INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

On the N-gram Approximation of Pre-trained Language Models

Aravind Krishnan[1,2], *Jesujoba O. Alabi*[1,3], *Dietrich Klakow*[1]

[1]Spoken Language Systems Group, Saarland University, Germany
[2]German Research Center for Artificial Intelligence (DFKI)
[3]Saarland Informatics Campus, Germany
{akrishnan|jalabi|dietrich}@lsv.uni-saarland.de

Abstract

Large pre-trained language models (PLMs) have shown remarkable performance across various natural language understanding (NLU) tasks, particularly in low-resource settings. Nevertheless, their potential in Automatic Speech Recognition (ASR) remains largely unexplored. This study investigates the potential usage of PLMs for language modelling in ASR. We compare the application of large-scale text sampling and probability conversion for approximating GPT-2 into an n-gram model. Furthermore, we introduce a vocabulary-restricted decoding method for random sampling, and evaluate the effects of domain difficulty and data size on the usability of generated text. Our findings across eight domain-specific corpora support the use of sampling-based approximation and show that interpolating with a large sampled corpus improves test perplexity over a baseline trigram by 15%. Our vocabulary-restricted decoding method pushes this improvement further by 5% in domain-specific settings.

Index Terms: domain adaptation, approximation, GPT-2

1. Introduction

The advent of Pre-trained Language Models (PLMs) such as BERT [1] and GPT-2 [2] has led to significant strides in language processing tasks, particularly in low-resource scenarios. These models outperform conventional neural language models such as RNNLMs and transformer-based language models (LMs) through the pre-train/fine-tune paradigm. Nevertheless, the potential of these models in automatic speech recognition (ASR) has been insufficiently investigated. Within existing literature, several studies examine their use in lattice rescoring [3, 4, 5], but their utilization in first-pass decoding remains under-explored. This is justified to some extent, since using PLMs directly in first-pass decoding would suffer from the same limitation as using RNNLMs, which is computational inefficiency [6, 7].

Despite the computational bottleneck, the use of RNNLMs in ASR has been studied extensively. In the context of first-pass decoding, two methods using RNNLMs have been explored, namely offline and online methods [8, 9]. The offline method involves approximating an RNNLM to generate an n-gram LM that is used for first-pass decoding and the online approach employs a cache to store the RNNLM states and prunes them during decoding [10]. Several offline approaches to approximate RNNLMs to n-grams have been explored previously [6, 8, 9, 11], but to the best of our knowledge, only the study by [12] looks at a similar approximation for PLMs. Their approach involves pre-training a GPT-like model on a very large general-domain corpus, fine-tuning it on a smaller domain-specific corpus, and then using it to generate a large corpus for approximation into an n-gram LM. They use pre-training corpora that have hundreds of thousands of sentences but approximation has not been explored in a low-resource scenario, where trends can vary significantly in comparison. Moreover, these approximation methods have not been studied for GPT-2, which is much larger and more powerful than GPT and has shown impressive few- and zero-shot capabilities [2] in comparison. Hence, this paper explores the use of GPT-2 in approximating and augmenting n-gram LMs under low-resource scenarios. Specifically, we compare text sampling-based conversion and probability-based conversion as a means of approximating GPT-2 in domain-specific settings. Our research contributions are:

1. Comparing the efficiency of text sampling and probability conversion in approximating GPT-2 into an n-gram model.
2. Introducing a vocabulary-restricted decoding method that improves text generation in domain-specific scenarios.
3. Evaluating the competitiveness of sampling-based approximation across domain difficulty and data scarcity.

2. Data

Taskmaster-2 is a spoken dialogue corpus [13] of 17,289 conversations between users and call-center operators. The data spans human conversations in six domains: restaurant-search, food-ordering, sports, music, movies, and flights. For our experiments, we combine all utterances in a call into a single sample. A train:dev:test split of 70:20:10 is used. **HUB4 1996** [14] is a broadcast news speech corpus that contains 104 hours of transcribed speech from television networks. To simulate a low-resource scenario, we use transcripts from the speech data (LDC97T22) for training and the standard development and test datasets (LDC97S66). **ATCO2** is an air traffic domain corpus [15] that contains air traffic communication between controllers and pilots. We use the 4-hour test set for training and the publicly released 1-hour test set for testing. We choose these datasets to experiment with different domains, task difficulty, and resource availability. The sentence and word level distribution across all datasets is shown in Table 1.

