All You Need is Attention: Lightweight Attention-based Data Augmentation for Text Classification

Junehyung Kim
Sungkyunkwan University
kalpa093@g.skku.edu

Sungjae Hwang*
Sungkyunkwan University
sungjaeh@skku.edu

Abstract

This paper introduces LADAM, a novel method for enhancing the performance of text classification tasks. LADAM employs attention mechanisms to exchange semantically similar words between sentences. This approach generates a greater diversity of synthetic sentences compared to simpler operations like random insertions, while maintaining the context of the original sentences. Additionally, LADAM is an easy-to-use, lightweight technique that does not require external datasets or large language models. Our experimental results across five datasets demonstrate that LADAM consistently outperforms baseline methods across diverse conditions.

1 Introduction

Text classification is a prominent research area in natural language processing (NLP), where performance heavily relies on dataset quality. With the recent emergence of large language models (LLMs), there has naturally been an increasing need for substantial datasets. For example, GPT-3 (Brown et al., 2020) utilizes 175 billion parameters, emphasizing the crucial requirement for extensive dataset training. In this context, data augmentation (DA) plays a critical role in improving text classification tasks.

Previous studies on DA have employed simple noising operations such as random insertions and deletions of words (Wei and Zou, 2019; Karimi et al., 2021). These techniques often struggle to generate diverse contexts in augmented texts and can inadvertently alter the original meaning due to their random nature. Alternatively, model-based methods (Kobayashi, 2018; Wu et al., 2019) have been employed to replace words with synonyms, aiming to maintain the original context of sentences. However, the effectiveness of this approach heavily relies on the quality of the pre-trained dataset used for the model and may be challenging to implement due to dependencies on external resources like heavy language models. Interpolation is another DA method (Zhang et al., 2017; Guo et al., 2019; Sun et al., 2020). However, when applied directly to raw data such as words, this approach tends to produce nonsensical sentences that can alter labels or meanings of the original sentences (Thulasidasan et al., 2019).

To address these limitations, we propose LADAM (**L**ightweight **A**ttention-based **D**ata **A**ugmentation **M**ethod), which utilizes attention mechanisms to identify synonyms. LADAM generates diverse new sentences by replacing words with synonyms while preserving the original meaning. It is lightweight, as it operates without external datasets or language models, ensuring ease of use. Our evaluation results demonstrate it' s effectiveness in text classification tasks, and we have made LADAM publicly available[1].

2 Related Work

Noising Methods. One approach in text data augmentation (DA) uses noising operators (e.g., insertion and deletion). Easy Data Augmentation (EDA) (Wei and Zou, 2019) employs four operations: *Random Insertion*, *Random Swap*, *Random Deletion*, and *Synonym Replacement* on selected words from sentences. EDA utilizes an external datasets for insertion and replacement, such as WordNet (Miller, 1995). Text AutoAugment (TAA) (Ren et al., 2021) uses EDA's four operations with a language model to select operations. In addition, An Easier Data Augmentation (AEDA) (Karimi et al., 2021) inserts one of the six punctuation marks (e.g., ".", ";", "?", ":", "!", ","). **Model-based Methods.** Another line of text DA research utilizes model-based methods. Contextual Augmentation (Kobayashi, 2018) employed

*Corresponding author

[1] https://github.com/kalpa093/ladam

Findings of the Association for Computational Linguistics: EACL 2024, pages 12866–12873

