A Call for Clarity in Beam Search: How It Works and When It Stops

Jungo Kasai[♡] **Keisuke Sakaguchi**[♣□] **Ronan Le Bras**[◇]
Dragomir Radev[♠] **Yejin Choi**[♥◇] **Noah A. Smith**[♥◇]

[♡]Toyota Technological Institute at Chicago [♣]Tohoku University [□]RIKEN
[◇]Allen Institute for AI [♠]Department of Computer Science, Yale University
[♥]Paul G. Allen School of Computer Science & Engineering, University of Washington
jkasai@ttic.edu

Abstract

Text generation with beam search has proven successful in a wide range of applications. We point out that, though largely overlooked in the literature, the commonly-used implementation of beam decoding (e.g., Hugging Face Transformers and fairseq) uses a *first come, first served* heuristic: it keeps a set of already completed sequences over time steps and stops when the size of this set reaches the beam size. Based on this finding, we introduce a *patience factor*, a simple modification to this beam decoding implementation, that generalizes the stopping criterion and provides flexibility to the depth of search. Empirical results demonstrate that adjusting this patience factor improves decoding performance of strong pretrained models on news text summarization and machine translation over diverse language pairs, with a negligible inference slowdown. Our approach only modifies one line of code and can be thus readily incorporated in any implementation. Further, we find that different versions of beam decoding result in large performance differences in summarization, demonstrating the need for clarity in specifying the beam search implementation in research work. Our code will be available upon publication.

Keywords: generation, decoding, beam search, inference, summarization, machine translation

1. Introduction

Beam search has become a dominant inference algorithm for a wide range of language generation tasks, such as machine translation (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017), summarization (Nallapati et al., 2016; See et al., 2017), and image captioning (Anderson et al., 2018; Li et al., 2020). Beam decoding is an approximate, pruned version of breadth-first search that seeks the highest-probability sequence under an autoregressive (left-to-right) language generation model. In this work, we examine a popular implementation of beam decoding and propose a simple modification (one line of code) that improves the decoding performance of strong, neural language generation models (Fig. 1).

We first bring attention to implementation variations of beam decoding that are largely ignored in the literature: a widely-used implementation of beam language decoding (e.g., fairseq, Ott et al., 2019; Hugging Face's Transformers, Wolf et al., 2020)[1] follows a *first come, first served* (FCFS) heuristic: when a total of k finished candidates is found (k is the beam size), it returns the best one from the k candidates and discards all of the current, unfinished k sequences in the beam. Thus in practice, beam size k determines both the breadth and depth of search. We propose a *patience fac-*

FCFS Beam Decoding with Controlled Patience

```
k: beam size, M: maximum length, V: Vocabulary
score(·): scoring function, p: patience factor.
B_0 ← {⟨0, BOS⟩}, F_0 ← ∅
for t ∈ {1, ..., M−1} :
    H ← ∅, F_t ← F_{t−1}
    for ⟨s, y⟩ ∈ B_{t−1} :    # Expansion.
        for y ∈ V :
            s ← score(y ∘ y),   H.add(⟨s, y ∘ y⟩)
    B_t ← ∅
    while |B_t| < k :    # Find top k w/o EOS from H.
        ⟨s, y⟩ ← H.max()
        if y.last() = EOS :
            F_t.add(⟨s, y⟩)    # Finished hypotheses.
        else B_t.add(⟨s, y⟩)
        if |F_t| ≥ k · p :    # Originally, p=1.
            return F_t.max()
        H.remove(⟨s, y⟩)
return F_t.max()
```

Figure 1: First come, first served (FCFS) beam decoding with patience factor p. The common implementation can be considered as a special case where $p = 1$. The highlighted line is the *only* modification that this work introduces for performance improvement. F_t: already completed sequences; B_t: beam of continuing sequences. H_t: expanded hypotheses before the top-k operation. The input sequence to score is omitted.

[1] https://github.com/pytorch/fairseq/blob/main/fairseq/sequence_generator.py; https://github.com/huggingface/transformers/blob/master/src/transformers/generation_utils.py.

LREC-COLING 2024, pages 77–90
20-25 May, 2024.

tor (Fig. 1) that decomposes these two roles and controls how many finished candidates have to be found before terminating the decoding. The patience factor generalizes the commonly-used implementation and provides flexibility in the depth of beam search by changing the stopping criterion.

