Which Discriminator for Cooperative Text Generation?

Antoine Chaffin, Thomas Scialom, Sylvain Lamprier, Jacopo Staiano, Benjamin Piwowarski, Ewa Kijak, Vincent Claveau

Which Discriminator for Cooperative Text Generation?

Antoine Chaffin
IRISA, IMATAG
Rennes, France
antoine.chaffin@irisa.fr
antoine.chaffin@imatag.com

Thomas Scialom
ISIR - Sorbonne Université, reciTAL
Paris, France
thomas@recital.ai

Sylvain Lamprier
ISIR - Sorbonne Université
Paris, France
sylvain.lamprier@isir.upmc.fr

Jacopo Staiano
reciTAL
Paris, France
jacopo@recital.ai

Benjamin Piwowarski
CNRS, ISIR - Sorbonne Université
Paris, France
benjamin.piwowarski@lip6.fr

Ewa Kijak
Université Rennes, IRISA
Rennes, France
ewa.kijak@irisa.fr

Vincent Claveau
CNRS, IRISA
Rennes, France
vincent.claveau@irisa.fr

ABSTRACT

Language models generate texts by successively predicting probability distributions for next tokens given past ones. A growing field of interest tries to leverage external information in the decoding process so that the generated texts have desired properties, such as being more natural, non toxic, faithful, or having a specific writing style. A solution is to use a classifier at each generation step, resulting in a cooperative environment where the classifier guides the decoding of the language model distribution towards relevant texts for the task at hand. In this paper, we examine three families of (transformer-based) discriminators for this specific task of cooperative decoding: bidirectional, left-to-right and generative ones. We evaluate the pros and cons of these different types of discriminators for cooperative generation, exploring respective accuracy on classification tasks along with their impact on the resulting sample quality and computational performances. We also provide the code of a batched implementation of the powerful cooperative decoding strategy used for our experiments, the Monte Carlo Tree Search, working with each discriminator for Natural Language Generation.

CCS CONCEPTS

• **General and reference** → **Empirical studies**; • **Computing methodologies** → **Natural language generation**; *Supervised learning by classification*; • **Information systems** → *Query suggestion.*

KEYWORDS

natural language generation, cooperative, discriminator, monte carlo tree search, attention, empirical, performance

1 INTRODUCTION

Transformer [34] architectures, coupled with an increase in computing capabilities, allows current Language Models (LM) to generate very plausible texts. Given an initial sequence of tokens (the prompt), the LM computes a probability distribution for the next token. A token is then sampled from this distribution and added to the initial sequence to generate the following token auto-regressively. Choosing the next token given the distribution (decoding) is commonly done using greedy search, beam search [8] or top-k/p sampling [11, 16]; they select the next token only based on the likelihood (according to the LM) of the resulting sequence, which offers only limited control over the text finally generated.

Yet, large LMs trained with non curated data are known to produce toxic and inappropriate content [2, 13]. This is particularly problematic for Information Retrieval tasks that imply text generation, such as question-answering from the Web [22, 25], query-focused multi-documents summarization [28], query expansion [5], query suggestion [24], or chatbots for interactive search [27], which leverage contents from various – and sometimes untrusted – information sources.

Classifiers can be trained to identify a specific property of a text and thus provide useful information to guide the LM towards the desired property. For instance, following Generative Adversarial Networks [14], many studies train binary discriminators to distinguish real from generated contents, to approximate distributions of observed documents [37]. Other studies train classifiers on semantic properties such as polarity to learn the generation process towards positive or negative texts [3, 7, 19]. In the context of Information Retrieval, this might also be used for instance to increase relevance of synthetic answers w.r.t. to the user's query. For all these purposes, there is an increasing interest for discriminator-generator cooperative decoding, where discriminators are used to guide generation [1, 12, 17, 31].