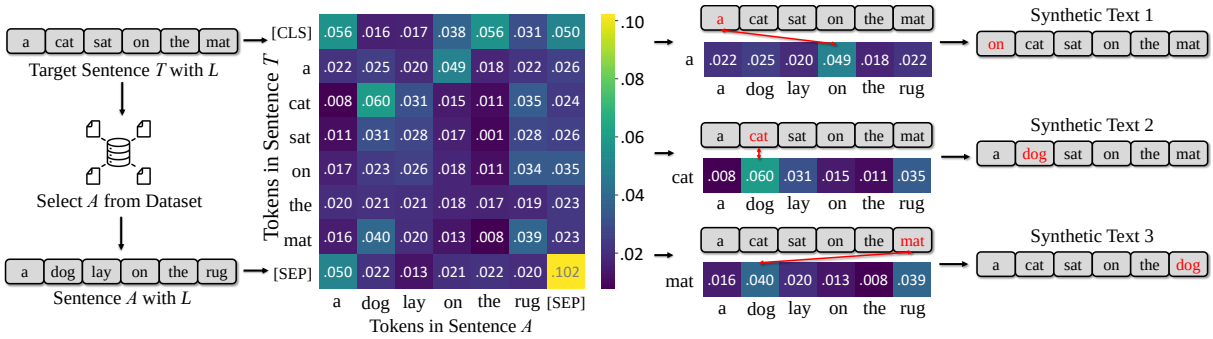

Figure 1: Overview of the data augmentation process in LADAM.

bi-directional LSTM-RNN to replace words with synonyms identified by the model. Similarly, Conditional BERT (C-BERT) (Wu et al., 2019) incorporated BERT and a conditional Masked Language Model to find synonyms.

Interpolation Methods. Interpolation methods, such as Mixup (Zhang et al., 2017) originally developed for image DA, have been adapted for text data in recent studies such as wordMixup and senMixup(Guo et al., 2019). In wordMixup, sentences are zero-padded to uniform length, and interpolation is performed across each dimension of the words in a sentence. Conversely, senMixup generates hidden embeddings for two sentences, followed by linear interpolation between them. Mixup-Transformer (Sun et al., 2020) adapted a similar approach to senMixup but utilizes BERT for generating embeddings for the two sentences.

3 LADAM

LADAM generates synthetic texts while preserving the original sentence's meaning. Key differences from existing methods include: ① minimizing the scope of word replacements, ② directly applying to raw data, and ③ being the first approach to use attention scores to preserve the original sentence's context.

Figure 1 illustrates the overall architecture of LADAM, which consists of two phases: *Context-based Sentence Selection* and *Attention-based Word Selection*. In the context-based sentence selection phase, a random target sentence T is selected for augmentation, and another random sentence A with the same label L is chosen as the assistant sentence. The two sentences are tokenized into words, and each word is vectorized using BERT Word Embedding (Devlin et al., 2019) and concatenated in the format `"[CLS] T [SEP] A [SEP]"`. In the attention-based word selection phase, LADAM

Datasets	# Classes	Avg. Length	Train Set	Test Set
CR	2	19	2,715	679
SST2	2	22	9,096	2,274
SUBJ	2	24	8,000	2,000
MPQA	2	3	8,587	1,061
TREC	6	10	4,906	500
biased_CR	2	18	1,830	458
biased_SST2	2	22	5,144	1,287
biased_SUBJ	2	23	4,400	1,100
biased_MPQA	2	2	3,293	1,061
biased_TREC	6	10	1,436	500

Table 1: Statistics of the datasets.

employs the attention mechanism in Transformer models (Vaswani et al., 2017), deriving attention scores using scaled dot-product for each word in sentences T and A. The attention scores are extracted via $softmax(\frac{QK^T}{\sqrt{d_k}})$ where Q is query, K is key, and d_k is dimension of key. At this stage, only the attention scores from T with respect to the A need to be extracted, so we focus on the attention scores where the query (Q) is T and the key (K) is A. Next, based on the random word in the T's embeddings, we identify the position of the word in the A's embeddings that has the highest attention score, then generate a synthetic sentence by exchanging the original target word with the word we choose from A. Note that as we did not modify the embeddings to create synthetic text, plaintext is obtained without detokenization. For sentences of different lengths, padding equalizes them, and masking distinguishes padding from actual tokens to minimize its impact on results.