We apply the one-line modification to strong off-the-shelf transformer models without any change to the trained models for machine translation (Tang et al., 2021) and text summarization (Lewis et al., 2020). Our experiments demonstrate that the simple modification improves performance on the CNN/Dailymail (Hermann et al., 2015) and XSUM (Narayan et al., 2018) news summarization tasks and the WMT 2020/2021 machine translation tasks (Barrault et al., 2020; Akhbardeh et al., 2021) across diverse language pairs. Further, the introduction of the patience factor only results in a negligible inference slowdown, confirming its practical advantage in downstream applications.

Our analysis shows that, while the performance gain is sensitive to hyperparameters of beam decoding (beam size and length penalty; Johnson et al., 2017), the patience factor is consistently beneficial. Moreover, we extensively compare our results with the *vanilla* implementation of beam search that much prior work assumes (Meister et al., 2020b; Stahlberg and Byrne, 2019, *inter alia*), though it is not used in popular libraries in practice. Empirically, we found that the vanilla algorithm performs competitively with FCFS on machine translation but **substantially underperforms on summarization**. Therefore, although much prior work on text decoding implicitly assumes the vanilla version of beam decoding, researchers should specify which implementation is used when reporting results.

2. A Call for Clarity in Beam Decoding

Vanilla and FCFS Implementations Beam decoding has been applied to sequence-to-sequence models (Graves, 2012; Boulanger-Lewandowski et al., 2013a,b), and it is now used in many state-of-the-art systems for language generation tasks (Zhang et al., 2020, 2021; Tran et al., 2021; Raffel et al., 2020, *inter alia*). **While largely ignored in the literature**, beam decoding has two major variations (Figs. 1 and 2). They differ primarily in the treatment of finished sequences with the EOS symbol at the end: FCFS collects finished sequences in a *first come, first served manner* and removes them from the beam (Line 11, Fig. 1), whereas the vanilla version finds the top

Vanilla Beam Decoding

k: beam size, M: maximum length,
$\mathcal{V}$: Vocabulary, score$(\cdot)$: scoring function.

1: $B_0 \leftarrow \{\langle 0, \text{BOS}\rangle\}$
2: **for** $t \in \{1, \dots, M-1\}$:
3: **for** $\langle s, \mathbf{y}\rangle \in B_{t-1}$:
4: **if** $\mathbf{y}$.last() $=$ EOS :
5: H.add($\langle s, \mathbf{y}\rangle$)
6: **continue**
7: **for** $y \in \mathcal{V}$:
8: $s \leftarrow$ score($\mathbf{y} \circ y$), H.add($\langle s, \mathbf{y} \circ y\rangle$)
9: $B_t \leftarrow \varnothing$
10: **while** $|B_t| < k$: # Find top k from H.
11: $\langle s, \mathbf{y}\rangle \leftarrow H$.max(), B_t.add($\langle s, \mathbf{y}\rangle$)
12: H.remove($\langle s, \mathbf{y}\rangle$)
13: **if** $\mathbf{y}$.last() $=$ EOS, $\forall \mathbf{y} \in B_t$: # All finished.
14: **return** B_t.max()
15: **return** B_t.max()

Figure 2: The vanilla version of beam decoding that much prior work assumes **departs from popular libraries, such as Hugging Face** Transformers. The top-k operation is applied over H, the union of the finished and continuing sequences. This is implemented, for example, in the TensorFlow Addons library (Abadi et al., 2015).[2] See also Stahlberg and Byrne (2019); Meister et al. (2020b).

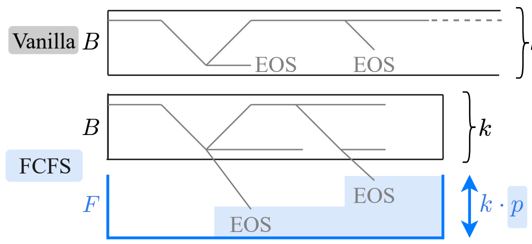

Figure 3: FCFS with patience factor p vs. vanilla beam decoding. k denotes the beam size. FCFS stores finished sentences in F, but they stay in (and later may fall off from) beam B during vanilla decoding. $k \cdot p$ determines the size of F. The illustration of beam decoding here is inspired by Huang et al. (2012).

k sequences, including both finished and unfinished sequences (Line 5 in Fig. 2). Our experiments will show that this difference can affect the downstream performance substantially, especially on summarization (§3.2).

Further comparing Figs. 1 and 2, we see their difference in terms of the breadth and depth of search. Given the same beam size k, FCFS has a wider breadth since it collects k unfinished sequences at every step regardless of how many sequences are finished with the EOS symbol.[3] The

[2] https://www.tensorflow.org/addons/api_docs/python/tfa/seq2seq/BeamSearchDecoder.