Currently, top performing discriminators are transformers using bidirectional attention [9], but this does not fit the iterative nature of the generation process. Indeed, it requires to recompute every hidden state of the whole sequence for any additional token, preventing the use of cached hidden states and resulting in a quadratic cost w.r.t. the sequence length at each timestep. On the other hand, unidirectional transformers, which employ left-to-right masks to only depend on past tokens for text encoding/decoding [29], induce hidden states that can be reused for subsequent steps, hence involving linear computing complexity. However, these two types of discriminators only score one sequence at a time, given as input of the model. This limits the number of possible tokens to be considered at each decoding step, to avoid a computationally prohibitive cost. Solving this issue, recently introduced Generative Discriminators (GeDi) [19] give scores for all tokens from the vocabulary at once, hence dramatically reducing the cost of width exploration. In this paper, we explore the pros and cons of these three types of discriminators (bidirectional, unidirectional, generative) when used in cooperative language decoding.

In parallel, approaches relying on Monte Carlo Tree Search algorithm (MCTS) [6] have been used for cooperative generation with more sophisticated exploration strategies than beam search. This non-myopic discriminator-guided decoding lead to state-of-the-art results in different applications [3, 20, 21, 32]. We therefore use this promising cooperative decoding approach for our experiments and provide an implementation of the MCTS that allows to generate texts in batch for each type of discriminator[1] based on the HuggingFace's transformers library [35].

Before exposing our experimental study, we further define the task of cooperative decoding and justify our study in the next section.

2 BACKGROUND AND MOTIVATIONS

2.1 Cooperative Decoding with MCTS

In cooperative text generation, information from the discriminator is combined to the generator distribution to skew the generation towards the desired property defined by one class of the discriminator [1, 4, 17, 31, 38]. For instance, inspired from Value-guided beam search [15, 30], but using class discriminators rather than value networks, Discriminative Adversarial Search (DAS) [31] proposed to rerank beam-generated sequences according to their discrimination scores. Among these approaches, MCTS-based ones [3, 20, 21, 32] allowed to obtain state-of-the-art results in various NLG tasks, by overcoming the limitations of myopic left-to-right decoding (and difficult value-network learning [21]).

MCTS [33] is an algorithm that iteratively builds a (generation) tree at each decision step, to take short-term decisions that might be promising in the long run. Each iteration is composed of three steps. First, during **selection**, a search toward an unexplored node is driven by a compromise between exploiting good partially generated sequences and exploring promising ones. This trade-off is controlled by the parameter $c_{puct} \in \mathbb{R}$ (higher values mean more exploration). Then, **expansion** is performed by creating children of the selected node. Finally, the corresponding sequence is scored by the discriminator and the score of every parent up to the tree root are updated accordingly during a **backpropagation** phase. In MCTS, this back-propagated score is usually computed from the selected node by **rolling out** until a terminal node and by evaluating the resulting full sequence. As done in other approaches for cooperative decoding, we replace these costly roll-outs in our experiments by scores provided by discriminators trained on unfinished text sequences. In this work, we experiment on which kinds of discriminators are the best cooperative partners for generating with MCTS.

[1] https://github.com/NohTow/PPL-MCTS/tree/main/teammates

2.2 Choosing the Right Teammate

By default, attention layers as defined in [34] are bidirectional: every token can attend to tokens at every position. When it comes to discrimination, models based on such bidirectional attention are commonly used since "intuitively, it is reasonable to believe that a deep bidirectional model is strictly more powerful than [...] a left-to-right model" [9]. However, while it brings some capacity to the model, it also makes it non auto-regressive: when a token is added at the end of a sequence, every hidden states need to be re-computed.

One way to train a transformer based LM for text generation is to use unidirectional attention masks [29]. In this unidirectional setting, any extra token added at the end of a sequence does not change the already calculated hidden states, since previous tokens do not attend to it. Thus, starting from an already classified sequence $x_{1:t-1}$, classifying $x_{1:t}$ only requires to compute t attention scores, rather than the whole set of t^2 scores per self-attention layer, as it would be required in the bidirectional setting. In common discriminative tasks, this does not matter since only entire sequences are discriminated. Hence, none of the hidden states needs to be reused for another next sample. However, for a use in auto-regressive cooperative decoding, where input sequences are often the continuation of already discriminated ones unidirectional attention allows to reuse contextual encoding of previous tokens, hence greatly speeding up the process.