3. Approximating GPT-2

Existing literature supports the use of three offline methods for approximating RNNLMs into n-gram LMs: sampling-based approximation [6], probability-based approximation [11] and iterative approximation [16]. We concentrate on sampling and probability-based approximation methods for GPT-2 and briefly describe these methods in Sections 3.1 and 3.2. We also present a vocabulary-restricted decoding scheme that controls the domain drift during text generation in Section 3.3.

10.21437/Interspeech.2023-2182

Dataset	#Sentences / #Words		
	Train	Dev	Test
ATCO2	2337 / 27K	537 / 6K	826 / 10k
HUB4	46573 / 735K	4178 / 65K	346 / 19K
Movies	2139 / 344K	612 / 97K	305 / 49K
Restaurant-Search	2293 / 351k	656 / 98k	327 / 49k
Music	1122 / 142K	321 / 40K	160 / 19K
Flights	1736 / 332K	497 / 96K	248 / 47K
Sports	2436 / 279K	697 / 81K	348 / 39K
Food-Ordering	735 / 80K	210 / 22K	105 / 11K

Table 1: *Dataset distribution and splits.*

3.1. Sampling-Based Approximation

Given a background corpus B, sampling-based approximation (SBA) involves generating additional text samples from the same distribution as B using a language model such as an RNNLM or Transformer-based LM. The generated text is then used to train a traditional n-gram LM, which approximates the probability distribution of the neural model itself [6]. The performance of this approach is often improved by interpolating the approximated n-gram LM with the baseline n-gram LM [8, 11] trained on B. This approach has been studied with RNNLMs [6, 8, 11] and Transformer-based LMs [12] but has not been explored with massively pre-trained PLMs. In this work, we study how this approximation performs with GPT-2 across datasets, data sizes, and domains.

3.2. Probability-Based Approximation

We also explore the use of probability-based approximation (PBA) for GPT-2, which was initially proposed by [11] as a way to approximate RNNLMs into n-gram LMs. Instead of using count-based probabilities, this method involves extracting and assigning n-gram probabilities directly from trained RNNLMs. Since GPT-2 uses subword (BPE) tokens, we obtain word-level probabilities by multiplying the conditional probabilities of each BPE token in the target word. For a word W composed of BPEs $\{w_1, w_2 \dots w_n\}$, the word level probability is computed as:

$$p(W|H) = \prod_{i=1}^{n} p_{gpt2}(w_i|w_{i-1} \dots w_1 H) \quad (1)$$

Where H is the sentence history and p_{gpt2} is the conditional probability assigned to a BPE token by the fine-tuned GPT-2. The conditional word probabilities are reassigned to n-gram probabilities and averaged according to the original setup. Backing off is done from history sums taken from a trigram LM developed from the training corpus. The authors in [11] observe that the resulting model produces the best perplexities when unigram probabilities are borrowed from the baseline trigram LM.

3.3. Vocabulary-Restricted Decoding

In addition, we propose a vocabulary-constrained extension to SBA in the PLM setup. In this approach, we restrict the pool of BPE tokens that can be generated to the BPE vocabulary of the training set. We start by creating a BPE vocabulary over all words in the training data. Each word is tokenized [1] and all composite BPEs are added to the BPE vocabulary. During random sampling, softmax operations are performed only over the BPEs in our vocabulary. The sampling algorithm is thus restricted to picking from the BPEs present in this vocabulary. We call this Vocabulary-Restricted Decoding.

We distinguish our decoding scheme from lexically constrained decoding [17, 18], where restrictions are placed on *including* specific tokens during generation. Our approach uses the training set vocabulary exclusively during decoding, thus *excluding* non-relevant tokens. We expect that in domain-specific scenarios, constraining the generation vocabulary will suppress tokens and force GPT-2 to model the words in the training set better.

4. Experiments

4.1. Experimental Setup

We use the 124M parameter version of GPT-2 from $huggingface$ [19] for all experiments. The model is fine-tuned with a learning rate of 5e-5 and a weight decay of 0.01. The learning rate is linearly warmed up using $min(\#train\ samples, 100)$ steps. Training is terminated using early stopping on the development set with patience of 5. Text generation uses top-p [20] sampling with p=0.95 and a temperature of 1.0. Unless specified otherwise, we generate 100 times the training data for all SBA experiments. The n-gram models are trigram models with Kneser-Ney smoothing [21], and are built and evaluated with the SRILM [22] Toolkit. **KN3** denotes the baseline model trained on just the training corpora; **RS-KN3** is trained on the random-sampled corpora and **VR-KN3** on corpora sampled using vocabulary-restricted decoding. Note that RS-KN3 and VR-KN3 are both SBA approaches with different decoding schema. All experiments use a vocabulary that combines all words in the training data and the 100x corpora generated for RS-KN3 and VR-KN3[2]. This results in an OOV rate of less than 2% for each dataset.

