Generative Data Augmentation via Wasserstein Autoencoder for Text Classification

Kyohoon Jin
Department of Image Science and Arts
Chung-Ang University
Seoul, South Korea
fhzh123@cau.ac.kr

Junho Lee
Department of Artificial Intelligence
Chung-Ang University
Seoul, South Korea
jhjo32@cau.ac.kr

Juhwan Choi
School of Electrical and Electronics Engineering
Chung-Ang University
Seoul, South Korea
gold5230@cau.ac.kr

Soojin Jang
Department of Image Science and Arts
Chung-Ang University
Seoul, South Korea
sujin0110@cau.ac.kr

Youngbin Kim
Department of Image Science and Arts
Chung-Ang University
Seoul, South Korea
ybkim85@cau.ac.kr

***Abstract*—Generative latent variable models are commonly used in text generation and augmentation. However generative latent variable models such as the variational autoencoder(VAE) experience a posterior collapse problem ignoring learning for a subset of latent variables during training. In particular, this phenomenon frequently occurs when the VAE is applied to natural language processing, which may degrade the reconstruction performance. In this paper, we propose a data augmentation method based on the pre-trained language model (PLM) using the Wasserstein autoencoder (WAE) structure. The WAE was used to prevent a posterior collapse in the generative model, and the PLM was placed in the encoder and decoder to improve the augmentation performance. We evaluated the proposed method on seven benchmark datasets and proved the augmentation effect.**

***Index Terms*—Text augmentation, Generative model, Text classification**

I. Introduction

Many deep learning architectures—ranging from recurrent neural networks(RNNs) to transformers [1]—have outstanding performance in natural language processing (NLP) tasks. Recent language models pre-trained with a large amount of data, such as BERT [2] and GPT [3], outperformed humans. The performance of these models is influenced by the amount and diversity of their training data [4].

The model cannot generalize well and is easily overfitted if data are small or highly biased. Various augmentation strategies have been proposed to overcome this problem. For example, data augmentation algorithms have been developed for the image domain, but their use is limited in the text-domain. Image-domain approaches only have a minor influence on the label [5]. However, text augmentation is more complex than data augmentation in the image domain because text augmentation directly affects labels.

Text augmentation methods using generative latent variable models have been studied to alleviate this problem. A commonly used model is the variational autoencoder (VAE). The

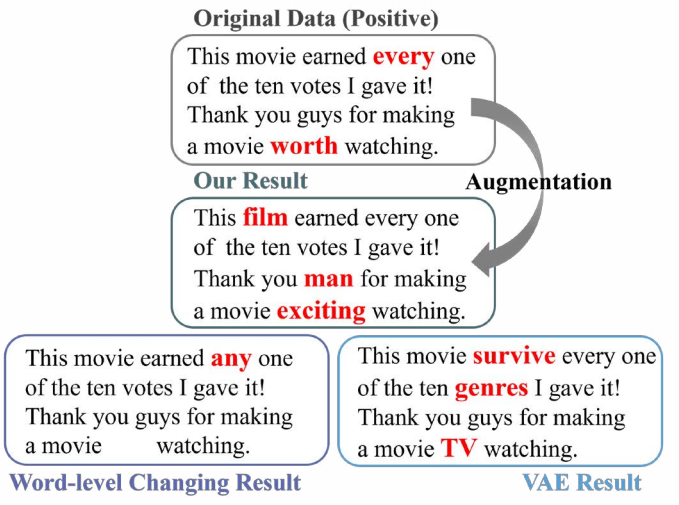

Fig. 1. Comparing augmented text by word-level changing, VAE and proposed method. Proposed method preserves the true label but other methods failing label preservation.

VAE is widely used in text generation tasks because it can add diversity in text generation [6]. However, limitations exist in applying these VAEs to NLP due to posterior collapse, where the model learns to ignore a subset of the latent variables during training. This phenomenon is common when using a strong autoregressive neural network as a generator, and it is more pronounced when modeling discrete data [7], [8].

In this paper, we propose a simple text augmentation method using a Wasserstein autoencoder (WAE, [9]) to mitigate posterior collapse during training. The WAE can prevent posterior collapse while regularizing the aggregated posterior distribution to the prior distribution. Furthermore, we used a pre-trained language model (PLM) to effectively extract the latent variable and smoothly generate sentences from the latent

variable.