However, even with unidirectional discriminators, evaluating every possible continuation of a given sequence is intractable since, for a vocabulary of size $|\mathcal{V}|$, it requires $|\mathcal{V}|$ forward passes at each decoding step. $|\mathcal{V}|$ being in the order of ten thousand, discriminating every possible continuation of decoding sequences is too costly. Thus, cooperative approaches have to circumvent this issue by limiting the number of sequences actually evaluated by the discriminator. For example, DAS pre-filters potential continuations on the nucleus of the LM distribution [16]. This choice necessarily biases the resulting generated distribution.

Recently, [19] introduced Generative Discriminators (GeDi) that exploit Class-Conditionnal Language Models (CC-LMs) [18] to discriminate every token at once. CC-LMs condition distributions of sequence x on a desired class of interest c: $p(x \mid c) = \prod_t p(x_t \mid x_{1:t-1}, c)$. Assuming a uniform prior distribution of classes $c \in C$, Bayes' rule enables to use this for discrimination: $p(c \mid x_{1:T}) \propto p(x_{1:T} \mid c)$. Thus, it only requires $|C|$ forward passes to get the discrimination scores of all possible sequence continuations. $|C|$ being usually much lower than $|\mathcal{V}|$, this makes the consideration of every token tractable for sequential discriminative decoding. To improve discriminatory capacity of such models, training of CC-LMs used in GeDi leverages a discriminative loss $\mathcal{L}_d$ in addition to the traditional language modeling loss $\mathcal{L}_g$. This discriminative loss corresponds to a cross-entropy loss using the model as a discriminator and a hyper-parameter λ is used to define the balance between the two objectives: $\mathcal{L}_{total} = \lambda \mathcal{L}_g + (1 - \lambda)\mathcal{L}_d$.

These three types of discriminators offer different capacity / complexity trade-off, which are studied in this paper for cooperative decoding with MCTS. More precisely, three questions are explored: 1) How these models differ in pure discrimination accuracy? 2) To which extend are these differences noticeable in generated texts? 3)

How do these methods compare in terms of computation complexity for cooperative decoding?

3 EMPIRICAL STUDY

According to previous studies, unidirectional models should yield worse accuracy than bidirectional ones [9, 29] and better than discriminative generators [26, 36]. To thoroughly assess the pros and cons of these models using state-of-the-art transformer architectures, it is crucial that the only difference is the studied property (uni- vs bi- directionality, and discriminative vs generative). Thus, we propose to use the same backbone for all settings to prevent any external confounding factors, with a single fully connected output layer on top of the contextual embedding of the last token to produce discrimination scores. Starting from BERT [9] as bidirectional discriminator, a triangular self-attention mask is applied for adapting it from the bidirectional to the unidirectional setting in our experiments, following [10]. Then, the generative discriminator is the same as the left-to-right one, the only difference being the size of the output layer that changes from $(hidden\_size, num\_classes)$ to $(hidden\_size, vocab\_size)$.

Experiments are made on two datasets from [39]: amazon_polarity which is a binary (positive or negative) online reviews classification task and AG_news which is a topic classification task with 4 labels (`world`, `sport`, `business` and `science`). These datasets allows to study results of cooperative generation on two rather different constraints and domains: applying polarity on online reviews and writing news about a specific topic. Also, AG_news allows to study the generalization to non-binary classification and texts with more diverse content. Each model is trained for 20 epochs using AdamW [23] with HuggingFace's trainer default parameters ($\beta_1 = 0.9, \beta_2 = 0.999, eps = 1e-8$) and a linear scheduler with no warmup. Batch size is set to the maximum that can fit in the memory of a Quadro RTX 6000 during GeDi training (4 for AG_news and 8 for amazon_polarity). Gradient accumulation is set to emulate a batch size of 128. For training GeDi, we set $\lambda = 0.6$ according to the authors (and did not observe significant difference when setting $\lambda = 0$ to strengthen the classification capacity).

