Mosaic Augmentation for Text: Cropping and Collaging as Cross-Domain Techniques

Anonymous ACL submission

Abstract

We present new visually inspired *cropping* and *collaging* data augmentations for text. We test how these augmentations impact data-scarce scenarios over multiple NLP tasks: Name Entity Recognition, Extractive Question Answering and Abstractive Summarization. Ablation studies show different prevailing reasons for the augmentations' effectiveness for each different task, but all benefit from our approach. We achieve significant improvements over baselines, in particular for limited data use cases.

1 Introduction

Data augmentations are a set of techniques used to generate additional data examples based on existing training sets, and are particularly useful when the data source is scarce. These leverage data manipulations at character (Belinkov and Bisk, 2018), word (Zhang et al., 2015) phrase (Shi et al., 2021), or document (Shen et al., 2020) level.[1] Beyond textual applications, data augmentation is widely used in various fields of machine learning, including computer vision (CV) (Shorten and Khoshgoftaar, 2019) and audio processing (Park et al., 2019). However, input-space augmentations tend to be developed with a specific modality in mind (e.g., speech, vision, or text) and are generally applied only within that domain.

In this work, we develop textual augmentations inspired by concepts originally conceived in the vision domain, thus opening the door for a vast body of literature and potential applications by adopting methodologies across modalities. In particular, we build upon *Mosaic*, a popular CV augmentation introduced by (Bochkovskiy et al., 2020) and used in various follow-up works (Hao and Zhili, 2020; Jocher et al., 2020; Wei et al., 2020).

We rely extensively on two main ideas implemented in *Mosaic*, namely *cropping* and *collaging*,

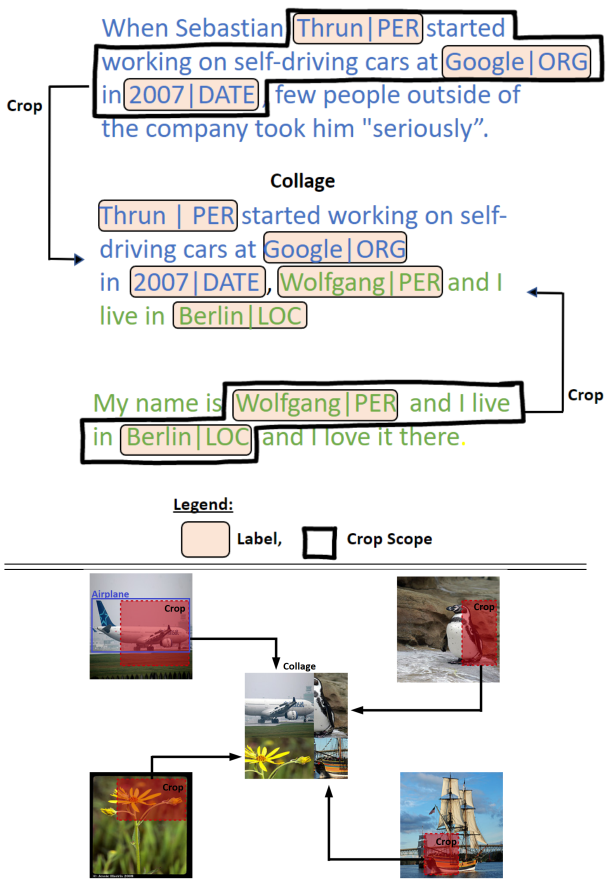


Figure 1: Top: Mosaic augmentation for Named Entity Recognition. Using two data examples, we *crop* random regions around labeled entities, then *collage* them by concatenation. Bottom: *cropping* and *collaging* augmentation visualization for images, Figure adapted from (Takahashi et al., 2020)

as exemplified in Fig 1. We chose them for their simplicity, intuitive articulation, their wide usage in CV implementations and their ability to compose with other augmentations.

We combine *cropping* and *collaging* into a new, fully analogous, mosaic augmentation for the text domain and show performance improvements over baselines on 3 new tasks: Named Entity Recognition (NER), Extractive Question Answering and

[1] See (Shorten et al., 2021), Dhole et al. (2021) and Bayer et al. (2021) for recent surveys of data augmentations in NLP.

Abstractive Summarization.

Our main contributions are: (1) We articulate and implement *cropping* and *collaging* inspired augmentations for three NLP tasks. (2) We demonstrate adoption of augmentation concepts from CV to NLP, opening the door to cross modality, domain-free augmentations. (3) We identify the effects and key reasons of why these augmentations help, in particular for low data resources scenarios.

2 Background: *Mosaic*, *Cropping* and *Collaging* in Images

In this section we briefly describe the *Mosaic* approach to CV augmentation, particularly focusing on *cropping* and *collaging*, which we later adapt to the textual domain.

Mosaic image augmentations were popularized by YOLOv4 (Bochkovskiy et al., 2020), which used the method extensively in object detection and built upon prior works describing related image combination approaches, including CutMix (Yun et al., 2019), Mixup (Zhang et al., 2018) and Cutout (DeVries and Taylor, 2017). *Mosaic* is composed of two components: *cropping* and *collaging*.

First, in *cropping* (Krizhevsky et al., 2012; Szegedy et al., 2015), a random region of the original image is used as the new example, keeping the same label, and transforming bounding boxes as applicable. For example, in Fig 1 a random crop of the airplane is taken and used as a new training example. This enriches the variety of features learned to be associated with that semantic label.