We used seven benchmark datasets to validate the proposed method, which was evaluated using three models, and the accuracy improved by about 1%p to 2%p. In addition, we observed that posterior collapse affects data augmentation through the comparison with the VAE, and demonstrated the scalability of the proposed method by applying different PLMs to the encoder and decoder. Figure 1 shows comparing augmented text by word-level, VAE and proposed method.

II. Related Work

A common strategy for preventing performance degradation due to overfitting or bias in the model is data augmentation [10]. Simple methods, such as mixing images or features [11], [12], and a generative model [13] were used in the image domain. In contrast, the text data require a different approach, as the syntactics and semantics can change when the text is directly modified, as in the abovementioned methods.

A simple method is to make changes at the word-level, for example, inject new words into a sentence by replacing one letter with a neighboring letter on the keyboard [14] or randomly deleting, swapping or adding noise to create new sentences [15], [16]. These methods improve the text classification performance by allowing models to generalize to words. However, it is challenging to preserve the label and semantics of a sentence when words are transformed by random deletion and random swapping [17].

A text augmentation method using a generative latent variable model was recently studied to alleviate such problems, and the VAE has primarily been used for text augmentation based on generative latent variable models, as it can extract meaningful latent variables from a given text and does not require external procedures [18], [19]. Moreover, the VAE may result in posterior collapse, ignoring learning for a subset of latent variables. This phenomenon is typically observed when the generator is parameterized with a strong autoregressive neural network on the text [8]. Several approaches such as KL annealing and word dropout, have been introduced. These alleviate the problem, but do not completely solve it [20].

In addition, the PLM is advantageous in that it learns rich text expressions from large-scale data, so various studies have conducted augmentation using the PLM. The conditional-BERT has a modified-structure BERT by altering the segment embedding [21]. Aside from BERT, autoregressive models, such as GPT, have been used in conjunction with various methods [10], [22]. However, these methods also have limitations. Conditional-BERT relied on modifying embedding of the BERT, and it cannot be generalized to other PLMs without segment embeddings [23]. In addition, these methods discussed the limitation that the same performance could not be maintained when applying other language models [24].

In this paper, we propose a strategy based on the WAE. Unlike the VAE, the WAE regularizes the aggregated latent variable to the prior distribution. The proposed method can avoid posterior collapse, in which the model learns to ignore the learning subset of the latent variable, and the approximate posterior imitates the prior distribution. As no significant restrictions exist on using the encoder and decoder structures, implementing the proposed method in the PLM is relatively simple.

III. Method

Our model aim to avoid posterior collapse while giving variation to latent z. We replaced KL Annealing by using a PLM that has more information and can produce meaningful latent z. And prevent the mutual information of input x and latent variable z from getting smaller when using the VAE structure, the WAE structure was used to achieve the same purpose except for the mutual information term.

A. Preliminaries

We let $X = \{\mathrm{x}^{(i)}\}_{i=1}^{N}$ be a dataset of N sentences and $Z = \{z^{(i)}\}_{i=1}^{N}$ be a subset of latent variables. Each $\mathrm{x}^{(i)}$ is a sentence, and $z^{(i)}$ is a latent variable corresponding to $\mathrm{x}^{(i)}$. Autoencoders (AEs) encode a given input x in a subset of latent variable z, from which the input x is reconstructed. In other words, for input x, the output of the encoder Q is z, and we input z into the decoder G to predict x. The training objective for the AE is the cross-entropy loss, defined as follows:

$$J_{AE} = -\mathbb{E}_{z\sim Q(z|\mathrm{x})}[log\ G(\mathrm{x}|z)] \tag{1}$$

The VAE extends the AE by imposing a prior distribution P_Z on the subset of the latent variable z [20]. The VAE requires that the posterior $Q(z|\mathrm{x})$ of z for a given input x approximates P_Z. Thus, the KL divergence between $Q(z|\mathrm{x})$ and P_Z regularizes the objective function as a penalty during training. However, minimizing the KL divergence creates an overlap between the latent distributions. Therefore, the latent variable $z^{(i)}$ does not contain meaningful information about the given input $\mathrm{x}^{(i)}$ which is known as posterior collapse.

B. Wasserstein Autoencoder

WAE regularizes the posterior by imposing the constraint that the aggregated posterior distribution for the latent variable z should be equal to the prior distribution of z : $Q_Z := \mathbb{E}_{P_X}[Q(z|\mathrm{x})] = P_Z$ where P_X is data distribution [9]. In contrast, for the VAE, the posterior $Q(z|\mathrm{x})$ should be identical to P_Z for $\mathrm{x} \in X$. Thus, when using the WAE, a more meaningful latent variable z for a given x can be used as input of the decoder.