3.1 Discrimination Strength

For discriminators, accuracy has an utmost importance: it defines how well it solves the intended task. In the context of cooperative generation, having a good accuracy on complete sequences is not sufficient: an informative output with uncomplete sequences is needed so that the discriminator can be used throughout the generation process. Thus, plotting the accuracy w.r.t. the number of input tokens gives information about the capacity of the model to guide the generation at different timesteps, the main property expected for discriminators in cooperative generation. Note that, following common practice in cooperative generation, the discriminator is trained on sequences of variable lengths to avoid a mismatch between training and test tasks.

Results reported in Fig. 1 show that every discriminator exhibits the same behavior: starting from random predictions, accuracy quickly increases with the input length until reaching a plateau. The expected ordering is observed: bidirectional models perform better than unidirectional models, which themselves perform better than generative ones. However, it should be noted that the gap is rather small and only appears when approaching the plateau. Favoring bidirectional models in accuracy-critical tasks is justified, but it is not necessarily clear that these small differences will reflect in the quality of cooperatively generated texts.

Please note that this corresponds to the accuracy on *in domain* data, and that complementary - non reported - experiments, on random sequences to be discriminated, showed however that GeDi is more robust to *out of domain* sequences: its discrimination scores are greatly closer to maximal uncertainty (i.e., $p(c|x) = 0.5$), than those of discriminative models which tend to greatly favor one class over the other ones in such cases. However, this may not impact results in generation, since such random samples are not likely to be observed during MCTS decoding, because of the language model prior guiding search towards in distribution sequences.

3.2 Generation Quality

To assess whether the – relatively small – differences in classification accuracies impact the results on cooperative generation with MCTS, we follow the PPL-MCTS [3] setup by constraining the generation process towards a desired class c using $p(c|x)$ given by the considered discriminator. Automatic metrics are used to study the quality of the guiding signal brought by the discriminator: 1) Accuracy corresponds to the average rate of generated sequences for any class c to be correctly classified as c by an oracle discriminator trained on disjoint data, 2) Self-BLEU [40] focuses on diversity across samples, by measuring BLEU scores between generated sequences, and 3) Oracle perplexity stands for the perplexity of an oracle LM trained on disjoint data, allowing to control the writing quality of generated texts. We used a bidirectional BERT model as oracle discriminator to get the most accurate evaluation possible. Language models are also BERT models with an LM head in order to use the same tokenizer. Average results over 500 sampled test texts using each type of discriminator on the two datasets are reported in Tab. 1. We also report results obtained using the vanilla LM likelihood $p(x)$ as back-propagated score in MCTS evaluation, to provide baseline results achievable without discriminators. Results are obtained using best performing hyper-parameters in the literature ($c_{puct} = 3$, temperature $\tau = 1$) and 50 iterations of MCTS per token, unless specified otherwise. We report statistical significance between each type of discriminator using t-test with p-value=0.01.

The difference of generation accuracy when using bidirectional and unidirectional discriminator shows that the difference in raw accuracy reflects in resulting samples when used for cooperative generation. The higher difference on amazon_polarity also results in a higher difference in generation accuracy. However, this difference is relatively limited and the generation does not seem to deviate too much using unidirectional discriminators. Results using generative discriminators are different, with a significantly greater drop of accuracy than between uni- and bi-directional models on AG_news, although the gap in raw accuracy is similar. More surprising is the result on amazon_polarity where, despite similar raw accuracies, we observe a 10-point drop of generation accuracy. We hypothesize that this is because the signal is not as informative: while raw accuracies are pretty similar, the average score attributed to the ground truth class in evaluation is significantly lower in the case

