Asking the Right Questions: Question Paraphrasing Using Cross-Domain Abstractive Summarization and Backtranslation

Student 1
****@utexas.edu

Student 2
*****@utexas.edu

Abstract

A common issue when asking questions is that they might be prone to misinterpretation: most of us have experienced when a colleague or teacher misinterprets a question and provides an answer which is tangential to the information we desire, or incomplete. This problem is exacerbated over text, where visual and emotion cues are not transmittable. We hypothesize that question answering models face the same issues as the human responder in such situations: when asked an ambiguous question, they might be unsure what to retrieve from the given passage. We propose paraphrasing the question with pre-trained language models, to improve answer retrieval and robustness to ambiguous questions. We introduce a new scoring metric, GROK, to evaluate and select good paraphrases. We show that this metric improved upon paraphrase selection via beam search for downstream tasks, and that this metric combined with data augmentation via backtranslation increases question answering performance on the NewsQA and BioASQ datasets, improving EM by 2.5% and F1 by 1.9% over-and-above the baseline on the latter.

1 Introduction

Data augmentation has long been a popular method in computer vision (CV) to enhance datasets and improve generalization. In CV, there are many possible adjustments that can be made to input data without fundamentally changing the identity of the data. Natural language, unfortunately, does not share this property. In natural language, even small changes in a phrase can completely change the semantics. This makes sophisticated, robust data augmentation a challenge. Though challenging, the same benefits that have driven augmentation research in CV remain powerful motivators for creating robust augmentation systems for Natural Language Processing (NLP).

With these benefits in mind, we have explored methods for data augmentation for the question answering NLP task. Here, data augmentation can be applied most naturally to the questions being asked. Just as humans often request that a question be repeated or rephrased so that they can understand and respond appropriately, so too can neural models benefit from such "clarifications", or paraphrases, of the question. To generate these paraphrases in a semantic consistency manner, we leverage *other* NLP models, pre-trained on large datasets to generate paraphrases of the questions in the NewsQA (Trischler et al., 2016) and BioASQ datasets (Tsatsaronis et al., 2015). We use models trained for both abstractive summarization and machine translation to act as our paraphrasing engines. All of these paraphrasing engines have been trained on other datasets, so their application to the NewsQA and BioASQ data is cross-domain. The summarization model was set to output strings of similar length to the input, thus becoming a paraphraser rather than a true summarizer. Two paired machine translation models, one from the input language to a target language, and one from that target language back to the input language, were used to "backtranslate", and thus paraphrase, an input sequence. Both paraphrasing techniques were implemented prior to training - there is no "on the fly" augmentation.

In addition to augmentation, we also experimented with data substitution. For substitution, instead of creating a larger dataset with both the original and paraphrased training sample, we replaced the original training samples with their paraphrases. This tests the potential improvements due to different, often better, question phrasing.

Our observations are that data augmentation (concatenation with existing dataset) improved performance on BioASQ and NewsQA significantly, and that, surprisingly, even substitution occasion-

ally improved performance when paraphrases were selected via our new GROK metric.

2 Overview of Previous Work

Multiple methods of data augmentation have been explored for text data, ranging from simple synonym substitution to substitution with generative models. These methods have generally been successful.

Some simple text augmentation techniques have produced strong results. Zhang et al. 2015 introduced thesaurus based synonym replacement. Similarly, Wang and Yang 2015, Kobayashi 2018a, and others experimented with word replacement based on similarity in various embedding spaces using metrics like nearest neighbor or cosine similarity.

As pretrained generative models become pervasive in NLP, they were also applied to the text augmentation problem (Kobayashi, 2018b). These models use their "knowledge" of the language to generate realistic and semantically/syntactically sound insertions for masked words. The final evolution of this concept is to generate entire sequences that mimic the input sequence, as in Kafle et al. 2017. We follow this pathway to generate additional whole paraphrases, and then we add our own scoring methods to ensure robustness of additional data.

3 Data Augmentation Techniques

Each sample is comprised of a passage, answer start and end tokens, and a question. Following our hypothesis that rephrasing the question is analogous to clarifying in human conversation, we will be evaluating models trained on datasets with paraphrased questions.

