. URL: https://www.aclweb.org/anthology/D16-1264 (cit. on pp. 6, 24, 37, 38, 75, 84).

Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba (2015a). "Sequence level training with recurrent neural networks". In: *arXiv preprint arXiv:1511.06732* (cit. on pp. 1, 2, 18, 23).

Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba (2015b). "Sequence level training with recurrent neural networks". In: *arXiv preprint arXiv:1511.06732* (cit. on p. 17).

Rebuffel, Clément, Thomas Scialom, Laure Soulier, Benjamin Piwowarski, Sylvain Lamprier, Jacopo Staiano, Geoffrey Scoutheeten, and Patrick Gallinari (2021). "Data-QuestEval: A Referenceless Metric for Data to Text Semantic Evaluation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics (cit. on p. 48).

Reiter, Ehud and Robert Dale (1997). "Building applied natural language generation systems". In: *Natural Language Engineering* 3.1, pp. 57–87 (cit. on p. 1).

Riabi, Arij, Thomas Scialom, Rachel Keraron, Beno1.t Sagot, Djamé Seddah, and Jacopo Staiano (2020). "Synthetic Data Augmentation for Zero-Shot Cross-Lingual Question Answering". In: *arXiv preprint arXiv:2010.12643* (cit. on p. 49).

Rosin, Christopher D (2011). "Multi-armed bandits with episode context". In: *Annals of Mathematics and Artificial Intelligence* 61.3, pp. 203–230 (cit. on p. 83).

Rückstieß, Thomas, Martin Felder, and Jürgen Schmidhuber (2008). "State-dependent exploration for policy gradient methods". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 234–249 (cit. on p. 70).

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *nature* 323.6088, pp. 533–536 (cit. on p. 10).

Rush, Alexander M, Sumit Chopra, and Jason Weston (2015). "A neural attention model for abstractive sentence summarization". In: *arXiv preprint arXiv:1509.00685* (cit. on p. 14).

Sai, Ananya B, Akash Kumar Mohankumar, and Mitesh M Khapra (2020). "A survey of evaluation metrics used for NLG systems". In: *arXiv preprint arXiv:2008.12009* (cit. on p. 15).

Scialom, Thomas, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano (2020). "MLSUM: The Multilingual Summarization Corpus". In: *arXiv preprint arXiv:2004.14900* (cit. on p. 49).

Scialom, Thomas, Louis Martin, Jacopo Staiano, Éric Villemonte de la Clergerie, and Beno1.t Sagot (2021). "Rethinking Automatic Evaluation in Sentence Simplification". In: *arXiv preprint arXiv:2104.07560* (cit. on p. 48).

See, Abigail, Peter J Liu, and Christopher D Manning (2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *ACL (1)*, pp. 1073–1083 (cit. on pp. 23, 28, 31, 56, 60, 89).

Semeniuta, Stanislau, Aliaksei Severyn, and Sylvain Gelly (2018). "On accurate evaluation of gans for language generation". In: *arXiv preprint arXiv:1806.04936* (cit. on pp. 19, 66, 73, 75).

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. (2017). "Mastering the game of go without human knowledge". In: *nature* 550.7676, pp. 354–359 (cit. on p. 83).

Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhudinov (2015). "Unsupervised learning of video representations using lstms". In: *International conference on machine learning*. PMLR, pp. 843–852 (cit. on p. 10).

Sulem, Elior, Omri Abend, and Ari Rappoport (Oct. 2018). "BLEU is Not Suitable for the Evaluation of Text Simplification". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 738–744. URL: https://www.aclweb.org/anthology/D18-1081 (cit. on p. 16).

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*, pp. 3104–3112 (cit. on pp. 1, 7, 10, 12, 23, 67).

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press (cit. on p. 66).

Tevet, Guy, Gavriel Habib, Vered Shwartz, and Jonathan Berant (2018). "Evaluating text gans as language models". In: *arXiv preprint arXiv:1810.12686* (cit. on pp. 66, 75).

Trischler, Adam, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman (2016). "Newsqa: A machine comprehension dataset". In: *arXiv preprint arXiv:1611.09830* (cit. on p. 41).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008 (cit. on pp. 14, 57, 75).

Venkatraman, Arun, Martial Hebert, and J Andrew Bagnell (2015). "Improving multi-step prediction of learned time series models". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence* (cit. on p. 17).

Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly (2015). "Pointer networks". In: *Advances in Neural Information Processing Systems*, pp. 2692–2700 (cit. on pp. 19, 23).