[3] In practice, this is implemented by taking the top $2k$ sequences at every step. We find at most k EOS symbols, so there are always at least k *unfinished*

vanilla algorithm decodes until all top-k sequences are finished (Line 13, Fig. 2), and therefore it tends to result in deeper search. FCFS, in contrast, terminates when a total of k finished sequences is found.

Patience Factor for FCFS Beam size k in FCFS thus controls both the breadth and stopping criterion (i.e., depth) of search. We introduce the patience factor (Line 13, Fig. 1) that relaxes this assumption and separates the stopping criterion from the search breadth. Fig. 3 illustrates this patience factor as well as the difference between the FCFS and vanilla algorithms. The one-line change generalizes FCFS ($p=1$) and adds flexibility. We will show that this flexibility is beneficial, espeically on news text summarization (§3.2).

3. Experiments

We compare implementation variations of beam decoding on text summarization and machine translation over a wide range of language pairs.

3.1. Experimental Setup for Beam Decoding

In addition to **greedy search**, we evaluate three beam decoding variations on machine translation and summarization: **vanilla**, **FCFS**, and **FCFS with the patience factor**.[4] For machine translation, we use multilingual BART (Tang et al., 2021), a strong, pretrained transformer model,[5] and WMT 2020/2021 news test data (Barrault et al., 2020; Akhbardeh et al., 2021) for four diverse language pairs (eight directions): WMT 2020 for EN$\leftrightarrow$PL (Polish) and 2021 for EN$\leftrightarrow$DE (German), EN$\leftrightarrow$JA (Japanese), and EN$\leftrightarrow$ZH (Chinese). We apply beam decoding with the same hyperparameters as Tang et al. (2021): beam size 5 and length penalty 1. We measure performance with the COMET score (Rei et al., 2020a,b), a state-of-the-art evaluation metric based on multilingual contextual representations. For summarization, we experiment with the CNN/Dailymail (CNNDM, Hermann et al., 2015) and XSUM (Narayan et al., 2018) datasets. We apply the off-the-shelf BART models (Lewis et al., 2020) that are finetuned on each dataset.[6]

Performance is measured with ROUGE scores (Lin, 2004). We follow the original setting in Lewis et al. (2020): beam sizes 4 and 6 and length penalty 2 and 1 for CNNDM and XSUM, respectively. More experimental details are described in the appendix (following the submission guideline, the appendix will be added in the final version).

We experiment with the same patience factor on all datasets for each task, based on our preliminary development: $p=2$ for machine translation and $p=0.5$ for summarization. Here we avoid additional effort and demonstrate the practical value of our simple modification. We present detailed sensitivity analysis over p in §3.3. Note that a larger p implies deeper search, but deeper (and thus more accurate) search does not necessarily result in better generations, similar to the *beam search curse* (Koehn and Knowles, 2017).

3.2. Results

Seen in Table 1 are results from our experiments. FCFS with the patience factor outperforms the widely-used FCFS variation across the board; e.g., 53.0 vs. 52.1 on EN$\rightarrow$PL. Particularly noteworthy are the performance gains on the two summarization datasets; e.g., 31.2 vs. 30.3 ROUGE-L on CNNDM. Comparing vanilla decoding and FCFS, we see that the former outperforms the latter (and is competitive with or slightly better than FCFS w/ p) on machine translation but underperforms substantially on summarization; e.g., 34.4 vs. 33.1 ROUGE-L on XSUM. Vanilla beam decoding even performs worse than greedy decoding in many cases. This large degradation from the vanilla implementation illustrates **the importance of specifying the version of beam decoding**, which is not commonly done in practice.

3.3. Analysis

Here we use the standard dev. split from the XSUM dataset and news test 2020 EN$\rightarrow$DE and ZH$\rightarrow$EN data. We fixed the value of p for each task so far, but Fig. 4 explores varying patience factors and their effects on the performance (A: EN$\rightarrow$DE; B: XSUM) and the inference speed (C). The translation performance improves with larger patience factors with diminishing gains. Surprisingly, summarization benefits from patience factors smaller than the original value of 1, possibly due to the nature of the summarization task that aims to generate concise text. Note that this observation does *not* contradict with prior work because the search accuracy of beam decoding does *not* always correlate with the output quality (Koehn and Knowles, 2017; Stahlberg and Byrne, 2019; Meister et al., 2020a, *inter alia*). Regardless, our patience factor provides useful flexibility for any generation task.

sequences. See https://github.com/huggingface/transformers/.