4.2. Comparing Approximation methods

We compare the test set perplexities of the approximation approaches discussed with the "true" perplexity of a fine-tuned GPT-2 and present the results in Table 3. Word level perplexities are obtained from GPT-2 using the computation described in [23] and elaborated in [24]. To simplify, we add the test vocabulary to the n-gram models, thus having zero OOV test rate. Except for *sports* and *food-ordering*, both approximation methods incur at-least 2x degradation when compared to the true perplexity but PBA incurs significantly higher deterioration in comparison. This is consistent with previously reported results using PBA [8, 11], but more drastic with GPT-2. This could be explained by PLMs' tendency to overestimate n-gram probabilities in comparison to RNNLMs. A closer look confirms that GPT-2 overestimates lower-order n-gram probabilities during PBA because of probability sharing between higher and lower-order n-grams. It is to be noted that the vocabulary set for GPT-2 is BPE based and is more fine-grained than the word-level model. We can therefore only assume that our computation for the non-approximated GPT-2 perplexity underestimates the true metric. Nevertheless, as the underestimated metric outperforms the word-based model, the granularity of the vocabulary does not interfere with our finding. We conclude that PBA is a lossy conversion scheme for GPT-2 approximation and focus on the more performant SBA method for the rest of this paper.

[1] GPT-2 tokenizes words with and without preceding blanks ("$apple$" and "$\_apple$") differently. We add both versions into the vocabulary

[2] Except in Section 4.2 where the test vocabulary is additionally introduced

Model	ATCO2	HUB4	Taskmaster					
			Movies	Restaurant	Music	Flights	Sports	Food
KN3	**31.95** / 428	**275.19** / 517	**22.31** / 1210	23.01 / 1404	34.41 / 1269	19.50 / 556	13.24 / 639	12.62 / 296
RS-KN3	44.00 / 446	310.19 / 469	25.67 / 1004	25.09 / 1241	31.11 / 639	17.59 / 397	12.77 / 605	9.96 / 119
VR-KN3	35.53 / 445	342.72 / 452	23.54 / 980	**23.34** / 1066	**28.41** / 969	**16.47** / 404	**12.35** / 593	**9.91** / 260
KN3 + RS-KN3	29.27	**207.16**	19.58	20.26	26.03	15.96	11.45	**9.33**
KN3 + VR-KN3	**28.60**	238.21	**19.12**	**19.71**	**25.47**	**15.70**	**11.38**	9.53
RS-KN3 + VR-KN3	35.21	241.74	25.92	22.96	26.64	16.34	12.19	9.49
Total Interpolation	28.54 / 415	207.32 / 337	18.91 / 795	19.54 / 955	24.38 / 440	15.53 / 249	11.29 / 583	9.10 / 98

Table 2: *Perplexities and OOV rates for the n-gram models constructed from sampled corpora. KN3 denotes the trigram LM baseline. Metrics are of the format: $Perplexity/\#OOVs$. All perplexities are computed using the vocabulary obtained from combining all three corpora, whose OOV rate is indicated in blue.*

Dataset	GPT-2	RS-KN3	PBA
ATCO2	25.04	47.04	80.04
HUB4	189.91	369.90	632.92
Movies	13.53	27.92	49.89
Restaurant-Search	13.26	28.39	46.60
Music	11.55	35.23	57.83
Flights	7.23	17.83	18.10
Sports	12.77	13.46	30.74
Food-Ordering	9.96	10.37	23.31

Table 3: *Sampling-based approximation of GPT-2 into an n-gram model consistently outperforms probability-based approximation*

4.3. Vocabulary Restriction and Interpolation

Table 2 compares the n-gram models developed from the training corpora and the corpora generated from random sampling and vocabulary-restricted decoding. Except for ATCO2, both sampling methods produce an OOV reduction on the test set. Combining vocabularies from the original corpus and the sampled corpora reduces the OOV count significantly across all datasets. A closer look at the OOVs contributed shows that vocabulary-restricted decoding caters to affixing and derivational inflections while random sampling has additional coverage over novel nouns and verbs.