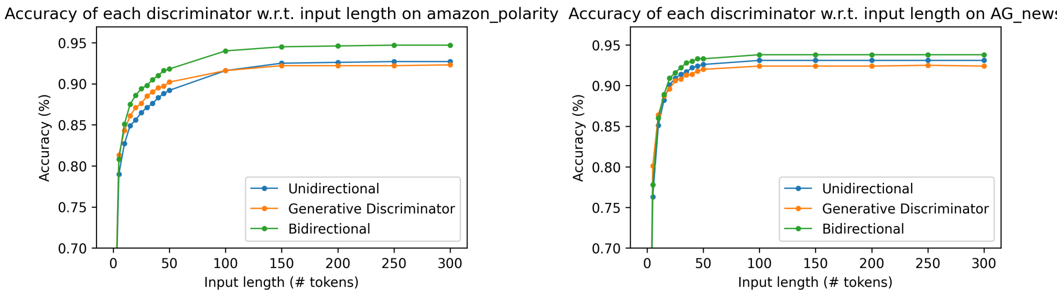

Figure 1: Accuracy (%) of the different type of discriminators w.r.t. the input length (# tokens)

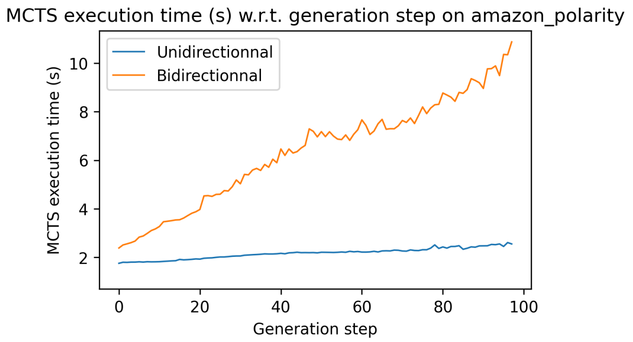

Figure 2: Execution time of MCTS iterations (s) w.r.t. generation step (averaged over 10 batchs of 30 sequences)

of GeDi. This means that its signal promotes less good solutions than standard discriminators when guiding the generation. The type of discriminator has no significant impact on the other metrics. Please note that the general difference of Self-BLEU and oracle perplexity between the two datasets is due to the difference in their content: AG_news is more diverse, which results in lower Self-BLEU and higher perplexity. Finally, we notice that doubling the number of MCTS iterations allows to increase the accuracy results of the unidirectional model, bridging the gap between both model for a still lower computational cost (see next section).

3.3 Computational Gain

Beyond generation accuracy, we are interested in computation complexity of the various models to be used in cooperative generation. Fig. 2 reports MCTS execution times w.r.t. each generation step t (i.e., time required to decode token at step t of any sequence), using a bidirectional model compared to a unidirectional one. Unsurprisingly, since the complexity is quadratic in the bidirectional case and only linear in the unidirectional one, the difference in generation time is significant, and increases linearly w.r.t. the sequence length. Note also that this difference increases with the number of MCTS iterations. At last, we note that the number of MTCS iterations with unidirectional discriminator can be much more than doubled compared to the case of bidirectional one, while keeping the computational cost significantly lower, even for small text sequences.

In the case of the generative discriminator, a great potential computational gain may arise from the fact that discrimination scores can be computed for every child of an expanded node at once. More specifically, while computing scores for each of the $|\mathcal{V}|$ children nodes would cost $|\mathcal{V}|$ forward passes in the case of discriminative classifiers, it only requires $|C|$ forward passes for generative classifiers (i.e., one pass per class for getting all scores, rather than one pass per child node). Since usually $|C| << |\mathcal{V}|$, the use of generative discriminator could be way advantageous and allow to increase the number of MCTS iterations to expect to, at least, fill the gap with accuracy results of discriminative approaches.

However, this potential gain heavily depends on the exploration of the tree and therefore the parameter c_{puct}. If less than $|C|$ children are considered at each level of the tree, then the generative approach is at least as costly as the discriminative one and can even be more costly. Indeed, we empirically observed that for usual value $c_{puct} = 3$, generative discriminators needs in average 1 685 more forward passes on amazon_polarity (where $|C|$ is only 2), meaning there is more depth than width explorations. Increasing c_{puct} decreases this difference but also the resulting generation accuracy. At $c_{puct} = 15$, the accuracy already drops for 10 points and the difference is still to the disadvantage of GeDi for more than 600 forward pass. These results show that generative discriminators are only beneficial if exploration is wider than deeper, which is not the case for MCTS operating points. This is consistent with GeDi results [19], which observed an important gain in a beam search decoding approach where the width is crucial. These new results suggest to seek at ways for better leveraging this GeDi potential with more efficient exploration in width of the MCTS or to use methods that do it by construction as beam search.