We will be tested question paraphrases using a variety of techniques. First, we divide our paraphrases by method: Abstract Summarization or Machine Translation. In addition, we will be evaluating whether a custom scoring metric, GROK (Section 3.3), which evaluates the quality of paraphrases, actually helps performance. Finally, we can test whether the beam size during paraphrasing can affect the quality of paraphrases, and thus the resulting performance of the models.

3.1 Abstractive Summarization

Abstractive summarization is the task of generating concise, semantically consistent paraphrases from a large input context. The goal is to shorten an input, but maintain the meaning. We chose pretrained neural abstractive summarization models over other summarization models because they are least likely to directly copy the input sequence, and thus produce a more novel summarization. This is critical to our paraphrasing task, where novelty is important to actually augment the dataset. To turn a summarizer into a paraphraser, we ensured that the output sequence length was within a certain threshold of the input sequence.

Popular state of the art summarizers are BART (Lewis et al., 2019) and PEGASUS (Zhang et al., 2019). BART trains a transformer-based encoder using masked language modelling similar to BERT, but then uses a de-noising autoencoder to perform the unmasking. We can call this BERT extended to language generation tasks. PEGASUS, on the other hand, is trained on a dual objective: the PEGASUS encoder works similar to a masked-language-model as in BART and BERT, but the PEGASUS decoder uses a new Gap Sentence Generation objective, where it must predict entire intermediate sentences which have been masked out. This means that it must learn from very little context and receives training feedback from an entire predicted gap sentence, unlike BERT and BART which only learns from the loss of masked tokens.

Since it is currently the state of the art for abstractive summarization, we choose PEGASUS, and use the implementation from Huggingface (Wolf et al., 2020). We applied the pretrained model directly from the work of Zhang et al. 2019 (Large), as well as variants that have been finetuned on the Extreme Summarization (XSUM) dataset (Narayan et al., 2018), CNN-Daily Mail dataset (CNN/DM) (See et al. 2017 and Hermann et al. 2015), and Multi-News (MN) dataset (Fabbri et al., 2019). PEGASUS-CNN/DM is finetuned on a similar data source as NewsQA, although for the summarization task instead of question-answering task. We thus expect this to be closest to the performance of an "in-domain" model.

3.2 Neural Machine Backtranslation

Backtranslation is a form of paraphrasing that takes a phrase in one language, translates it to another, and then translates it back to the original language. Backtranslation is commonly used to generate synthetic data for NLP tasks because it has been trained on a large corpus of data and has similar properties to language models in that it will frequently

generate syntactically correct phrases. When translating to and from another language, a phrase often undergoes synonym replacement and token reorganization, but in a syntactically sound way. We use sampling to generate more varied questions than beam search (Edunov et al., 2018).Thus, we can produce valid questions that have surface level differences but are fundamentally similar semantically to the original phrase. This is a less manual way of augmentation that encompasses many of the common text augmentation techniques discussed in 2.

We use paired pretrained models for English $\leftrightarrow$ German from Facebook's winning entry to the WMT19 News Translation competition, documented in (Ng et al., 2019) and implemented in pytorch in Facebook's fairseq library (Ott et al., 2019).

3.3 GROK Scoring Function

We propose a novel scoring function, **G**ROK **R**estricts **O**utrageous **K**eywords as a heuristic to select the best paraphrases. We define a good paraphrase to have certain properties: rewording (using different articles), few phrases copied directly from the input, few repeating ngrams within the paraphrase, and fewer unrelated "hallucinations". The GROK score incorporates all of these in its components:

- Length of longest non-contiguous match (LNCM, lower = better)
- Number of unique n-grams in the paraphrase (UN, higher = better)
- Hallucinated unigrams, not including stopwords and punctuation (HU, lower=better)
- Missing unigrams, not including stopwords and punctuation (MU, lower = better)

A non-contiguous matching token sequence between an original question, q_0 and it's paraphrased version, q_p is a sequence of shared tokens that appear in the same order in both q_0 and q_p, but are not necessarily contiguous. For example, if q_0 = "I am the original question!" and q_p = "the original, true question is me", then the longest non-contiguous matching token sequence is "the", "original", "question"], since those tokens appear in the same left to right order, even though they are non-contiguous. The intuition behind this aspect of GROK is to punish exact or near-exact copies with only one-word differences, since these are often not true paraphrases. The closer to an exact copy, or the more tokens that are exactly copied in order from q_0, higher the LNCM count.