[4]Other beam decoding variants target various purposes, like output diversity (stochastic beam search; Kool et al.; Meister et al., 2021) or efficiency (best-first beam search; Meister et al., 2020b). Here we focus on the commonly-adopted implementations and the top-1 output quality.

[5]https://github.com/pytorch/fairseq/tree/main/examples/multilingual#mbart50-models.

[6]https://github.com/pytorch/fairseq/tree/main/examples/bart.

	WMT 2020/2021 Machine Translation ($p = 2$)								Summarization ($p = 0.5$)					
	EN↔DE		**EN↔JA**		**EN↔PL**		**EN↔ZH**		**CNNDM**			**XSUM**		
Algorithm	→	←	→	←	→	←	→	←	R-2	R-3	R-L	R-2	R-3	R-L
Greedy	43.7	66.2	33.6	9.5	46.0	53.5	32.5	23.5	21.1	11.9	30.7	19.8	10.7	34.3
Vanilla	48.2	66.3	**38.7**	**15.7**	52.7	58.2	**33.9**	29.9	19.2	11.0	28.0	19.5	10.7	33.1
FCFS	47.9	66.2	38.0	15.0	52.1	58.1	33.7	29.6	20.4	11.6	30.3	20.4	11.4	34.4
FCFS w/ p	**48.3**	**66.4**	38.4	15.6	**53.0**	**58.4**	33.8	**30.2**	**21.4**	**12.4**	**31.2**	**21.0**	**11.8**	**35.4**

Table 1: We evaluate three beam decoding variations, as well as greedy decoding, on the machine translation and summarization test data with the COMET score (Rei et al., 2020b) and ROUGE scores (ROUGE-2/3/L), respectively. FCFS w/ p indicates our FCSF algorithm with the patience factor ($p = 2$ for machine translation and $p = 0.5$ for summarization). For CNNDM, we used 100 test articles with 10 human-written references (Kryscinski et al., 2019).

As expected, generation slows down as p increases (Fig. 4C). The inference slowdown from around $p = 2$ is still negligible, again showing the practicality of our method. Fig. 5 explores the performance gains from the patience factor over varying beam sizes. The amount of improvement changes, but the patience factor is generally beneficial. We see similar trends for various values of the length penalty (see the appendix).

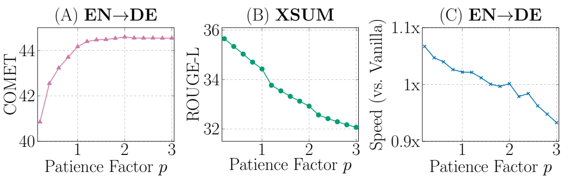

Figure 4: Effects of varying patience factors p on the dev. score (A and B) and inference speed (C). The inference speed is measured with batch size 20, relative to the vanilla version on the same single Nvidia A100-SXM GPU. Other languages pairs were similar to EN→DE (A). CNNDM also had similar trends to XSUM (B).

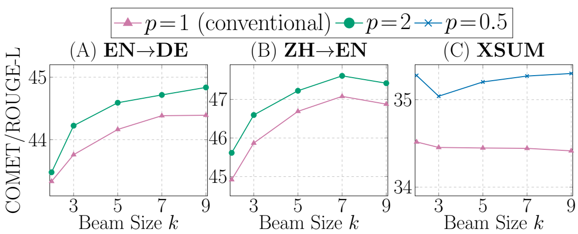

Figure 5: Effects of controlled patience on the dev. data over varying beam sizes. The length penalty value is 1. We evaluate with COMET for machine translation and ROUGE-L for XSUM summarization.

4. Further Related Work

Stopping Criteria for Beam Decoding The patience factor changes the stopping criterion and adds flexibility in the search depth of the widespread variation of beam decoding. Similarly, several prior works studied stopping criteria to improve machine translation (Huang et al., 2017; Yang et al., 2018; Ma et al., 2019). Our experiments are consistent with their findings: stopping criteria that yield deeper search can improve the machine translation performance.