4 CONCLUSION

Cooperative generation has proven to be an effective way to augment traditional text generation with external information from a discriminator. While transformers with bidirectional attention are usually preferred for discriminative tasks, they are not autoregressive and are therefore much more expensive when used to guide generation. Although a little less precise, unidirectional transformers allow to achieve very similar results for a much more reasonable and consistent cost. As a consequence, our study shows that unidirectional discriminators should be preferred for cooperative generation, for which slight accuracy drops can be balanced by reinvesting part of the computational gain. Given the size of

Value	amazon_polarity Accuracy ↑	5 - Self-BLEU ↓	Oracle perplexity ↓	AG_news Accuracy ↑	5 - Self-BLEU ↓	Oracle perplexity ↓
$p(x)$	70.8	0.652	10.49	86.6	0.306	29.08
Bidirectional	96.0*	0.531*	12.25	94.8*	0.319	29.13
Unidirectional	93.0*	0.528*	11.98	93.4	0.313	29.99
Unidirectional (100 its)	93.6*	0.522*	10.73	94.6*	0.323	30.92
Generative discriminator	84.4	0.576	11.92	91.8	0.321	29.43

Table 1: Performance of MCTS w.r.t. the metric to optimize on amazon_polarity (left) and AG_news (right) datasets. $*$ indicates statistically significant improvement against Generative Discriminator. Note that no model demonstrated significant improvement over unidirectional discriminator.

usual vocabularies, generative discriminators seem very interesting at first glance to allow wider search. However, while achieving similar results in terms of classification accuracy, scoring the whole vocabulary comes at the price of a less informative signal. Moreover, although counter-intuitive, this width is not necessarily useful as shown by the search performed by the state-of-the-art Monte Carlo Tree Search, which usually explores more in depth than in width. Thus, such models will prove useful when used with methods that make particular use of this width information. We leave such explorations for future work.

To allow reproduction and further experiments on this subject, the code used for our experiments is made available for the community at https://github.com/NohTow/PPL-MCTS/tree/main/teammates.

REFERENCES

[1] Anton Bakhtin, Yuntian Deng, Sam Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. 2021. Residual Energy-Based Models for Text. *Journal of Machine Learning Research* 22, 40 (2021), 1–41. http://jmlr.org/papers/v22/20-326.html

[2] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, Madeleine Clare Elish, William Isaac, and Richard S. Zemel (Eds.). ACM, 610–623. https://doi.org/10.1145/3442188.3445922

[3] Antoine Chaffin, Vincent Claveau, and Ewa Kijak. 2021. PPL-MCTS: Constrained Textual Generation Through Discriminator-Guided MCTS Decoding. *CoRR* abs/2109.13582 (2021). arXiv:2109.13582 https://arxiv.org/abs/2109.13582

[4] Xingyuan Chen, Ping Cai, Peng Jin, Hongjun Wang, Xinyu Dai, and Jiajun Chen. 2020. Adding A Filter Based on The Discriminator to Improve Unconditional Text Generation. *arXiv preprint arXiv:2004.02135* (2020).

[5] Vincent Claveau. 2021. Neural text generation for query expansion in information retrieval. In *WI-IAT 2021 - 20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (Proceedings of the WI-IAT Conference)*. IEEE, Melbourne, Australia, 1–8. https://doi.org/10.1145/3486622.3493957

[6] Rémi Coulom. 2006. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In *Computers and Games, 5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. Revised Papers (Lecture Notes in Computer Science, Vol. 4630)*, H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. Donkers (Eds.). Springer, 72–83. https://doi.org/10.1007/978-3-540-75538-8_7

[7] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=H1edEyBKDS

[8] Carnegie-Mellon University.Computer Science Dept. 2018. Speech understanding systems: summary of results of the five-year research effort at Carnegie-Mellon University. https://doi.org/10.1184/R1/6609821.v1