Second, *collaging* (Yun et al., 2019; Takahashi et al., 2020) tiles several (possibly cropped) images into a combined sample. For example, in Fig 1, cropped regions of four different images are combined to create a new sample, shown in the figure center. This process can help models to handle occlusions (Fong and Vedaldi, 2019), reduces chances for shortcut learning (Geirhos et al., 2020), increases effective batch size, and limits overfitting on global context.

3 Mosaic in the Text Domain

As described above, *mosaic* is composed of *cropping* and *collaging*.

We make the analogy of *cropping* in the text domain by selecting text substrings of the contexts, constraining positions according to task label bounds where appropriate. For example, in the NER task, start and end indices of the entity are the label bounds, and we crop contexts that include these, as seen in Fig 1. TODO: sentence on why we constrain

To realize *collaging* in the text domain, we concatenate examples' contexts together, adjusting the label positions as needed. This is illustrated in Fig 1, where we combine *cropped* portions from top blue and bottom green sentences and *collage* them together by concatenating them into a single example. Note that combining images together requires an additional rescaling or filling-in strategy, as the new image they combine to is usually bounded by a fixed size. The direct corollary to text is the bound imposed by the tokenizer and architecture's maximum number tokens.

Related to our method, text concatenation is used for data augmentation in neural machine translation. (Nguyen et al., 2021) concatenates translation pairs among four target/source languages, while (Kondo et al., 2021) concatenates sources and their back-translations. senMixup and wordMixup from (Guo et al., 2019) use a Mixup (Zhang et al., 2018) inspired strategy in text embedding space. Our work differs from these by taking a broad view of *collaging*, adapting it to several NLP tasks, and by combining it with *cropping* to make the augmentation analogous to image mosaic.

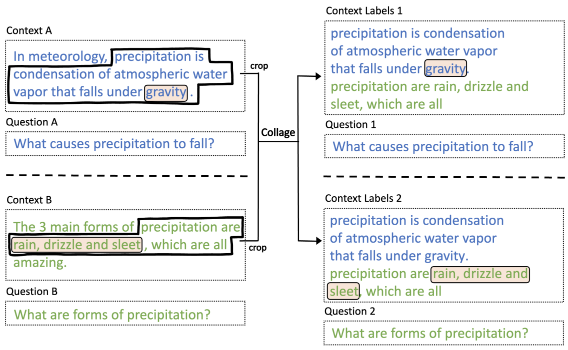
4 Methodology

In this section, we present our *cropping*- and *collaging*- inspired augmentation for three major NLP tasks, namely NER, Extractive Question Answering, and Abstractive Summarization. In each of these, we outline the analogy and adaptions of the visual concepts into the textual domain.

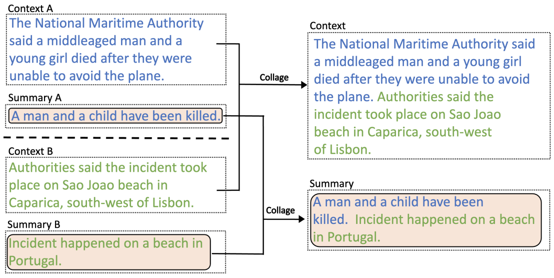
4.1 Per Task Augmentations Articulation

At every epoch for all tasks, we first randomly shuffle the dataset and apply our augmentation to successive pairs of examples as described below, each time creating a new pair of examples. With this process, different examples are paired in each epoch, and the total number of training steps is maintained.

Name Entity Recognition. In this task, each data example from the dataset is defined by a context and label for every token in the context. We define the mosaic augmentation in this task as follows. Given two examples, (1) from each example, crop a random region containing all entity-labeled tokens;



(a) Mosaic augmentation for Extractive Question Answering on 2 examples. From each data example, *collage* the contexts and summaries together by concatenating them together.



(b) *concat* augmentation for Abstractive Summarization on 2 examples. From each data example, *collage* the contexts and summaries together by concatenating them together.

Figure 2: Augmentations, Left: Extractive Question Answering, Right: Abstractive Summarization

(2) concatenate the cropped contexts in each order, to generate two new training samples. See Fig 1.

Extractive Question Answering. In this task, each data example from the dataset is defined by a triplet: a context, a question, and an answer supplied as the word positions in the context that contain the answer. We define the mosaic augmentation in this task as follows. Given two examples, (1) from each example context, crop a random region that contains the answer; (2) concatenate the cropped contexts; (3) using the combined context, generate two new training samples, one with each question/answer pair. See Fig 2a.

Abstractive Summarization. Each data example from the dataset is defined by a context and a target summary. In this task we only *concatenate* the different contexts and corresponding summaries, as there is no way to verify we don't drop text used in the summary, as seen in Fig 2b.

5 Experiments

5.1 Experimental Setup

We perform extensive experiments on three standard NLP tasks. For each task, we trained a relevant transformer architecture without any augmentations as baseline, and compared with same architecture trained with each of our augmentations.

For NER, we used the MRQA (Fisch et al., 2019) version of five datasets: bc2gm (Smith et al., 2008), conll2003 (Tjong Kim Sang and De Meulder, 2003), ncbi-disease (Doğan et al., 2014), species800 (Pafilis et al., 2013), wnut17 (Derczynski et al., 2017). For Extractive Qustion Answering, we average over 2 datasets: SQuAD (Rajpurkar et al., 2016), hotpotqa (Yang et al., 2018). For AS, we measure on the samsum (Gliwa et al., 2019) and xsum (Narayan et al., 2018) datasets.