4 Experimental Setup

4.1 Baselines

As baselines, we selected ① two recent noising approaches (EDA and AEDA), ② one model-based method (C-BERT), and ③ one interpolation method (senMixup) - all of which are open source. These baselines were selected based on their popu-

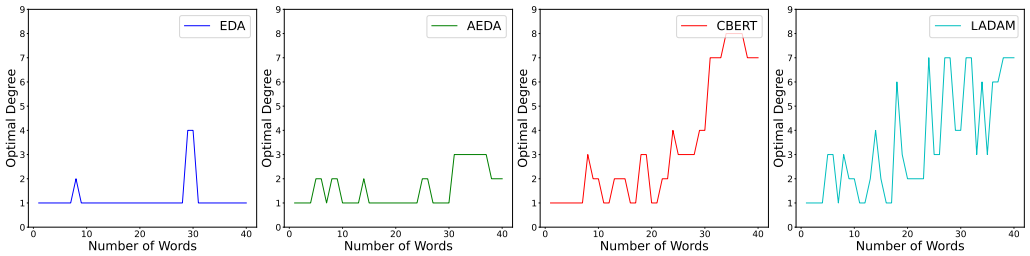

Figure 2: Optimal degree based on number of words for baselines and LADAM.

larity, as reflected by the number of citations and GitHub stars. To ensure a fair comparison, we used the original codebases of these methods without making any modifications to their architectures.

4.2 Datasets

We used five benchmarks, as outlined in Table 1. **CR**[2] (Ding et al., 2008) contains customer reviews and forum posts labeled as positive or negative. **SST-2**[3] (Socher et al., 2013) consists of single sentences from movie reviews, also labeled as positive or negative. **SUBJ**[4] (Pang and Lee, 2004) includes movie reviews labeled as subjective or objective. **MPQA**[5] (Wiebe et al., 2005) contains short news phrases labeled for positive or negative sentiment. **TREC**[6] (Hovy et al., 2001; Li and Roth, 2002) consists of question sentences categorized into six different labels.

For the SUBJ and TREC datasets, we used the same versions as the baseline methods. However, while the SUBJ dataset was identical, there was a slight difference in size; we applied an 8:2 split for cross-validation, whereas the baselines used a 9:1 split. For SST-2, we employed the same dataset version as C-BERT and senMixup (Socher et al., 2013). For MPQA, we used version 1.2, the latest version focused on contextual polarity, which had not been used by the baseline methods. Similarly, for CR, we used the latest version (Ding et al., 2008), which differs from the version used by EDA and AEDA (Hu and Liu, 2004).

Using the same benchmarks, we also constructed a biased dataset by reducing the size of all labels, except one, to 10% of their original size to evaluate LADAM under biased conditions. We also preprocessed each dataset as detailed in Section B. These datasets are open-source, freely available, and have been validated to ensure ethical compliance.

4.3 Models

For text classification models, we employed four models: BERT$_{base}$ (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2020), and distilBERT (Sanh et al., 2019) for classifiers of our experiments from huggingface (Wolf et al., 2019). They were selected due to their high performance in text classification as demonstrated by GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016). We initialized the classifiers with pre-trained parameters from the HuggingFace (Wolf et al., 2019) and employed automatic training with early stopping, using a patience of 20 epochs. All hyperparameters of each model is used by default settings except learning rate ($1e-7$) and batch size (16).

5 Results

We compared LADAM with baseline methods under various conditions, repeating each experiment five times and reporting the average results. All experiments were conducted on a GeForce RTX 3060 GPU with 12 GB of memory.

5.1 Degree of Augmentation

The degree parameter, which determines the number of word-level operations in sentence generation, significantly impacts model performance (Ren et al., 2021). We analyzed the optimal degree for both baselines and LADAM, as shown in Figure 2. For EDA and AEDA, the optimal degree remains consistently small (e.g., 1), even for long sentences, and is not proportional to sentence length. Therefore, we set the degree to a constant value of 1, regardless of sentence length. In contrast, the optimal degree for C-BERT and LADAM varies with sentence length. Using linear regression, we calculated the degree-to-word ratios as 0.16 for C-BERT and 0.17 for LADAM, and applied these optimal

[2] https://huggingface.co/datasets/SetFit/CR
[3] https://github.com/YJiangcm/SST-2-sentiment-analysis
[4] https://www.cs.cornell.edu/people/pabo/movie-review-data/
[5] https://mpqa.cs.pitt.edu/corpora/mpqa_corpus/
[6] https://huggingface.co/datasets/CogComp/trec