Breadth of Beam Decoding Much prior work explored downstream effects of the search breadth (Koehn and Knowles, 2017; Murray and Chiang, 2018; Ott et al., 2018a; Cohen and Beck, 2019; Stahlberg and Byrne, 2019, *inter alia*). Beam decoding with larger beam sizes can find sequences with higher scores but lead to performance degradation (often called the *beam search curse*; Yang et al., 2018). Recent work (Meister et al., 2020a) argued that beam decoding with small beams introduces bias that is related to the uniform information density of human-produced text (Levy, 2005). Freitag and Al-Onaizan (2017) proposed a method to adaptively shrink the beam width based on the partial scores to speed up inference. We focused on the stopping criteria (i.e., depth) and separated them from the breadth of the common implementation.

5. Conclusion

We brought attention to the crucial yet overlooked implementation differences of beam search. We then introduced the patience factor for the widespread implementation. Our experiments showed that the patience factor improves the generation performance especially on summarization, with an insignificant slowdown in generation. As it only requires a minimal change in code, we hope that many researchers and practitioners of language generation will benefit from our simple yet effective modification.

Limitations

We evaluated our decoding method both on machine translation and news summarization. Our machine translation experiments span diverse languages, including morphologically rich languages (e.g., Japanese and Polish) and languages with non-Latin scripts (e.g., Japanese and Chinese). Nonetheless, our summarization experiments are limited to English and the news domain mainly due to our budget constraints. There are also many other language generation tasks for which our method can be useful. Since our improvement only requires one line of code, we hope that practitioners will implement it for the domain and the task of their interest and further assess how our decoding algorithm performs over a wider range of applications.

Evaluating language generation remains a challenging research problem. We carefully set up our experiments to mitigate potential evaluation issues. The WMT 2020/2021 test data consist only of news text written in the original language, in contrast to the test data from WMT 2018 (Bojar et al., 2018) or earlier. For example, the WMT 2021 EN→DE and DE→EN test data come from completely different documents. This avoids the *translationese effect* that would overestimate the translation performance due to the simplicity of translated text (Graham et al., 2020). Moreover, some language pairs in the WMT 2020 and 2021 test data have multiple references per instance, which increases the correlation of automatic evaluations with human judgment (Kasai et al., 2022a). We presented results using automatic metrics from recent work (Rei et al., 2020b) as well as conventional, n-gram overlap metrics (Papineni et al., 2002; Lin, 2004). Recent automatic metrics have shown to have higher correlation with human judgements, but human judgments are sometimes inconsistent, especially when crowdsourced (Clark et al., 2021; Kasai et al., 2022b). Since our decoding method is a generalization of the widely-used beam search algorithm, we hope that it will be tested and used in real-world systems of language generation.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding long and structured inputs in transformers. In *Proc. of EMNLP*.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proc. of WMT*.

Maximilian Alber, Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Fei Sha. 2017. An empirical study on the properties of random bases for kernel methods. In *Proc. of NeurIPS*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. of CVPR*.

Haim Avron, L. Kenneth Clarkson, and P. David and Woodruff. 2017. Faster kernel ridge regression using sketching and preconditioning. *SIAM J. Matrix Analysis Applications*.

Haim Avron, Vikas Sindhwani, Jiyan Yang, and Michael W. Mahoney. 2016. Quasi-Monte Carlo feature maps for shift-invariant kernels. *Journal of Machine Learning Research*, 17(120):1–38.

Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. 2016. Using fast weights to attend to the recent past. In *Proc. of NeurIPS*.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *Proc. of ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proc. of WMT*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

S. Bochner. 1955. *Harmonic Analysis and the Theory of Probability*. University of California Press.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proc. of WMT*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proc. of WMT*.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2013a. Audio chord recognition with recurrent neural networks. In *Proc. of ISMIR*.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2013b. High-dimensional sequence transduction. In *Proc. of ICASSP*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv: 2005.14165*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proc. of IWSLT*.

K. Chaudhuri, C. Monteleoni, and A.D. Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109.

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2019. Recurrent positional embedding for neural machine translation. In *Proc. of EMNLP*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv: 1904.10509*.

Kyunghyun Cho, 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 *Proc. of EMNLP*.

Youngmin Cho and Lawrence K. Saul. 2009. Kernel methods for deep learning. In *Proc. of NeurIPS*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *Proc. of ICLR*.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proc. of ACL*.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proc. of ICLR*.

Eldan Cohen and Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *Proc. of ICML*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proc. of ACL*.

Kornél Csernai. 2017, accessed September 1, 2020. *First Quora Dataset Release: Question Pairs*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proc. of ACL*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. Universal transformers. In *Proc. of ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of ICLR*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proc. of NAACL*.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proc. of ACL*.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *Proc. of ICLR*.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proc. of NGT*.