For each task, 5 different random seeds were used for all architectures and datasets, and their results averaged, to mitigate seed outlier effects as described in (Picard, 2021). Full results including means and standard deviations are shown in the appendix tables.

All models were trained on a single GPU over 10 epochs. For NER, we train *bert-base-uncased* (Devlin et al., 2019) using default huggingface parameters. For EQA, we train *roberta-base* (Zhuang et al., 2021) using default parameters from (Ram et al., 2021). For AS, we train *t5-small* (Raffel et al., 2020) model with fixed "*summarize:*" prompt using default huggingface parameters.[2] We make our code publicly available.

5.2 Augmentations

We evaluate mosaic and each of its component augmentations in our experiments:

concat combines two distinct examples by only concatenating contexts, but without cropping.

crop applies only cropping to each example, without concatenating.

mosaic combines two examples by cropping and concatenating contexts as described in Sec. 4.

In all cases we shift the labels (start/end indices of answers/entities) according to the length of the sequence added before the context for EQA and NER. For AS, we concatenate the summaries.

We compare against two baselines: **baseline** does not apply any augmentations. For NER, we

[2] https://github.com/huggingface/transformers/blob/master/examples/

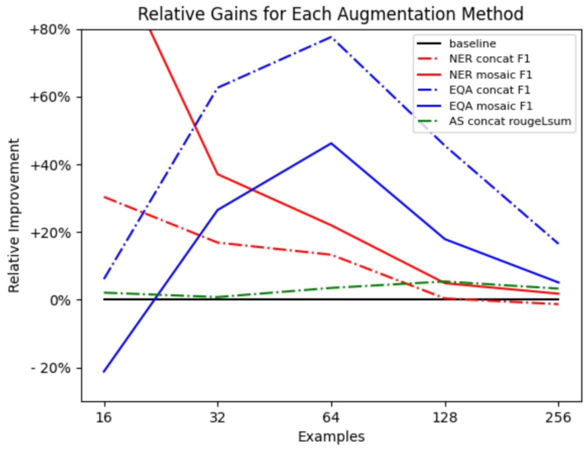


Figure 3: Relative improvements using our augmentations on the relevant metric (F1, rougeLsum) per task. Our augenations improve over baseline for all tasks with dataset size at least 32.

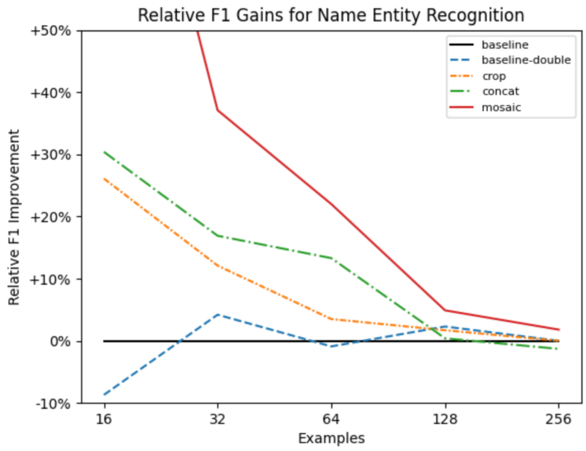


Figure 4: Ablations for Named Entity Recognition. *Mosaic* is better not only than *concat* or *crop* alone, but also than the combination of their individual contributions.

also include **baseline-double**, which repeats each training sample twice in each epoch (before shuffling) and doubles the batch size, so that the total number of training steps is the same but each example is seen twice per epoch. Since the samples generated by our augmentations from each example pair may contain data from both original examples, we include this stronger baseline to control for this possible doubling effect.

6 Results and Discussion

Fig. 3 shows a summary of the results. Each line shows *relative improvement* for each method over the baseline.

First, Fig. 3 shows that our augmentations improve F1 scores for all dataset sizes in Named Entity Recogntion (NER) and all but the smallest size for Extractive Question Answering (EQA). Abstractive Summarization (AS) improves in rouge score by a small but consistent amount of 1-5%.

For Named Entity Recognition, the smallest data sizes tend to benefit the most, with improvements up to 108% relative for 16 original examples, and 47% for 32 examples. Larger data sizes with 256 original examples do not benefit as much, but still show improvement.

Fig. 4 shows further ablation studies on the NER task. *Mosaic* is better not only than either *concat* or *crop* alone, but also than the combination of their individual contributions: for 256 dataset size in particular, *crop* shows no gain over baseline and *concat* a slight degradation (-1.7% relative), while combining them into a mosaic results in a 1.5% relative *improvement*. This shows that not just concatenation or cropping, but their combination is important to realize best performance for this task. Furthermore, *baseline-double*, which doubles the batch size and examples seen each epoch, performs similarly to *baseline*, showing that variation from our augmentation operations, and not possible data repetition, causes the increased performance.