The second component of GROK is the number of unique n-grams, UN. Most human-generated sentences and summaries vary with the exact phrases used, even while keeping semantics the same, which reflects in a high unique n-gram count. This also acts as a filter for a common issue with generative models, which is to repeat short text phrases continually.

The final two components are complementary: HU measures the number of new unigrams which the model generates which are not in the original, whereas MU captures the number of unigrams from the original which the model does not generate. The former catches "hallucinations": random insertions of information which are not in the original content. The latter catches omissions: important pieces of information which the model missed while generating. We filter out stopwords while calculating both, to avoid penalizing the model when it paraphrases using different prepositions, articles and punctuation (which is desirable in a paraphrase!).

Altogether, these components are combined in a simple function:

$$\text{GROK}(ngram, thresh) = \frac{\text{UN}}{\text{LNCM + HU + MU}}$$

We observe that when using $ngram = 1$, an exact copy of the original has a GROK score of 1.0; this captures the fact that although the paraphrase does not help, it does not hurt either. A generated sentence which is an exact copy of the original, but with half the unique tokens missing, will be assigned a GROK score of 0.5, as will a sentence with half the unique tokens replaced by a random selection. One caveat is that GROK does not penalize sentence-swapping (e.g. a high GROK score would be given to the paraphrase: "WE MUST IMPOSE SANCTIONS", SAID OBAMA TO US CONGRESS LAST TUESDAY."IT IS OUR RIGHT AS A NATION." $\longrightarrow$ "IT IS OUR RIGHT AS A NATION." SAID OBAMA ON TUESDAY TO CONGRESS. "WE MUST IMPOSE SANCTIONS").

Empirically, we have found that GROK scores above 1.2 are good paraphrases.

3.4 Beam Sizes

Choosing beam size required balancing model performance and computational efficiency. Ideally, a large number of beams will imrpove the paraphrases, but at high memory and computation cost. We originally chose a beam size of 75 for PEGASUS on BioASQ, but found this to be prohibitively slow on NewsQA. Thus, we dropped beam size to 30 for NewsQA. Though this does slightly reduce performance, it is a necessary trade off for the model to run in a reasonable time.

Our reasoning for experimenting with large beam sizes to begin with is to showcase our GROK function, and to compare it's performance over candidate paraphrases with the top beam score candidate. A larger number of beams gives our GROK function more inputs to select from.

4 Experiments

Since we are evaluating data augmentations, our model and training methods were held constant through the experiments. While this likely did not allow us to maximize performance on each specific augmented dataset, it does allow us to attribute performance differences to the dataset itself and make "apples to apples" comparisons. Below we detail the model and training schema.

4.1 Model

We employ a bidirectional LSTM with aligned attention and question encoding based on the reader from (Chen et al., 2017) as implemented by Greg Durret, et al. We use 300 dimensional pretrained GloVe word embeddings. (Pennington et al., 2014).

4.2 Training Method

Each model was trained for approximately 150 gradient updates on GPU. Full hyperparameters are listed in Table. All training hyperparameters, except epoch number, remained constant to ensure fair performance comparisons between augmented datasets.

In order to ensure a fair comparison between datasets of different sizes, we implemented an adaptive epoch count that equated the number of gradient updates per model. For example, a model being trained on twice the amount of data in the base dataset will be trained for half as many epochs as the base dataset, giving the model the same number of opportunities to learn (udpate parameters) in each situation. Even with this adaptive technique, we still used early stopping to prevent over fitting, so in some cases gradient updates will not be equal and the number cited above is an upper bound.

5 Results

We report results for question augmentation and question substitution on the BioASQ and NewsQA datasets in Tables 1 and 2. We report the three-run averaged exact match (EM) and F1 scores on the question answering task associated with each dataset. All PEGASUS variants use a beam size of 75 (for BioASQ) and 30 (for NewsQA) and temperature of 1.5. All FairSeq variants use a beam size of 20 and temperature of 1.5.