Yingbo Gao, Christian Herold, Weiyue Wang, and Hermann Ney. 2019. Exploring kernel functions in the softmax layer for contextual word classification. In *International Workshop on Spoken Language Translation*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Anirudh Goyal, Aniket Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Mozer, and Yoshua Bengio. 2021. Coordination among neural modules through a shared global workspace.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Translationese in machine translation evaluation.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. In *Representation Learning Workshop*.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. 2015. Learning to transduce with unbounded memory. In *Proc. of NeurIPS*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *Proc. of ICLR*.

Jie Hao, Xing Wang, Baosong Yang, Longyue Wang, Jinfeng Zhang, and Zhaopeng Tu. 2019. Modeling recurrence for transformer. In *Proc. of NAACL*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proc. of NeurIPS*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPs Deep Learning and Representation Learning Workshop*.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. 2020. Axial attention in multidimensional transformers. *arXiv: 1912.12180*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. 2008. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proc. of ICML*.

Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proc. of NAACL*.

Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to finish? optimal beam search for neural text generation (modulo beam size). In *Proc. of EMNLP*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*.

Armand Joulin and Tomás Mikolov. 2015. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Proc. of NeurIPS*.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Anna Potapenko, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Martin Steinegger, Michalina Pacholska, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. 2021. High accuracy protein structure prediction using deep learning. *Nature*, 596:583–589.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2021a. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *Proc. of ICLR*.

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. 2021b. Finetuning pretrained transformers into RNNs. In *Proc. of EMNLP*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander R. Fabbri, Yejin Choi, and Noah A. Smith. 2022a. Bidimensional leaderboards: Generate and evaluate language hand in hand. In *Proc. of NAACL*.

Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith. 2022b. Transparent human evaluation for image captioning. In *Proc. of NAACL*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Francois Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proc. of ICML*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proc. of ICLR*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proc. of ICLR*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proc. of NGT*.

Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: The Gumbel-top-k trick for sampling sequences without replacement. In *Proc. of ACL*.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proc. of EMNLP*.

Frances Y. Kuo and Dirk Nuyens. 2016. Application of quasi-monte carlo methods to elliptic pdes with random diffusion coefficients: A survey of analysis and implementation. *Foundations of Computational Mathematics*, 16(6):1631–1696.

Hung Le, Truyen Tran, and Svetha Venkatesh. 2020. Self-attentive associative memory. In *Proc. of ICML*.

Quoc Le, Tamas Sarlos, and Alex Smola. 2013. Fastfood - approximating kernel expansions in loglinear time. In *Proc. of ICML*.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proc. of ICML*.

Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. 2018. Simple recurrent units for highly parallelizable recurrence. In *Proc. of EMNLP*.

Roger Levy. 2005. *Probabilistic Models of Word Order and Syntactic Discontinuity*. Ph.D. thesis, Stanford University.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Proc. of NeurIPS*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proc. of ECCV*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proc. of Text Summarization Branches Out*.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Łukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by summarizing long sequences. In *Proc. of ICLR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Ilya Tolstikhin. 2015. Towards a learning theory of cause-effect inference. In *Proc. of ICML*.

Mingbo Ma, Renjie Zheng, and Liang Huang. 2019. Learning to stop in structured prediction for neural machine translation. In *Proc. of NAACL*.

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. 2021. Luna: Linear unified nested attention. In *Proc. of NeurIPS*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proc. of ACL*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proc. of ACL*.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 metrics shared task. In *Proc. of WMT*.

Clara Meister, Afra Amini, Tim Vieira, and Ryan Cotterell. 2021. Conditional Poisson stochastic beams. In *Proc. of EMNLP*.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020a. If beam search is the answer, what was the question? In *Proc. of EMNLP*.

Clara Meister, Tim Vieira, and Ryan Cotterell. 2020b. Best-first beam search. *TACL*.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and Optimizing LSTM Language Models. In *Proc. of ICLR*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proc. of ICLR*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Proc. of NeurIPS*.

Thomas Miconi, Kenneth Stanley, and Jeff Clune. 2018. Differentiable plasticity: training plastic neural networks with backpropagation. In *Proc. of ICML*.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proc. of EMNLP*.

Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer. 2019. Transformers with convolutional context for ASR. *arXiv: 1904.11660*.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proc. of WMT*.

Sebastian Nagel. 2016. News dataset available. https://commoncrawl.org/2016/10/news-dataset-available/.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proc. of CoNLL*.