For Extractive Question Answering, our method achieves highest relative improvement for data sizes of 64 examples (77.6% *concat* and 46.2% *mosaic*, blue lines in Fig. 3), with smaller but meaningful improvements in both larger and smaller dataset sizes. In contrast to NER, cropping does not seem to help in this task, with *concat* alone performing best. We believe this is because in the EQA task, the model must compare between question and context to find the answer, and the longer training contexts supply more negative "distractor" segments in the training-time comparison. For this task, this appears to be a larger effect than that offered by more variation in crops and positions.

Applied to Abstract Summarization, *concat* yields small but consistent gains, between 1% to 5% relative improvement in rougeLsum at all data sizes (green line in Fig. 3), demonstrating its applicability to a wide range of tasks.

7 Conclusion

We adapt mosaic data augmentations to text, finding it effective in three tested NLP tasks, with largest gains in NER. More broadly, we hope to adapt more augmentations from CV to NLP, e.g., scaling and color shift, which may apply to token representations.

References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Gloria Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. 2021. Nl-augmenter: A framework for task-sensitive natural language augmentation.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*.

Ruth Fong and Andrea Vedaldi. 2019. Occlusions for effective data augmentation in image classification.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study.

Wang Hao and Song Zhili. 2020. Improved mosaic: Algorithms for more complex images. *Journal of Physics: Conference Series*, 1684(1):012094.

Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu , changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. 2020. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements.

Seiichiro Kondo, Kengo Hotate, Tosho Hirasawa, Masahiro Kaneko, and Mamoru Komachi. 2021. Sentence concatenation approach to data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 143–149, Online. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

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. *ArXiv*, abs/1808.08745.

Toan Q. Nguyen, Kenton Murray, and David Chiang. 2021. Data augmentation by concatenation for low-resource translation: A mystery and a solution. *CoRR*, abs/2105.01691.

Evangelos Pafilis, Sune P. Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen. 2013. The SPECIES and ORGANISMS resources for fast and accurate identification of taxonomic names in text. *PLoS ONE*, 8(6):e65390.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*.

David Picard. 2021. Torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *CoRR*, abs/2109.08203.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3066–3079, Online. Association for Computational Linguistics.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation.

Haoyue Shi, Karen Livescu, and Kevin Gimpel. 2021. Substructure substitution: Structured data augmentation for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3494–3508, Online. Association for Computational Linguistics.

Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1).

Connor Shorten, Taghi M. Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of Big Data*, 8(1).

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A Struble, Richard J Povinelli, Andreas Vlachos, William A Baumgartner, Lawrence Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W John Wilbur. 2008. Overview of BioCreative II gene mention recognition. *Genome Biology*, 9(S2).

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. 2020. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2917–2931.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Zhiwei Wei, Chenzhen Duan, Xinghao Song, Ye Tian, and Hongpeng Wang. 2020. Amrnet: Chips augmentation in aerial images object detection.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

S. Yun, D. Han, S. Chun, S. Oh, Y. Yoo, and J. Choe. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer*

Vision (ICCV), pages 6022–6031, Los Alamitos, CA, USA. IEEE Computer Society.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. fn. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Example Appendix

Examples	Aug	Exact Match	F1
16.0	`concat`	+0.447(+7.0%)	+0.666(+6.1%)
16.0	`mosaic`	-2.21(-34.7%)	-2.298(-21.2%)
32.0	`concat`	+4.697(+77.2%)	+6.441(+62.6%)
32.0	`mosaic`	+1.047(+17.2%)	+2.722(+26.5%)
64.0	`concat`	+8.585(+86.0%)	+12.151(+77.6%)
64.0	`mosaic`	+3.148(+31.5%)	+7.237(+46.2%)
128.0	`concat`	+9.547(+48.9%)	+12.951(+45.5%)
128.0	`mosaic`	+1.305(+6.7%)	+5.106(+17.9%)
256.0	`concat`	+5.775(+18.8%)	+7.1(+16.5%)
256.0	`mosaic`	-1.0(-3.3%)	+2.188(+5.1%)

Table 1: Task: Extractive Question Answering. Results average across over datasets: SQuAD, hotpotqa. Results show deltas from baseline in format <Absolute delta>(<Relative delta>). Model:roberta-base. Averaged over 5 random seeds [42-46]

Examples	Aug	Exact Match	F1
		hotpotqa	
16.0	baseline	6.372 ± 1.551	10.838 ± 2.240
16.0	concat	6.819 ± 3.080	11.505 ± 4.649
16.0	mosaic	4.162 ± 1.678	8.541 ± 2.269
32.0	baseline	6.087 ± 1.137	10.287 ± 1.555
32.0	concat	10.785 ± 2.657	16.728 ± 3.758
32.0	mosaic	7.135 ± 1.497	13.009 ± 2.889
64.0	baseline	9.981 ± 2.164	15.655 ± 3.321
64.0	concat	18.566 ± 2.810	27.806 ± 3.788
64.0	mosaic	13.129 ± 1.436	22.892 ± 1.453
128.0	baseline	19.539 ± 4.957	28.486 ± 6.914
128.0	concat	29.087 ± 1.257	41.436 ± 2.073
128.0	mosaic	20.844 ± 2.655	33.592 ± 3.502
256.0	baseline	30.656 ± 0.916	43.054 ± 1.243
256.0	concat	36.431 ± 1.926	50.154 ± 2.206
256.0	mosaic	29.656 ± 1.414	45.242 ± 0.329
		SQuAD	
16.0	baseline	5.012 ± 2.681	8.589 ± 4.769
16.0	concat	5.573 ± 3.014	8.700 ± 4.200
16.0	mosaic	6.993 ± 3.047	11.247 ± 4.420
32.0	baseline	12.416 ± 4.581	18.556 ± 6.170
32.0	concat	14.539 ± 3.384	20.898 ± 5.052
32.0	mosaic	15.527 ± 1.257	22.233 ± 2.055
64.0	baseline	23.560 ± 1.816	30.596 ± 2.967
64.0	concat	25.930 ± 3.188	34.723 ± 3.403
64.0	mosaic	28.233 ± 2.556	36.641 ± 2.780
128.0	baseline	32.457 ± 5.131	40.666 ± 5.847
128.0	concat	38.730 ± 5.253	48.397 ± 5.312
128.0	mosaic	39.060 ± 4.066	47.383 ± 4.447
256.0	baseline	46.147 ± 4.812	55.676 ± 4.996
256.0	concat	51.804 ± 0.479	61.301 ± 0.168
256.0	mosaic	49.776 ± 2.214	59.122 ± 2.388