For abstractive summarization models, the max length multiplier is the truncation length of paraphrases, in relation to the original phrase. For example, if the original sentence is 20 words long, a maximum length paraphrase of 1.5 restricts the max paraphrase to 30 words. All paraphrase candidates generated by the model were scored and the top paraphrase was selected according to beam search score, our own custom GROK metric. When using GROK, if the highest-scoring candidate does not meet the score threshold, we use the original paraphrase instead. The percentage of paraphrases that meet this selection criteria is reported as Paraphrase %. Note that not all paraphrases are unique from the original question.

5.1 Efficacy of GROK

Using Table 1, we compare model performance when trained on data selected by beam search against our GROK metric. In all cases, our GROK metric outperforms beam search on the downstream question-answering task, even with as many as 75 beams. This suggests that the paraphrases chosen by beam search are linguistically suboptimal, and that the intuition behind GROK is sound - it chooses good paraphrases that lead to more robust model learning. In the substitution task, we replace human written questions with machine paraphrases. In this case, we do not expect that our paraphrases would improve over human questions. But we see that in one case, we actually improve performance on the downstream task when we substitute high quality paraphrases into the dataset to replace lower quality human written questions.

We also note that the stricter GROK is in selecting paraphrases (lower paraphrase %), the better model performance is. We suspect that this is be-

Paraphrasing Engine	Max Length Multiplier	Candidate Selection	BioASQ			NewsQA		
			Paraphr. %	EM	F1	Paraphr. %	EM	F1
Baseline (no paraphrasing)	-	-	0	67.6	73.3	0	30.5	45.1
PEGASUS CNN/DM	1.5	Beam	92.9	62.2	69.3	15.4	30.5	45.0
	1.5	GROK(1, 1.5)	36.1	66.3	72.5	15.4	30.1	44.6
	1.5	GROK(1, 1.2)	74.6	63.1	69	85.2	28.7	42.2
	1.5	GROK(2, 1.2)	55.7	66.0	71.1	15.4	30.1	44.7
	2.0	GROK(1, 1.2)	90.5	63.5	69.9	-	-	-
PEGASUS Large	1.5	Beam	64.4	62.1	69.2	15.4	30.3	44.9
	1.5	GROK(1, 1.5)	4.2	66.4	72.4	7.8	30.2	44.5
	1.5	GROK(1, 1.2)	21.7	65.7	72.2	15.4	30.3	44.8
	1.5	GROK(2, 1.2)	**6.4**	**68.0**	**73.7**	11.8	30.0	44.4
	2.0	GROK(1, 1.2)	57.1	62.3	69.4	-	-	-
PEGASUS Multinews	1.5	Beam	100	31.3	40.5	15.4	29.8	44.4
	1.5	GROK(1, 1.5)	5.2	66.2	72.2	4.9	30.1	44.2
	1.5	GROK(1, 1.2)	22.9	66.2	72.2	15.4	30.3	44.8
	1.5	GROK(2, 1.2)	10.1	65.5	72.1	9.7	30.3	44.8
	2.0	GROK(1, 1.2)	69.3	57.4	65.5	-	-	-
PEGASUS XSUM	1.5	Beam	97.4	35.7	45.6	15.4	30.1	44.4
	1.5	GROK(1, 1.5)	15.5	67.3	72.8	15.4	29.9	44.2
	1.5	GROK(1, 1.2)	53.7	60.1	67.7	53.3	28.1	41.5
	1.5	GROK(2, 1.2)	42.6	61.9	68.8	15.4	30.2	44.7
	2.0	GROK(1, 1.2)	80.4	50.9	61.1	-	-	-
Fairseq	1.5	Sampling	100	62	69.5	15.4	30.3	44.8
	1.5	Sampling + GROK(1, 1.5)	98.7	61.3	68.9	15.4	30.2	44.4
	1.5	Sampling + GROK(1, 1.2)	98.0	63.2	69.6	15.4	30.6	44.7
	1.5	Sampling + GROK(2, 1.2)	80.8	63.6	70.2	15.5	30.1	44.3

Table 1: Substitution. See Section 5 for details. Performance is average of 3 runs.

cause these human written questions are, in general, good, so GROK only induces improvements when it "fixes" a few poorly written or ambiguous questions. In this sense, GROK acts as a way to de-noise the dataset when used for substitution.