Methods	Datasets						Biased Datasets					
	CR	SST2	SUBJ	MPQA	TREC	Avg.	CR	SST2	SUBJ	MPQA	TREC	Avg.
No Aug	87.62±2.98	94.03±2.68	95.40±1.93	90.57±2.59	94.58±2.63	93.52	73.67±3.29	76.18±3.05	90.76±2.58	**81.58**±2.93	38.07±4.46	79.55
EDA	92.95±3.21	94.03±3.07	98.32±1.05	90.33±2.83	90.50±3.57	93.77	76.50±3.67	89.23±2.52	96.08±1.27	81.30±2.64	38.03±5.61	85.24
AEDA	95.47±2.09	97.14±1.39	98.43±1.27	90.67±3.21	96.47±2.18	95.79	81.38±2.56	90.11±2.03	96.04±1.26	81.30±2.86	**53.59**±3.37	86.71
C-BERT	93.50±1.98	96.52±1.53	98.17±1.26	89.85±3.13	96.12±1.60	95.13	64.94±3.09	81.85±2.47	91.90±2.56	78.54±3.39	45.40±4.91	80.05
senMixup	93.96±1.34	96.03±1.05	97.21±0.91	90.44±2.37	**97.29**±0.86	95.06	85.13±2.61	89.37±1.84	96.57±1.22	80.35±2.91	46.79±3.89	86.20
LADAM	**95.70**±2.35	**98.12**±0.89	**98.74**±0.93	**90.85**±2.18	94.78±2.50	**96.01**	**86.81**±2.05	**90.38**±2.28	**98.18**±0.71	80.67±2.81	46.40±3.82	**87.21**

Table 2: F1-scores of a BERT$_{base}$ classifier for each augmentation method applied to five datasets. Number after ± is a variance of five results.

DA Methods	RoBERTa	DeBERTa	distilBERT	Avg.
No Aug	93.71±2.42	93.90±1.73	92.18±1.83	93.22
EDA	95.97±2.89	95.81±2.70	93.19±2.77	94.68
AEDA	96.43±2.50	96.18±2.16	94.85±2.29	95.81
C-BERT	95.34±1.52	95.34±1.68	94.52±1.61	95.01
senMixup	96.10±1.88	96.24±1.59	93.73±1.77	95.28
LADAM	**96.69**±2.09	**96.40**±1.93	**95.94**±1.75	**96.23**

Table 3: Average F1-scores of various BERT classifiers on five datasets. Number after ± is an average variance.

values in our experiments. Notably, we did not analyze the degree for senMixup, as it does not involve word-level editing. The process for determining the optimal degree is detailed in Section A.

5.2 Main Results

Original Datasets. Table 2 presents F1-scores of the BERT$_{base}$ model for the text classification task, comparing the performance of LADAM with baselines on five datasets. As shown, LADAM achieved the highest average performance.

Biased Datasets. LADAM's performance may be affected by imbalanced label distributions, as it replaces words from sentences with the same label. However, as shown in Table 2, LADAM outperforms the baselines, proving its effectiveness despite dataset imbalance.

Classifiers. We applied LADAM to various BERT model families. As summarized in Table 3, LADAM consistently shows better performance than the baselines. This indicates that LADAM can be effectively applied across different models.

5.3 Context Preservation

Now, we verify LADAM's preservation of the original sentence meaning using Cosine Similarity (Li and Han, 2013) and Locally Linear Embedding (LLE) (Roweis and Saul, 2000).

Cosine Similarity. Using Cosine Similarity, we measured the similarity between vectors of the original and augmented sentences across five different datasets. The Cosine Similarity values for each dataset are as follows: CR (0.9996), SST-2 (0.9999), SUBJ (0.9982), MPQA (0.9977), and TREC (0.9999). These high Cosine Similarity values indicate that LADAM effectively preserves the meaning of the original sentences.