Table 2: Task: Extractive Question Answering. Results for datasets: SQuAD, hotpotqa. Model:roberta-base. Averaged over 5 random seeds

Examples	Aug	Accuracy	Recall	Precision	F1
16.0	`double_baseline`	+0.0(+0.0%)	-0.001(-7.1%)	-0.008(-8.7%)	-0.002(-8.7%)
16.0	`concat`	+0.001(+0.1%)	+0.006(+42.9%)	+0.035(+38.0%)	+0.007(+30.4%)
16.0	`crop`	+0.001(+0.1%)	+0.005(+35.7%)	+0.031(+33.7%)	+0.006(+26.1%)
16.0	`mosaic`	+0.001(+0.1%)	+0.02(+142.9%)	+0.042(+45.7%)	+0.025(+108.7%)
32.0	`double_baseline`	+0.0(+0.0%)	+0.003(+2.2%)	+0.014(+6.3%)	+0.007(+4.2%)
32.0	`concat`	+0.006(+0.7%)	+0.044(+31.9%)	+0.03(+13.5%)	+0.043(+26.1%)
32.0	`crop`	+0.002(+0.2%)	+0.021(+15.2%)	+0.013(+5.9%)	+0.02(+12.1%)
32.0	`mosaic`	+0.009(+1.0%)	+0.086(+62.3%)	+0.052(+23.4%)	+0.079(+47.9%)
64.0	`double_baseline`	+0.0(+0.0%)	+0.005(+1.5%)	+0.057(+13.8%)	-0.003(-0.9%)
64.0	`concat`	+0.003(+0.3%)	+0.052(+15.9%)	+0.042(+10.2%)	+0.046(+13.3%)
64.0	`crop`	+0.001(+0.1%)	+0.008(+2.4%)	+0.007(+1.7%)	+0.012(+3.5%)
64.0	`mosaic`	+0.004(+0.4%)	+0.092(+28.0%)	+0.05(+12.1%)	+0.076(+22.0%)
128.0	`double_baseline`	+0.001(+0.1%)	+0.008(+1.7%)	+0.014(+2.7%)	+0.011(+2.3%)
128.0	`concat`	+0.001(+0.1%)	+0.017(+3.7%)	+0.009(+1.7%)	+0.013(+2.7%)
128.0	`crop`	+0.0(+0.0%)	-0.002(-0.4%)	+0.008(+1.5%)	+0.008(+1.7%)
128.0	`mosaic`	+0.002(+0.2%)	+0.049(+10.7%)	+0.004(+0.8%)	+0.035(+7.4%)
256.0	`double_baseline`	+0.0(+0.0%)	-0.001(-0.2%)	+0.002(+0.3%)	+0.0(+0.0%)
256.0	`concat`	-0.001(-0.1%)	-0.012(-2.3%)	+0.0(+0.0%)	-0.009(-1.7%)
256.0	`crop`	+0.0(+0.0%)	-0.014(-2.7%)	+0.021(+3.6%)	+0.0(+0.0%)
256.0	`mosaic`	+0.0(+0.0%)	+0.021(+4.1%)	-0.009(-1.5%)	+0.008(+1.5%)

Table 3: **Task:** Name Entity Recognition. Results average across **5 Datasets:** bc2gm, conll2003, ncbi-disease, species800, wnut17. Results show deltas from baseline in format <Absolute delta>(<Relative delta>). **Model:**bert-base-uncased. Averaged over **5 random seeds**