In relative terms, we see that RS-KN3 under-performs the baseline considerably (+40% ppl) with ATCO2, where the unconventional grammatical structure adversely affects sampling. On the other hand, VR-KN3 performs strongly here with 20% improvement over KN3. This suggests that controlling the drift of random sampling is beneficial in tightly constrained domains. In general, we see that vocabulary restriction improves over random sampling when the domain (and consequently the vocabulary) is restricted. However, performance suffers with HUB4, where the domain is wide-ranging and the vocabulary cannot be restricted as dramatically. Vocabulary restriction also outperforms random sampling for all datasets in Taskmaster, where a satisfactory equilibrium exists between grammatical complexity and domain restriction.

In all cases, an improvement over the baseline is observed as soon as either of the sampling methods is interpolated with the baseline KN3 (interpolation weights are tuned with the respective development sets). Here again, domain specificity plays its part, and interpolation with VR-KN3 outperforms RS-KN3 in the case of ATCO2 while the latter performs strongly for HUB4.

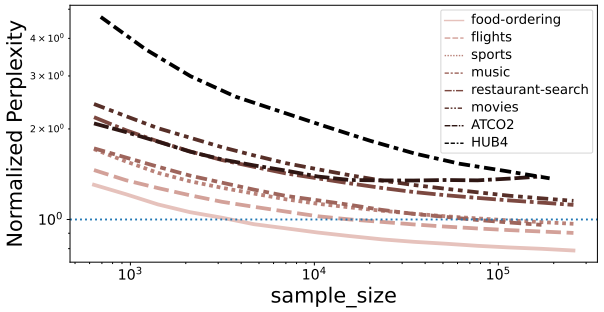


Figure 1: *Benefits of sampling depends on domain difficulty. Perplexities are normalized across the respective KN3, which is shown as the horizontal line at 1.0. Darker colour denotes higher perplexity on the dev-set.*

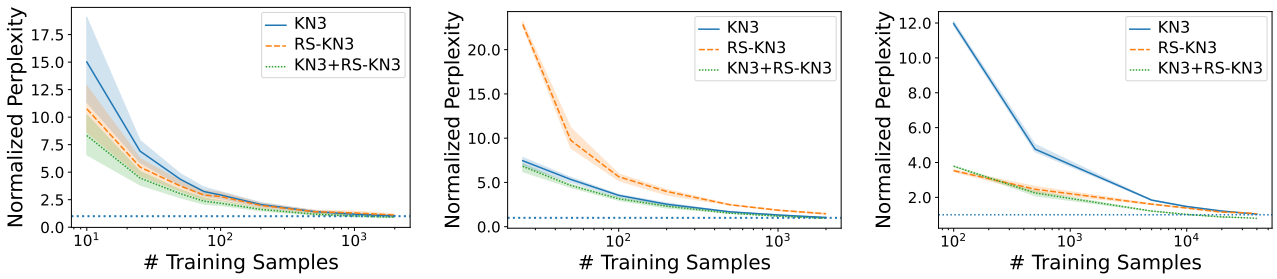
Interpolation with either sampled model shows promising improvement with Taskmaster. We observe the best performance across all datasets when all three models are interpolated together. On average, interpolating the baseline with RS-KN3 gives a 15% reduction in test perplexity and additional interpolation with VR-KN3 decreases perplexity further by 5%. We conclude that interpolation with a large-scale sampled corpus is almost always beneficial for a trigram LM and is a simple and effective method to exploit PLMs for first-pass decoding.

5. Discussion

We devote the discussion section to answering the following question - "Can the SBA LMs outperform the baseline KN3 without interpolation?". We isolate and study two variables that influence augmentation: the domain of the dataset and the number of training samples available.