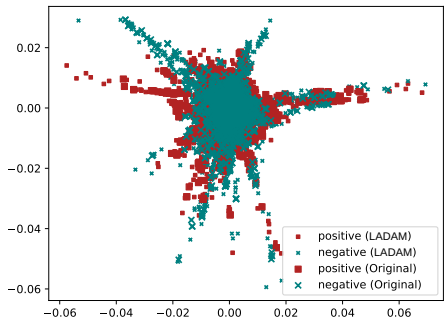

Figure 3: Data distributions of SST-2.

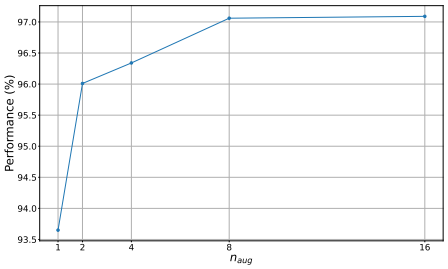

Figure 4: Performance achieved with various n_{aug}.

LLE. Figure 3 visualizes the vectors of original and augmented sentences on the SST-2 dataset using LLE. It confirms that LADAM preserves the meaning of the original sentences, as the augmented sentences closely cluster around the original sentences.

5.4 Ablation Study

In this section, we evaluted LADAM under various configurations.

5.4.1 Size of Augmentation

The parameter n_{aug} denotes the number of augmented sentences derived from a single sentence. To assess whether n_{aug} impacts text classification performance, we evaluated LADAM using n_{aug} values of 1, 2, 4, 8, 16 with a BERT$_{base}$ model. As

Methods	CR	SST2	SUBJ	MPQA	TREC
No Aug	87.62±2.98	94.03±2.68	95.40±1.93	90.57±2.59	94.58±2.63
LADAM	**95.70**±2.35	**98.12**±0.89	**98.74**±0.93	**90.85**±2.18	**94.78**±2.50
LADAM v.A	94.64±2.66	97.52±1.32	97.70±1.06	89.65±2.25	94.50±2.49
LADAM v.B	83.68±3.05	90.85±2.23	90.65±2.44	89.00±1.85	93.81±2.41

Table 4: F1-scores of variations of LADAM.

shown in Figure 4, performance converges at n_{aug} 8.

5.4.2 Effectiveness of Attention Mechanism

We evaluated LADAM in two additional configurations: LADAM v.A, which performs random word replacement without attention scores, and LADAM v.B, which replaces all words at corresponding positions in the target sentence, similar to applying SenMixup to raw data. As shown in Table 4, LADAM consistently outperforms both v.A and v.B. Notably, LADAM v.B underperforms compared to no augmentation across all benchmarks. This highlights the effectiveness of attention-based word selection and the importance of reducing interpolation to words in the raw data.

6 CONCLUSION

We propose LADAM, a lightweight text data augmentation approach designed to generate diverse augmented data while preserving the original sentence context. Our experiments show that LADAM outperforms baselines on both original and biased datasets, demonstrating it's effectiveness in text classification tasks. Future work will explore whether alternative attention scoring functions, such as concat and Bahdanau, can enhance LADAM's performance.

Limitations

Although LADAM outperforms baselines in text classification on the original MPQA dataset, the performance improvement is not significant. On the biased MPQA dataset, performance degradation is observed, with no augmentation outperforming all augmentation methods. This suggests that text data augmentation may not be effective for certain datasets. We plan to conduct a detailed analysis of this issue and improve LADAM in future work. Also, the classifiers used in the experiments have a relatively smaller number of parameters compared to the recently proposed large language models. However, the RoBERTa model, used in our experiments as an example, contains 355 million parameters, which is not insignificant. In fact, it is considered one of the largest models that can be practically used for current research purposes.

Additionally, LADAM employs a scaled dot-product attention function with a single attention layer. This poses a potential risk: alternative functions or multi-layered attention could either degrade or improve its ability to generate context-preserving synthetic sentences. In future research, we plan to explore various functions and multi-layered attention mechanisms for deriving attention scores in LADAM's architecture. Investigating the impact of different approaches on LADAM's performance will be a valuable area of study.