Examples	Aug	Accuracy	Recall	Precision	F1
			bc2gm		
16.0	baseline	0.895 ± 0.000	0.001 ± 0.001	0.037 ± 0.071	0.001 ± 0.002
16.0	baseline-double	0.894 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
16.0	mosaic	0.894 ± 0.001	0.002 ± 0.003	0.021 ± 0.021	0.004 ± 0.005
16.0	concat	0.894 ± 0.000	0.001 ± 0.001	0.017 ± 0.011	0.002 ± 0.002
32.0	baseline	0.906 ± 0.007	0.149 ± 0.102	0.242 ± 0.055	0.176 ± 0.094
32.0	baseline-double	0.907 ± 0.005	0.153 ± 0.042	0.246 ± 0.030	0.187 ± 0.040
32.0	mosaic	0.913 ± 0.008	0.219 ± 0.093	0.252 ± 0.048	0.232 ± 0.073
32.0	concat	0.911 ± 0.006	0.181 ± 0.061	0.251 ± 0.035	0.208 ± 0.054
64.0	baseline	0.919 ± 0.006	0.266 ± 0.071	0.329 ± 0.023	0.292 ± 0.052
64.0	baseline-double	0.922 ± 0.004	0.319 ± 0.073	0.322 ± 0.023	0.318 ± 0.049
64.0	mosaic	0.926 ± 0.004	0.330 ± 0.047	0.346 ± 0.017	0.337 ± 0.032
64.0	concat	0.924 ± 0.005	0.304 ± 0.056	0.334 ± 0.024	0.318 ± 0.041
128.0	baseline	0.933 ± 0.003	0.414 ± 0.051	0.406 ± 0.012	0.409 ± 0.031
128.0	baseline-double	0.933 ± 0.003	0.444 ± 0.068	0.411 ± 0.016	0.426 ± 0.040
128.0	mosaic	0.931 ± 0.003	0.398 ± 0.058	0.393 ± 0.023	0.394 ± 0.037
128.0	concat	0.932 ± 0.005	0.427 ± 0.046	0.403 ± 0.028	0.413 ± 0.030
256.0	baseline	0.934 ± 0.005	0.395 ± 0.074	0.421 ± 0.025	0.406 ± 0.051
256.0	baseline-double	0.934 ± 0.002	0.390 ± 0.036	0.425 ± 0.018	0.406 ± 0.025
256.0	mosaic	0.934 ± 0.004	0.413 ± 0.073	0.442 ± 0.022	0.424 ± 0.045
256.0	concat	0.933 ± 0.005	0.369 ± 0.073	0.434 ± 0.031	0.397 ± 0.057
			conll2003		
16.0	baseline	0.833 ± 0.001	0.011 ± 0.016	0.170 ± 0.159	0.020 ± 0.029
16.0	baseline-double	0.833 ± 0.001	0.011 ± 0.014	0.176 ± 0.186	0.020 ± 0.026
16.0	mosaic	0.835 ± 0.001	0.019 ± 0.013	0.332 ± 0.176	0.035 ± 0.023
16.0	concat	0.833 ± 0.001	0.008 ± 0.005	0.333 ± 0.210	0.014 ± 0.010
32.0	baseline	0.871 ± 0.014	0.235 ± 0.077	0.410 ± 0.028	0.292 ± 0.065
32.0	baseline-double	0.868 ± 0.010	0.225 ± 0.060	0.451 ± 0.041	0.294 ± 0.050
32.0	mosaic	0.901 ± 0.007	0.416 ± 0.050	0.498 ± 0.034	0.453 ± 0.040
32.0	concat	0.888 ± 0.011	0.339 ± 0.072	0.460 ± 0.040	0.386 ± 0.048
64.0	baseline	0.925 ± 0.003	0.574 ± 0.024	0.574 ± 0.043	0.574 ± 0.033
64.0	baseline-double	0.922 ± 0.006	0.556 ± 0.043	0.550 ± 0.050	0.553 ± 0.046
64.0	mosaic	0.934 ± 0.003	0.654 ± 0.017	0.632 ± 0.021	0.643 ± 0.018
64.0	concat	0.933 ± 0.002	0.637 ± 0.019	0.625 ± 0.031	0.631 ± 0.024
128.0	baseline	0.943 ± 0.002	0.684 ± 0.008	0.648 ± 0.007	0.666 ± 0.008
128.0	baseline-double	0.944 ± 0.001	0.687 ± 0.007	0.657 ± 0.007	0.672 ± 0.006
128.0	mosaic	0.945 ± 0.001	0.700 ± 0.004	0.661 ± 0.010	0.679 ± 0.007
128.0	concat	0.946 ± 0.002	0.699 ± 0.007	0.667 ± 0.014	0.683 ± 0.009
256.0	baseline	0.950 ± 0.001	0.732 ± 0.009	0.692 ± 0.014	0.712 ± 0.010
256.0	baseline-double	0.952 ± 0.001	0.737 ± 0.008	0.702 ± 0.010	0.719 ± 0.008
256.0	mosaic	0.950 ± 0.001	0.732 ± 0.004	0.684 ± 0.002	0.707 ± 0.002
256.0	concat	0.949 ± 0.002	0.728 ± 0.013	0.679 ± 0.011	0.702 ± 0.011

Table 4: Task: Name Entity Recognition. Results for datasets: bc2gm, conll2003. Model:bert-base-uncased. Averaged over 5 random seeds