5.1. Domain difficulty

Comparing language modeling difficulty across two domains is challenging in a statistical setup owing to differences in vocabulary and the lack of standardized background corpora that help isolate perplexity comparisons to domains. Sub-word modeling and pre-training mitigate these concerns and PLMs prove to be ideal sandboxes to compare difficulties in modeling different domains. Assuming that pre-training acquaints PLMs with everyday words, the adaptability of a domain over fine-tuning can then approximate the difficulty in modeling it. We quantify it by



(a) *Taskmaster* (b) *ATCO2* (c) *HUB4*

Figure 2: *Performance of language models across sub-sampled training corpora. Perplexities are normalized with the full-scale KN3 perplexity (which is shown as the blue dotted line at 1.0). The Taskmaster datasets are averaged to generate one plot.*

computing the dev set perplexities of fine-tuned[3] GPT-2 models and compare them across domains to rank their difficulty. Figure 1 shows the trend in test perplexity with an increase in the number of sampled sentences for RS-KN3. The figure shows that HUB4 and ATCO2 are both difficult for GPT-2 to model, but HUB4's steeper slope suggests a preference for additional sampling when domains are spread thin. In most cases, we see that an increase in the number of samples generated correlates with a decrease in test perplexity. Larger sampling has a slower effect on perplexity after a certain point and produces long tails afterward. For datasets that have a lower dev set perplexity, SBA is seen to beat the baseline KN3 early on. An increase in domain difficulty stretches the cross-over point and we see that the number of sentences required to outmatch the baseline is proportional to the difficulty of the domain. We observe similar trends for VR-KN3 models as well, with a slight downward shift along the y-axis for all curves (except for HUB4, where the shift is slightly upwards). This analysis provides us with two conclusions: 1) Generating additional data correlates with reductions in test perplexity, but 2) The amount of generated data required to match the baseline trigram model has an inverse relation with domain difficulty.

5.2. Few-Shot Approximation

This section compares the responses of SBA and the baseline KN3 under low-resource settings. We simulate few-shot scenarios by sub-sampling the train and development corpora while keeping the test corpus unchanged. For each setting, GPT-2 is fine-tuned on the sub-sampled training set, using the sub-sampled development set for early stopping. The fine-tuned model then generates 100x the size of the sub-sampled train set. The resulting perplexities are normalized with the full-scale KN3 value for each dataset (refer to Table 2). Each experiment is run across three random seeds and the results are plotted in Figure 2.

The results for ATCO2 reflect the domain difficulty yet again, with GPT-2 struggling to understand the domain in few-shot settings and the trigram model performing significantly better. GPT-2 does not outperform the trigram corpus in any data scenario, and the interpolated model hugs the KN3 curve. For Taskmaster and HUB4 on the other hand, the generated corpus fares significantly better in few-shot scenarios. The perplexity of the generated corpus is several times lower than the sub-sampled KN3 model when the number of samples available is less than 100. This can be attributed to large-scale pre-training, which helps the PLM generalize text quicker than a KN3 under low resource settings. However, the KN3 curve has a steeper decline, which suggests that the it benefits more from the availability of additional curated corpora than a PLM in few-shot scenarios. We also see that the improvements produced with the additional 100x data become decreasingly pronounced with increasing train data size. i.e., once the training data is large enough to generalize the domain well, the baseline KN3 becomes a tough competitor. For HUB4, the sub-sampled KN3 even outperforms the generated corpus for train sets larger than 20K samples. Improvements can be made by generating exponentially many samples even at "high" resource settings, but the cost-benefit ratio is significantly more encouraging in few-shot settings. This is in contrast to [9], where larger training data resulted in larger gains for word-based RNNLM models. We conclude that in domain-specific settings, although PLMs demonstrate impressive generational capabilities under few-shot conditions, n-gram models are tough baselines in comparatively higher-resourced scenarios.

[3]Perplexity from a pre-trained model could be misleading since the domain might look hard initially but be easily learnable.

6. Conclusion

In this work, we compare two approaches that approximate GPT-2 into an n-gram model and find that large-scale sampling after fine-tuning performs significantly better than a probability-based approximation. We then introduce a vocabulary-restricted decoding schema that proves helpful in low-resource domain-specific scenarios. We show that interpolating the baseline trigram LM from a domain-specific corpus with trigrams made from large-scale generation improves text perplexity by 20% across eight domains. Using dev set perplexity to compare domains, we show that the sampling volume required to match the baseline trigram perplexity is inversely proportional to the domain modeling difficulty. We also extend text-based augmentation to few-shot scenarios, where we see impressive performance with GPT-2. However, we find that the cost benefit ratio declines rapidly with additional training data and that the trigram model is a tough baseline at higher data sizes.