Völske, Michael, Martin Potthast, Shahbaz Syed, and Benno Stein (Sept. 2017a). "TL;DR: Mining Reddit to Learn Automatic Summarization". In: *Proceedings of the Workshop on New Frontiers in Summarization*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 59–63. URL: https://www.aclweb.org/anthology/W17-4508 (cit. on p. 56).

Völske, Michael, Martin Potthast, Shahbaz Syed, and Benno Stein (2017b). "Tl;dr: Mining reddit to learn automatic summarization". In: *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63 (cit. on p. 28).

Wang, Alex, Kyunghyun Cho, and Mike Lewis (2020). "Asking and answering questions to evaluate the factual consistency of summaries". In: *arXiv preprint arXiv:2004.04228* (cit. on pp. 35–39, 41, 47).

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman (2019). "Superglue: A stickier benchmark for general-purpose language understanding systems". In: *arXiv preprint arXiv:1905.00537*, pp. 3261–3275 (cit. on p. 67).

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (Nov. 2018). "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, pp. 353–355. URL: https://www.aclweb.org/anthology/W18-5446 (cit. on p. 67).

Wang, Ziyu, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas (2016). "Sample Efficient Actor-Critic with Experience Replay". In: (cit. on p. 72).

Weizenbaum, Joseph (1966). "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1, pp. 36–45 (cit. on p. 1).

Williams, Ronald J (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3-4, pp. 229–256 (cit. on p. 23).

Williams, Ronald J and David Zipser (1989). "A learning algorithm for continually running fully recurrent neural networks". In: *Neural computation* 1.2, pp. 270–280 (cit. on pp. 1, 7, 17).

Williams, Ronald J and David Zipser (1995). "Gradient-based learning algorithms for recurrent". In: *Backpropagation: Theory, architectures, and applications* 433, p. 17 (cit. on p. 10).

Winograd, Terry (1980). "What does it mean to understand language?" In: *Cognitive science* 4.3, pp. 209–241 (cit. on p. 1).

Wu, Harris, Dragomir R Radev, and Weiguo Fan (2002). "Towards answer-focused summarization". In: *Proceedings of the 1st International Conference on Information Technology and Applications* (cit. on p. 24).

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144* (cit. on pp. 9, 79).

Xingjian, SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo (2015). "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems*, pp. 802–810 (cit. on p. 10).

Yoon, Wonjin, Yoon Sun Yeo, Minbyul Jeong, Bong-Jun Yi, and Jaewoo Kang (2020). "Learning by Semantic Similarity Makes Abstractive Summarization Better". In: *arXiv preprint arXiv:2002.07767* (cit. on p. 77).

You, Quanzeng, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo (2016). "Image captioning with semantic attention". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4651–4659 (cit. on p. 14).

Yu, Lantao, Weinan Zhang, Jun Wang, and Yong Yu SeqGAN (2016). "Sequence Generative Adversarial Nets with Policy Gradient. arXiv e-prints, page". In: *arXiv preprint arXiv:1609.05473* (cit. on pp. 19, 66).

Zellers, Rowan, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi (2019). "Defending against neural fake news". In: *Advances in Neural Information Processing Systems*, pp. 9051–9062 (cit. on pp. 19, 93).

Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter J Liu (2019). "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization". In: *arXiv preprint arXiv:1912.08777* (cit. on p. 76).

Zhang, Shiyue and Mohit Bansal (2019). "Addressing Semantic Drift in Question Generation for Semi-Supervised Question Answering". In: *arXiv preprint arXiv:1909.06356* (cit. on p. 76).

Zhang, Yizhe, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin (2017). "Adversarial feature matching for text generation". In:

Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, pp. 4006–4015 (cit. on p. 66).

Zhou, Qingyu, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou (2017). "Neural question generation from text: A preliminary study". In: *National CCF Conference on Natural Language Processing and Chinese Computing*. Springer, pp. 662–671 (cit. on pp. 37, 75).

Zhou, Wangchunshu, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou (2020). "Self-Adversarial Learning with Comparative Discrimination for Text Generation". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=B1l8L6EtDS (cit. on pp. 19, 52).

Zhu, Yaoming, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu (2018). "Texygen: A benchmarking platform for text generation models". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1097–1100 (cit. on p. 73).

Ziegler, Daniel M, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving (2019). "Fine-tuning language models from human preferences". In: *arXiv preprint arXiv:1909.08593* (cit. on pp. 2, 66).