[9] Jacob Devlin, 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[10] Li Dong, 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 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 13042–13054. https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html

[11] Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, 889–898. https://doi.org/10.18653/v1/P18-1082

[12] Saadia Gabriel, Antoine Bosselut, Jeff Da, Ari Holtzman, Jan Buys, Kyle Lo, Asli Celikyilmaz, and Yejin Choi. 2021. Discourse Understanding and Factual Consistency in Abstractive Summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, 435–447. https://aclanthology.org/2021.eacl-main.34/

[13] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020 (Findings of ACL, Vol. EMNLP 2020)*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 3356–3369. https://doi.org/10.18653/v1/2020.findings-emnlp.301

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144. https://doi.org/10.1145/3422622

[15] Di He, Hanqing Lu, Yingce Xia, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2017. Decoding with value networks for neural machine translation. *Advances in Neural Information Processing Systems* 30 (2017).

[16] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rygGQyrFvH

[17] Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to Write with Cooperative Discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, 1638–1649. https://doi.org/10.18653/v1/P18-1152

[18] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *CoRR* abs/1909.05858 (2019). arXiv:1909.05858 http://arxiv.org/abs/1909.05858

[19] Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq R. Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative Discriminator Guided Sequence Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 4929–4952. https://aclanthology.org/2021.findings-emnlp.424

[20] Sylvain Lamprier, Thomas Scialom, Antoine Chaffin, Vincent Claveau, Ewa Kijak, Jacopo Staiano, and Benjamin Piwowarski. 2022. Generative Cooperative Networks for Natural Language Generation. *CoRR* abs/2201.12320 (2022). arXiv:2201.12320 https://arxiv.org/abs/2201.12320

[21] Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Jean-Baptiste Lespiau, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. 2021. Machine Translation Decoding beyond Beam Search. In *Proceedings of the 2021 Conference*

on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 8410–8434. https://aclanthology.org/2021.emnlp-main.662

[22] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 9459–9474. https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[23] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bkg6RiCqY7

[24] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2022. On the Study of Transformers for Query Suggestion. *ACM Trans. Inf. Syst.* 40, 1 (2022), 18:1–18:27. https://doi.org/10.1145/3470562

[25] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. WebGPT: Browser-assisted question-answering with human feedback. *CoRR* abs/2112.09332 (2021). arXiv:2112.09332 https://arxiv.org/abs/2112.09332

[26] Andrew Y. Ng and Michael I. Jordan. 2001. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani (Eds.). MIT Press, 841–848. https://proceedings.neurips.cc/paper/2001/hash/7b7a53e239400a13bd6be6c91c4f6c4e-Abstract.html

[27] Vishal Pallagani and Biplav Srivastava. 2021. A Generic Dialog Agent for Information Retrieval Based on Automated Planning Within a Reinforcement Learning Platform. *Bridging the Gap Between AI Planning and Reinforcement Learning (PRL)* (2021).

[28] Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data augmentation for abstractive query-focused multi-document summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2021). Online.* 13666–13674.

[29] Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

[30] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. Deep Reinforcement Learning-Based Image Captioning With Embedding Reward. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[31] Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. Discriminative Adversarial Search for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 8555–8564. http://proceedings.mlr.press/v119/scialom20a.html

[32] Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2021. To Beam Or Not To Beam: That is a Question of Cooperation for Language GANs. *Advances in neural information processing systems* (2021).

[33] David Silver, 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. *nature* 550, 7676 (2017), 354–359.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[35] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, Qun Liu and David Schlangen (Eds.). Association for Computational Linguistics, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[36] Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and Discriminative Text Classification with Recurrent Neural Networks. *CoRR* abs/1703.01898 (2017). arXiv:1703.01898 http://arxiv.org/abs/1703.01898

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

[38] Ruifeng Yuan, Zili Wang, and Wenjie Li. 2021. Event Graph based Sentence Fusion. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 4075–4084.

[39] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 649–657. https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html

[40] Yaoming Zhu, 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, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 1097–1100. https://doi.org/10.1145/3209978.3210080