Examples	Aug	Accuracy	Recall	Precision	F1
			ncbi-disease		
16.0	baseline	0.927 ± 0.003	0.059 ± 0.039	0.252 ± 0.038	0.092 ± 0.050
16.0	baseline-double	0.927 ± 0.003	0.054 ± 0.036	0.244 ± 0.047	0.086 ± 0.049
16.0	mosaic	0.933 ± 0.003	0.148 ± 0.059	0.311 ± 0.059	0.198 ± 0.066
16.0	concat	0.930 ± 0.002	0.089 ± 0.041	0.278 ± 0.054	0.133 ± 0.051
32.0	baseline	0.940 ± 0.002	0.287 ± 0.035	0.382 ± 0.015	0.327 ± 0.024
32.0	baseline-double	0.941 ± 0.001	0.308 ± 0.046	0.404 ± 0.008	0.348 ± 0.031
32.0	mosaic	0.943 ± 0.001	0.354 ± 0.032	0.387 ± 0.027	0.369 ± 0.027
32.0	concat	0.942 ± 0.001	0.318 ± 0.034	0.375 ± 0.012	0.343 ± 0.020
64.0	baseline	0.960 ± 0.001	0.524 ± 0.016	0.458 ± 0.010	0.489 ± 0.010
64.0	baseline-double	0.961 ± 0.001	0.549 ± 0.029	0.470 ± 0.015	0.506 ± 0.015
64.0	mosaic	0.957 ± 0.002	0.586 ± 0.024	0.414 ± 0.025	0.485 ± 0.022
64.0	concat	0.961 ± 0.002	0.582 ± 0.020	0.453 ± 0.029	0.509 ± 0.025
128.0	baseline	0.969 ± 0.001	0.646 ± 0.007	0.580 ± 0.024	0.611 ± 0.012
128.0	baseline-double	0.968 ± 0.001	0.627 ± 0.019	0.612 ± 0.009	0.619 ± 0.010
128.0	mosaic	0.967 ± 0.002	0.660 ± 0.014	0.560 ± 0.024	0.606 ± 0.009
128.0	concat	0.968 ± 0.001	0.654 ± 0.012	0.574 ± 0.022	0.611 ± 0.016
256.0	baseline	0.972 ± 0.001	0.654 ± 0.014	0.664 ± 0.019	0.659 ± 0.012
256.0	baseline-double	0.971 ± 0.001	0.657 ± 0.007	0.652 ± 0.011	0.654 ± 0.007
256.0	mosaic	0.970 ± 0.001	0.659 ± 0.009	0.620 ± 0.025	0.639 ± 0.013
256.0	concat	0.972 ± 0.000	0.666 ± 0.011	0.650 ± 0.016	0.658 ± 0.005
			species800		
16.0	baseline	0.960 ± 0.001	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
16.0	baseline-double	0.960 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
16.0	mosaic	0.959 ± 0.001	0.001 ± 0.001	0.006 ± 0.009	0.002 ± 0.002
16.0	concat	0.960 ± 0.001	0.001 ± 0.001	0.007 ± 0.017	0.001 ± 0.002
32.0	baseline	0.964 ± 0.002	0.019 ± 0.021	0.074 ± 0.065	0.030 ± 0.031
32.0	baseline-double	0.963 ± 0.002	0.018 ± 0.018	0.079 ± 0.062	0.029 ± 0.028
32.0	mosaic	0.968 ± 0.003	0.129 ± 0.076	0.235 ± 0.110	0.166 ± 0.091
32.0	concat	0.967 ± 0.002	0.072 ± 0.041	0.173 ± 0.077	0.101 ± 0.054
64.0	baseline	0.971 ± 0.002	0.214 ± 0.071	0.369 ± 0.080	0.269 ± 0.075
64.0	baseline-double	0.970 ± 0.002	0.176 ± 0.050	0.329 ± 0.110	0.229 ± 0.069
64.0	mosaic	0.972 ± 0.001	0.301 ± 0.018	0.443 ± 0.020	0.358 ± 0.013
64.0	concat	0.972 ± 0.002	0.267 ± 0.048	0.445 ± 0.060	0.334 ± 0.054
128.0	baseline	0.973 ± 0.000	0.353 ± 0.023	0.502 ± 0.022	0.413 ± 0.013
128.0	baseline-double	0.973 ± 0.001	0.353 ± 0.011	0.511 ± 0.040	0.417 ± 0.011
128.0	mosaic	0.973 ± 0.001	0.403 ± 0.031	0.521 ± 0.047	0.452 ± 0.014
128.0	concat	0.973 ± 0.001	0.360 ± 0.021	0.507 ± 0.012	0.421 ± 0.018
256.0	baseline	0.976 ± 0.001	0.373 ± 0.013	0.539 ± 0.025	0.441 ± 0.014
256.0	baseline-double	0.976 ± 0.001	0.367 ± 0.028	0.545 ± 0.020	0.438 ± 0.023
256.0	mosaic	0.977 ± 0.000	0.419 ± 0.016	0.576 ± 0.033	0.484 ± 0.012
256.0	concat	0.976 ± 0.001	0.370 ± 0.025	0.549 ± 0.021	0.441 ± 0.018

Table 5: Task: Name Entity Recognition. Results for datasets: ncbi-disease, species800, wnut17. Model:bert-base-uncased. Averaged over 5 random seeds