Nikita Nangia and Samuel Bowman. 2018. ListOps: A diagnostic dataset for latent tree learning. In *Proc. of NAACL Student Research Workshop*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proc. of EMNLP*.

Ani Nenkova. 2006. Summarization evaluation for text and speech: issues and approaches. In *Proc. of INTERSPEECH*.

Junier Oliva, William Neiswanger, Barnabas Poczos, Eric Xing, Hy Trac, Shirley Ho, and Jeff Schneider. 2015. Fast function to function regression. In *Proc. of AISTATS*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018a. Analyzing uncertainty in neural machine translation. In *Proc. of ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018b. Scaling neural machine translation. In *Proc. of WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Emilio Parisotto, H. Francis Song, Jack W. Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant M. Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, Matthew M. Botvinick, Nicolas Heess, and Raia Hadsell. 2020. Stabilizing transformers for reinforcement learning. In *Proc. of ICML*.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *Proc. of ICML*.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. 2021. Random feature attention. In *Proc. of ICLR*.

Hao Peng, Roy Schwartz, Dianqi Li, and Noah A. Smith. 2020. A mixture of $h-1$ heads is better than h heads. In *Proc. of ACL*.

Hao Peng, Roy Schwartz, Sam Thomson, and Noah A. Smith. 2018. Rational recurrences. In *Proc. of EMNLP*.

Matt Post. 2018a. A call for clarity in reporting BLEU scores. In *Proc. of WMT*.

Matt Post. 2018b. A call for clarity in reporting BLEU scores. In *Proc. of WMT*.

Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise self-attention for long document understanding. In *Findings of EMNLP*.

Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The ACL Anthology network. In *Proc. of the Workshop on Text and Citation Analysis for Scholarly Digital Libraries*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *Proc. of ICLR*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. In *Proc. of NeurIPS*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. of EMNLP*.

Ankit Singh Rawat, Jiecao Chen, Felix Xinnan X Yu, Ananda Theertha Suresh, and Sanjiv Kumar. 2019. Sampled softmax with random Fourier features. In *Proc. of NeurIPS*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proc. of EMNLP*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proc. of WMT*.

Aurko Roy, Mohammad Taghi Saffar, David Grangier, and Ashish Vaswani. 2020. Efficient content-based sparse attention with routing transformers. *TACL*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv: 1910.01108*.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *Proc. of ICML*.

J. Schmidhuber. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139.

J. Schmidhuber. 1993. Reducing the ratio between learning complexity and number of time varying variables in fully recurrent nets. In *Proc. of ICANN*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. of ACL*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Q-BERT: Hessian based ultra low precision quantization of BERT. In *Proc. of AAAI*.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. 2021. Efficient attention: Attention with linear complexities. In *Proc. of WACV*.

Aman Sinha and John C Duchi. 2016. Learning kernels with random features. In *Proc. of NeurIPS*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? In *Proc. of EMNLP*.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. In *Proc. of ACL*.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proc. of NeurIPS*.

Yitong Sun. 2019. *Random Features Methods in Supervised Learning*. Ph.D. thesis, The University of Michigan.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. of NeurIPS*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of ACL*.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2020a. Synthesizer: Rethinking self-attention in transformer models. *arXiv: 2005.00743*.

Yi Tay, Dara Bahri, Liu Yang, Don Metzler, and Da-Cheng Juan. 2020b. Sparse sinkhorn attention. In *Proc. of ICML*.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena: A benchmark for efficient transformers. In *Proc. of ICLR*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020c. Efficient transformers: A survey.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *Proc. of WMT*.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *Proc. of EMNLP*.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for neural machine translation. In *Proc. of AMTA*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NeurIPS*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proc. of ACL*.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. 2020. Fast transformers with clustered attention. In *Proc. of NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of ICLR*.

Shuohang Wang, Luowei Zhou, Zhe Gan, Yen-Chun Chen, Yuwei Fang, Siqi Sun, Yu Cheng, and Jingjing Liu. 2020a. Cluster-Former: Clustering-based sparse transformer for long-range dependency encoding. In *Findings of ACL*.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020b. Linformer: Self-attention with linear complexity.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *Proc. of ICLR*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL*.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proc. of EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proc. of EMNLP: System Demonstrations*.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *Proc. of ICLR*.

Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. 2020. Lite transformer with long-short range attention. In *Proc. of ICLR*.