Ethics Statement

We employed five classification benchmark datasets in our experiments. Data augmentation conducted by LADAM is a recomposition of sentences in the training datasets. Each benchmark dataset has been officially released and has undergone validation to ensure ethical considerations using human annotators. Furthermore, even if the pre-trained language model, used as the backbone for attention scoring, could have been exposed to toxic data during pre-training process, since attention scores are just used for word selection which does not generate unprecedented synthetic data.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was partly supported by the four Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (No.2022-0-00688; AI Platform to Fully Adapt and Reflect Privacy-Policy Changes, No.2024-00337703; Development of satellite security vulnerability detection techniques using AI and specification-based automation tools, No.2024-00398745; Proofs and responses against evidence tampering in the new digital environment, No.2022-II221199; Graduate School of Convergence Security, Sungkyunkwan University). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the Advances in neural information processing systems*, volume 33, pages 1877–1901.

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.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Proceedings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754.

Sosuke Kobayashi. 2018. 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.

Baoli Li and Liping Han. 2013. Distance weighted cosine similarity measure for text classification. In *Proceedings of the Intelligent Data Engineering and Automated Learning–IDEAL 2013*, volume 8206, pages 611–618.

Xin Li and Dan Roth. 2002. Leraning question classification. In *Proceedings of the COLING 2002: The 19th International Conference on Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

George A Miller. 1995. Wordnet: a lexical database for english. In *Proceedings of the Communications of the ACM*, volume 38, pages 39–41.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pages 271–278.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Shuhuai Ren, Jinchao Zhang, Lei Li, Xu Sun, and Jie Zhou. 2021. Text AutoAugment: Learning compositional augmentation policy for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9029–9043.

Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. In *Proceedings of the American Association for the Advancement of Science*, volume 290, pages 2323–2326.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Proceedings of the Advances in neural information processing systems*, volume 32.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in neural information processing systems*, volume 30, pages 6000–6010.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. In *Proceedings of the Language resources and evaluation*, volume 39, pages 165–210.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *Proceedings of the Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV 19*, volume 11539, pages 84–95.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

A Process of Deriving Optimal Degree

We define a parameter representing the number of edited words in a single sentence as the "Degree" (D). In previous methods, such as AEDA or C-BERT, researchers manually set degree values to a specific proportion of words in the entire sentence (Ren et al., 2021; Wu et al., 2019). If a single operation is conducted, the similarity of context may vary depending on the length of sentences, i.e., the number of words in the sentence. Consequently, we hypothesized that dynamically adjusting this value based on the length of the sentence could be beneficial for every baselines. To test this hypothesis, we divided the dataset based on the number of words and experimented with varying the D value accordingly. This approach enables us to conduct data augmentation with an optimal D value, ensuring optimal performance regardless of the length of the sentence. Since the Degree D is regarding to number of words, the augmentation methods targeting sentence cannot be applied the degree of augmentation. This experiments includes augmentations targeting word such as EDA, AEDA, C-BERT and LADAM. First, we have analyzed data distribution of number of words from each datasets. As shown in Figure 5, datasets are not evenly distributed. TREC dataset contains fewer than 30 instances where the number of words exceeds 20. MPQA dataset is concentrated on short phrases including 2,757 single word which is 32.1% from 8,587 data (Wiebe et al., 2005). SST-2 is excluded since it has literally no sentences with length under 15. To increase reliability throughout length of sentences, CR and SUBJ datasets are employed in this experiment in light of their even data distribution. The datasets were split into subsets based on the number of words in each instance, and training was conducted with varying degrees. In cases where some subsets contained too less than 100 instances for effective training, they were merged with adjacent subsets having a similar number of words. Consequently, the optimal degree for each subset, based on the number of words, was identified through this training approach. The training was conducted using the BERT model, and the dataset was split into an 8:2 ratio of training to testing sets. We set N as the number of words in the sentences. Figure 2 depicts the optimal degree D corresponding to each N based on the results of our experiments that the model shows highest performance on the degree D. EDA and AEDA exhibit an increase in performance when D remains low, even at high N values. Therefore, we set the D value as 1 for them. On the other hand, C-BERT and LADAM involve augmenting sentences by replacing a random number of words with different words, implying a proportional relationship between N and D. We could get D for C-BERT and LADAM by linear regression to $y = \beta x$. C-BERT yields the D as 0.16 of number of words, which is almost equivalent to the original ratio (i.e., 0.15) used in previous study. LADAM yields the D as 0.17 of number of words.