7. Acknowledgements

We are grateful for the feedback from Vagrant Gautam and the anonymous reviewers. Aravind Krishnan was funded by the ViCKI project within the framework of the German Civil Aviation Research Programme (LuFo VI-1) of the Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant number FKZ 20D1910D. Jesujoba Alabi was funded by the BMBF project SLIK under the Federal Ministry of Education and Research grant 01IS22015C.

8. References

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

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[3] S. Dingliwal, A. Shenoy, S. Bodapati, A. Gandhe, R. T. Gadde, and K. Kirchhoff, "Domain prompts: Towards memory and compute efficient domain adaptation of asr systems," *arXiv preprint arXiv:2112.08718*, 2021.

[4] T. Udagawa, M. Suzuki, G. Kurata, N. Itoh, and G. Saon, "Effect and analysis of large-scale language model rescoring on competitive asr systems," *arXiv preprint arXiv:2204.00212*, 2022.

[5] X. Zheng, C. Zhang, and P. C. Woodland, "Adapting gpt, gpt-2 and bert language models for speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 162–168.

[6] A. Deoras, T. Mikolov, S. Kombrink, M. Karafiát, and S. Khudanpur, "Variational approximation of long-span language models for lvcsr," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5532–5535.

[7] H. Schwenk and J.-L. Gauvain, "Connectionist language modeling for large vocabulary continuous speech recognition," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2002, pp. I–765–I–768.

[8] M. Singh, Y. Oualil, and D. Klakow, "Approximated and domain-adapted lstm language models for first-pass decoding in speech recognition." in *INTERSPEECH*, 2017, pp. 2720–2724.

[9] B. Tarján, G. Szaszák, T. Fegyó, and P. Mihajlik, "On the effectiveness of neural text generation based data augmentation for recognition of morphologically rich speech," in *Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8–11, 2020, Proceedings 23*. Springer, 2020, pp. 437–445.

[10] K. Li, H. Xu, Y. Wang, D. Povey, and S. Khudanpur, "Recurrent neural network language model adaptation for conversational speech recognition." in *Interspeech*, vol. 2018, 2018, pp. 3373–3377.

[11] H. Adel, K. Kirchhoff, N. T. Vu, D. Telaar, and T. Schultz, "Comparing approaches to convert recurrent neural networks into backoff language models for efficient decoding," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[12] Y. Wang, H. Huang, Z. Liu, Y. Pang, Y. Wang, C. Zhai, and F. Peng, "Improving n-gram language models with pre-trained deep transformer," *CoRR*, vol. abs/1911.10235, 2019.

[13] B. Byrne, K. Krishnamoorthi, C. Sankar, A. Neelakantan, D. Duckworth, S. Yavuz, B. Goodrich, A. Dubey, A. Cedilnik, and K.-Y. Kim, "Taskmaster-1: Toward a realistic and diverse dialog dataset," *arXiv preprint arXiv:1909.05358*, 2019.

[14] D. Graff, Z. Wu, R. MacIntyre, and M. Liberman, "The 1996 broadcast news speech and language-model corpus," in *Proceedings of the DARPA Workshop on Spoken Language technology*. Citeseer, 1997, pp. 11–14.

[15] J. Zuluaga-Gomez, K. Veselỳ, I. Szöke, P. Motlicek, M. Kocour, M. Rigault, K. Choukri, A. Prasad, S. S. Sarfjoo, I. Nigmatulina *et al.*, "Atco2 corpus: A large-scale dataset for research on automatic speech recognition and natural language understanding of air traffic control communications," *arXiv preprint arXiv:2211.04054*, 2022.

[16] E. Arısoy, S. F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8242–8246.

[17] C. Hokamp and Q. Liu, "Lexically constrained decoding for sequence generation using grid beam search," *arXiv preprint arXiv:1704.07138*, 2017.

[18] J. E. Hu, H. Khayrallah, R. Culkin, P. Xia, T. Chen, M. Post, and B. Van Durme, "Improved lexically constrained decoding for translation and monolingual rewriting," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 839–850.

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6

[20] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[21] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.

[22] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh international conference on spoken language processing*, 2002.

[23] S. J. Mielke, "Can you compare perplexity across different segmentations?" Apr 2019. [Online]. Available: https://sjmielke.com/comparing-perplexities.htm

[24] R. Cotterell, S. J. Mielke, J. Eisner, and B. Roark, "Are all languages equally hard to language-model?" in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 536–541. [Online]. Available: https://aclanthology.org/N18-2085