Jiyan Yang, Vikas Sindhwani, Haim Avron, and Michael Mahoney. 2014. Quasi-monte carlo feature maps for shift-invariant kernels. In *Proc. of ICML*.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proc. of EMNLP*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proc. of NeurIPS*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proc. of NAACL*.

Dani Yogatama, Yishu Miao, Gabor Melis, Wang Ling, Adhiguna Kuncoro, Chris Dyer, and Phil Blunsom. 2018. Memory architectures in recurrent neural network language models. In *Proc. of ICLR*.

Weiqiu You, Simeng Sun, and Mohit Iyyer. 2020. Hard-coded Gaussian attention for neural machine translation. In *Proc. of ACL*.

Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. 2016. Orthogonal random features. In *Proc. of NeurIPS*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Proc. of NeurIPS*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Proc. of NeurIPS*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proc. of ICML*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. VinVL: Making visual representations matter in vision-language models. In *Proc. of CVPR*.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah A. Smith. 2021. Challenges in automated debiasing for toxic language detection. In *Proc. of EACL*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proc. of ICCV*.

Appendices

Hyperparameter	Value
WMT Machine Translation (All Pairs)	
beam size	5
length penalty	1
CNNDM Summarization	
beam size	4
length penalty	2
max-len-b	140
min-len	55
no-repeat-ngram-size	3
XSUM Summarization	
beam size	6
length penalty	1
max-len-b	60
min-len	10
no-repeat-ngram-size	3

Table 2: Beam decoding hyperparameters. We generally followed prior work: Tang et al. (2021) for machine translation and Lewis et al. (2020) for CNNDM and XSUM summarization.

A. Beam Decoding Hyperparameters

Table 2 shows the beam decoding hyperparameters in our experiments. We generally follow the original settings of the pretrained, off-the-shelf models (Tang et al., 2021; Lewis et al., 2020).

B. Additional Results

Table 3 reports BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020b) scores for the machine translation and summarization experiments, respectively. We use the sacreBLEU implementation for BLEU (Post, 2018b). Note that much recent work (Mathur et al., 2020a; Kasai et al., 2022a,b; Edunov et al., 2020, *inter alia*) found poor correlation between BLEU scores and human judgment for evaluating strong language generation models. COMET is an automatic metric for machine translation that uses crosslingual contextual representations from XLM-RoBERTa (Conneau et al., 2020), but it can be used *monolingually* for evaluating summarization as well (Kasai et al., 2022a).

Fig. 6 explores the performance gains from the patience factor over varying length penalty values. Consistent with the trends from various beam sizes (Fig. 5), the amount of improvement changes, but the patience factor is generally beneficial.

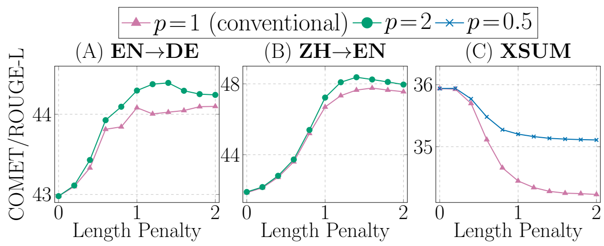

Figure 6: Effects of controlled patience on the dev. data over varying length penalty values. The beam sizes are all 5. We evaluate with COMET for machine translation and ROUGE-L for XSUM summarization.

	WMT 2020 and 2021 Machine Translation (BLEU)								Summarization	
	EN↔DE		**EN↔JA**		**EN↔PL**		**EN↔ZH**		**CNNDM**	**XSUM**
Algorithm	→	←	→	←	→	←	→	←	COMET	COMET
Greedy	42.9	46.6	20.2	17.4	19.8	30.7	31.2	21.7	**1.6**	0.1
Vanilla	**45.1**	48.4	21.6	19.7	**21.1**	**32.5**	**32.5**	23.6	-5.5	-1.6
FCFS	45.0	48.4	21.3	19.5	21.0	32.4	**32.6**	23.4	-4.2	2.2
FCFS w/ p	45.0	**48.5**	**21.7**	**19.8**	**21.1**	**32.5**	32.3	**23.7**	-1.1	**2.5**

Table 3: We evaluate three beam decoding variations, as well as greedy decoding, on machine translation and summarization and report BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020b) scores here. FCFS w/ p indicates our FCSF algorithm with the patience factor ($p = 2$ for machine translation and $p=0.5$ for summarization). COMET is an automatic metric for machine translation that uses crosslingual contextual representations from XLM-RoBERTa (Conneau et al., 2020), but it can be used for evaluating summarization as well (Kasai et al., 2022a).