Examples	Aug	Accuracy	Recall	Precision	F1
		wnut17			
16.0	baseline	0.921 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
16.0	baseline-double	0.921 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
16.0	mosaic	0.921 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
16.0	concat	0.921 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
32.0	baseline	0.921 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
32.0	baseline-double	0.921 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
32.0	mosaic	0.920 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
32.0	concat	0.921 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
64.0	baseline	0.924 ± 0.005	0.063 ± 0.085	0.331 ± 0.302	0.099 ± 0.131
64.0	baseline-double	0.924 ± 0.003	0.063 ± 0.058	0.675 ± 0.299	0.103 ± 0.088
64.0	mosaic	0.932 ± 0.006	0.228 ± 0.125	0.473 ± 0.068	0.280 ± 0.146
64.0	concat	0.927 ± 0.006	0.110 ± 0.097	0.412 ± 0.258	0.161 ± 0.131
128.0	baseline	0.934 ± 0.003	0.205 ± 0.054	0.449 ± 0.055	0.280 ± 0.060
128.0	baseline-double	0.935 ± 0.005	0.227 ± 0.081	0.463 ± 0.042	0.301 ± 0.082
128.0	mosaic	0.942 ± 0.004	0.386 ± 0.053	0.468 ± 0.060	0.423 ± 0.056
128.0	concat	0.935 ± 0.006	0.247 ± 0.098	0.477 ± 0.052	0.319 ± 0.094
256.0	baseline	0.947 ± 0.003	0.424 ± 0.049	0.605 ± 0.027	0.498 ± 0.042
256.0	baseline-double	0.947 ± 0.003	0.425 ± 0.051	0.607 ± 0.033	0.500 ± 0.047
256.0	mosaic	0.947 ± 0.004	0.464 ± 0.040	0.551 ± 0.046	0.503 ± 0.041
256.0	concat	0.945 ± 0.003	0.386 ± 0.054	0.609 ± 0.012	0.471 ± 0.043

Table 6: Task: Name Entity Recognition. Results for datasets: wnut17. Model:bert-base-uncased. Averaged over 5 random seeds

Examples	Aug	rouge1	rouge2	rougeL	rougeLsum
16.0	`concat`	+0.414(+1.6%)	+0.407(+6.5%)	+0.184(+1.0%)	+0.458(+2.1%)
32.0	`concat`	+0.07(+0.3%)	-0.107(-1.4%)	-0.27(-1.3%)	+0.192(+0.8%)
64.0	`concat`	+0.948(+3.3%)	+0.282(+3.2%)	+0.17(+0.8%)	+1.091(+4.5%)
128.0	`concat`	+1.134(+3.7%)	+0.279(+2.8%)	+0.158(+0.7%)	+1.411(+5.4%)
256.0	`concat`	+0.396(+1.2%)	+0.062(+0.6%)	-0.566(-2.2%)	+0.891(+3.3%)

Table 7: Task: Abstractive Summarization. Results Averaged across datasets: xsum, samsum, showing deltas from baseline in format <Absolute delta>(<Relative delta>). Model:t5-small - fixed-prompt: "summarize:". Averaged over 5 random seeds [42-46].

Examples	Aug	rouge1	rouge2	rougeL	rougeLsum
		samsum			
16.0	`baseline`	30.218 ± 0.386	9.465 ± 0.188	24.219 ± 0.247	27.490 ± 0.288
16.0	`concat`	30.896 ± 0.336	10.207 ± 0.167	24.641 ± 0.339	28.272 ± 0.308
32.0	`baseline`	34.372 ± 0.276	12.522 ± 0.213	27.994 ± 0.192	31.239 ± 0.279
32.0	`concat`	34.219 ± 0.235	12.222 ± 0.196	27.696 ± 0.223	31.335 ± 0.250
64.0	`baseline`	36.468 ± 0.267	13.979 ± 0.269	29.844 ± 0.310	33.229 ± 0.329
64.0	`concat`	37.440 ± 0.249	14.511 ± 0.162	30.422 ± 0.152	34.580 ± 0.187
128.0	`baseline`	38.751 ± 0.328	16.080 ± 0.319	31.648 ± 0.360	35.278 ± 0.321
128.0	`concat`	40.334 ± 0.217	16.652 ± 0.203	32.492 ± 0.115	37.275 ± 0.163
256.0	`baseline`	39.754 ± 0.346	16.834 ± 0.266	32.733 ± 0.259	36.445 ± 0.305
256.0	`concat`	40.952 ± 0.193	17.188 ± 0.159	32.932 ± 0.170	37.994 ± 0.186
		xsum			
16.0	`baseline`	20.154 ± 0.028	3.062 ± 0.010	14.306 ± 0.024	15.914 ± 0.019
16.0	`concat`	20.304 ± 0.018	3.134 ± 0.011	14.251 ± 0.021	16.047 ± 0.013
32.0	`baseline`	20.461 ± 0.055	3.188 ± 0.025	14.799 ± 0.035	15.913 ± 0.050
32.0	`concat`	20.754 ± 0.026	3.274 ± 0.019	14.555 ± 0.035	16.201 ± 0.025
64.0	`baseline`	20.363 ± 0.090	3.529 ± 0.042	15.372 ± 0.071	15.676 ± 0.082
64.0	`concat`	21.287 ± 0.034	3.561 ± 0.021	15.134 ± 0.027	16.506 ± 0.030
128.0	`baseline`	21.835 ± 0.109	4.135 ± 0.056	16.543 ± 0.089	16.607 ± 0.092
128.0	`concat`	22.520 ± 0.021	4.120 ± 0.017	16.017 ± 0.026	17.431 ± 0.025
256.0	`baseline`	24.149 ± 0.020	4.933 ± 0.033	18.152 ± 0.029	18.163 ± 0.032
256.0	`concat`	23.744 ± 0.036	4.705 ± 0.006	16.821 ± 0.021	18.396 ± 0.030

Table 8: Task: Abstractive Summarization. Results on xsum and samsum datasets. Model:t5-small - fixed-prompt: "summarize:". Averaged over 5 random seeds [42-46].