6 Conclusions

In this work, we observe the effect of machine-generated text for data augmentation in the question-answer setup, were we perform either a substitution of the question with a machine generated one, or add it to the original dataset. We use SOTA models for Abstractive Summarization (PEGASUS) and Neural Machine Translation (FairSeq) as paraphrasers. We observe that choosing the appropriate paraphrase among a list of candidates adds complexity to the problem, as naive approaches such as picking the best beam-search value do not generate good paraphrases for our downstream task. We thus propose a new metric, GROK, which we use for both candidate selection and post-hoc filtering of "good" paraphrases. When using GROK for substitution, we see that we are even able to improve upon the baseline. We offer two hypothesis for these performance increases: first, the paraphraser "smooths" the questions from a somewhat noisy, potentially ungrammatical state (as a result of crowdsourced or truncated input) and rephrases the questions in a more linguistically sound form. The second, and perhaps more intriguing explanation is that the paraphrasers generate questions that are more easily understood, natively, for the underlying QA model. In this sense, the paraphrasers act almost as translators between "human English" and "machine English". This is vaguely analogous to how humans may speak their non-native language. They may use the proper vocabulary, and often form syntactically sound phrases, but they do so in forms that a native speaker would recognize as unusual because they do not follow local connotations. Finally, we observe that using our GROK score to score and filter paraphrases when used in augmentation gives an overall performance improvement of 2.5% EM and 1.9% F1 on the BioASQ task. We believe this is an interesting area of research and further work is needed.

7 Future Work

Just as a respondent can benefit from the question being rephrased, so too might the question asker benefit from the answer being rephrased. Thus, in

Paraphrasing Engine	Max Length Multiplier	Candidate Selection	BioASQ		
			Paraphr. %	EM	F1
Baseline (no paraphrasing)	-	-	0	67.6	73.3
PEGASUS CNN/DM	1.5	Beam	**50**	**69.0**	**74.3**
	1.5	GROK(1, 1.2)	39.7	62.5	70.1
	1.5	GROK(2, 1.2)	32.8	65.5	71.5
PEGASUS Large	1.5	Beam	50	65.4	72.4
	1.5	GROK(1, 1.2)	15.4	65.3	71.8
	1.5	GROK(2, 1.2)	2.9	66.2	71.9
PEGASUS Multinews	1.5	Beam	50	66.7	72.4
	1.5	GROK(1, 1.2)	15.6	65.2	71.6
	1.5	GROK(2, 1.2)	5.3	67.0	71.3
PEGASUS XSUM	1.5	Beam	**50**	**68.8**	**74.6**
	1.5	GROK(1, 1.2)	32.6	65.9	72
	1.5	GROK(2, 1.2)	26.6	66.5	72.1
Fairseq	1.5	Sampling	**50**	**68.8**	**74.6**
	1.5	Sampling + GROK(1, 1.2)	**46.5**	**70.1**	**75.2**

Table 2: Augmentation/Concatenation. See Section 5 for details. Performance is average of 3 runs.

future work, we would like to also paraphrase the answers, as well as the context of the answer. This follows the general logic of data augmentation to show the model various versions of data to increase robustness and generalizability.

We'd also like to implement test time augmentation. Just like the model can benefit from a question being asked a different way (a beneficial rephrasing) during learning, so too can the model benefit from multiple opportunitues to "hear" a question at test time. In this proposed scheme, the model will answer the original question and its paraphrased form (potentially, multiple paraphrased forms) and ensemble an answer.

8 Supplementary Materials

Code and datasets produced for this paper are available in our Github repository (temporarily unlinked for anonymity's sake).

Acknowledgments

Thank you to Prof. Greg Durrett for compiling and teaching this course. The lectures were well paced and gave valuable insight into NLP fundamentals and some state of the art approaches.

Another large thank you to the TAs for this course. They have gone above and beyond to offer help and insight in all aspects of the course.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *CoRR*, abs/1808.09381.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model. *CoRR*, abs/1906.01749.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. 2017. Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 198–202, Santiago de Compostela, Spain. Association for Computational Linguistics.

Sosuke Kobayashi. 2018a. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Sosuke Kobayashi. 2018b. Contextual augmentation: Data augmentation by words with paradigmatic relations. *CoRR*, abs/1805.06201.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's WMT19 news translation task submission. *CoRR*, abs/1907.06616.

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 *Proceedings of NAACL-HLT 2019: Demonstrations*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel Ngonga, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.

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*, pages 38–45, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.