B Preprocessing Datasets

Prior to training, we performed several preprocessing steps on the dataset. First, we replaced special characters such as "’", "-", "‘", "\t", and "\n" with spaces. Second, all uppercase letters were converted to lowercase. Third, we removed extra spaces at the beginning and end of sentences. Fourth, we eliminated duplicate entries and any

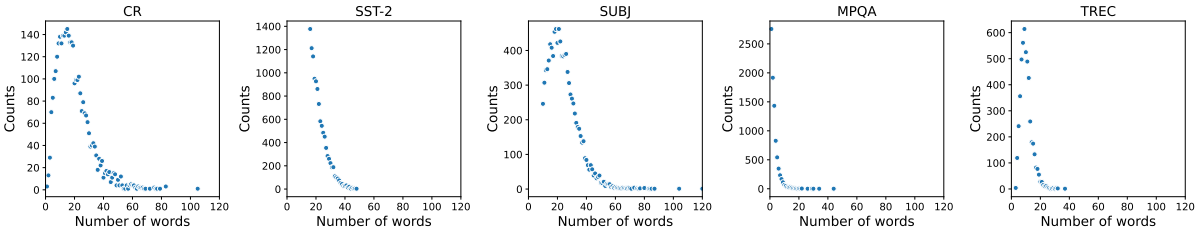

Figure 5: A distribution of each dataset based on the number of words per sentence.

	Datasets					
DA Methods	**CR**	**SST-2**	**SUBJ**	**MPQA**	**TREC**	**Avg.**
No Aug	87.68±2.98	94.07±2.68	95.40±1.93	90.61±2.60	94.76±2.65	93.57
EDA	93.07±3.22	94.03±3.07	98.32±1.05	90.41±2.84	90.55±3.58	93.81
AEDA	95.49±2.09	97.17±1.39	98.43±1.27	90.78±3.22	96.50±2.20	95.83
C-BERT	93.57±1.99	96.53±1.53	98.17±1.86	89.93±3.14	96.22±1.62	95.18
senMixup	93.98±1.34	96.03±1.05	97.21±0.91	90.48±2.37	**97.40**±0.89	95.11
LADAM	**95.73**±2.35	**98.13**±0.89	**98.74**±0.93	**90.92**±2.19	94.81±2.50	**96.03**

Table 5: Accuracy of a BERT$_{base}$ classifier for each augmentation method applied to five datasets. The numbers following ± is a variance of five results.

DA Methods	**RoBERTa**	**DeBERTa**	**distilBERT**	**Avg.**
No Aug	93.75±2.43	93.93±1.77	92.20±1.80	93.22
EDA	95.99±2.79	95.82±2.69	93.21±2.78	94.68
AEDA	96.47±2.48	96.21±2.15	94.88±2.29	95.81
C-BERT	95.39±1.50	95.38±1.68	94.53±1.61	95.01
senMixup	96.16±1.87	96.29±1.58	93.75±1.74	95.28
LADAM	**96.73**±2.04	**96.42**±1.93	**95.96**±1.76	**96.23**

Table 6: Average accuracy of various BERT classifiers on five datasets.

empty data consisting solely of spaces. Finally, we encoded the sentences in UTF-8 format. These preprocessing steps resulted in dataset sizes that differ from those used by the baselines, and the statistics of the preprocessed datasets are detailed in Section 4.

C Accuracy

This section provides the accuracy of the experimental results conducted in Section 5. Table 5 shows the accuracy of LADAM on our main experiment. Along with the F1-scores presented in Table 2, LADAM outperforms the baselines in accuracy. Moreover, as shown in Table 3 and Table 6, LADAM achieved the best performance among various families of BERT models in both accuracy and F1-score.