The constraints on Q_Z are alleviated by penalizing the Wasserstein distance. The penalty can be computed using the maximum mean discrepancy (MMD) [25] between Q_Z and P_Z:

$$MMD_k(P_Z, Q_Z) = \left\| \int_{z\sim P_Z} k(z,\cdot)dz - \int_{\tilde{z}\sim Q_Z} k(\tilde{z},\cdot)d\tilde{z} \right\|_{H_k} \tag{2}$$

where H_k is the reproducing kernel Hilbert space defined by k.

We define kernel k as an inverse multiquadratic kernel $k(x,y) = C/(C + ||x-y||_2^2)$, where $C = 2d_z\sigma_2^2$, which is the expected squared distance between two multivariate Gaussian vectors drawn from P_z. MMD can be estimated as follows:

$$\begin{aligned}\widehat{MMD}_k(P_Z, Q_Z) = \\ \frac{1}{N(N-1)}\sum_{l\neq j}[k(z^{(j)}, z^{(l)}) + k(\tilde{z}^{(j)}, \tilde{z}^{(l)})] \\ - \frac{2}{N^2}\sum_{l,j} k(z^{(j)}, \tilde{z}^{(l)})\end{aligned} \tag{3}$$

where $z^{(i)}$ is a sample from the prior distribution P_Z, $\tilde{z}^{(i)}$ is a sample from the aggregated posterior distribution Q_Z and the subscript (i) indicates the ith data point.

In summary, the training objective is as follows:

$$\begin{aligned}J_{WAE} = -\mathbb{E}_{Q_{\phi(Z|X)}}[log\ G_\theta(X|Z)] + \\ \frac{\lambda}{N(N-1)}\sum_{l\neq j}[k(z^{(l)}, z^{(j)}) + k(\tilde{z}^{(l)}, \tilde{z}^{(j)})] \\ - \frac{2\lambda}{N^2}\sum_{l,j} k(z^{(l)}, \tilde{z}^{(j)})\end{aligned} \tag{4}$$

where $\tilde{z}^{(i)}$ is a sample from $Q_\phi(z|\mathrm{x}^{(i)})$, $z^{(i)}$ is a sample from prior distribution P_Z and a hyperparameter λ balances the regularization term and the reconstruction term and set to 100 in our method. In addition, ϕ and θ represent the encoder and decoder parameters, respectively. To optimize ϕ and θ, we minimize Eq(4) during training time.

We used the PLM to better extract the latent variable and generate appropriate sentences for it. PLMs provided significant gains in language understanding or sentence generation by better capturing information such as semantics and syntactics [26], [27]. No structural limitation exists in the encoder and decoder for the WAE, thus, we can select and introduce the PLM as the desired model more freely than other methods. We conducted an experiment using the BART [28], a PLM that has both an encoder and decoder.

IV. Experiments

A. Datasets

To test the proposed method for various datasets, we used the SST2, SST5 [29], ProsCons (PC) [30], and MR [31] datasets, which are commonly used as performance evaluation benchmarks in data augmentation. Additionally, we included the IMDB [32], DBpedia [33], and Yelp Full[1] datasets, which are frequently used as performance performance evaluation benchmarks in text classification.

B. Experimental Setup

In the WAE, we used the BART for the encoder and decoder. The prior distribution was based on the Gaussian distribution. The maximum sentence length is 300 characters. The learning rate is 5e-6, and we used the AdamW optimizer for training.

[1]https://www.yelp.com/dataset

Classifier	SST2	SST5	PC	MR
CNN	78.64	40.66	89.84	68.16
+ Ours	79.30	41.97	90.04	71.54
RNN	76.26	36.64	91.99	71.13
+ Ours	78.33	39.62	92.79	74.25
BERT	89.61	50.42	93.65	84.05
+ Ours	91.25	51.13	95.56	85.52

TABLE I
PERFORMANCE FOR THREE CLASSIFICATION MODELS WITH AND WITHOUT AUGMENTED DATA.

	IMDB	DBpedia	Yelp Full
w/o Aug	91.95	99.22	65.52
w/ Aug	92.27	99.30	67.35

TABLE II
PERFORMANCE FOR LONG SEQUENCE DATASET FOR THE BERT MODEL WITH AND WITHOUT AUGMENTED DATA.

For text classification, we used models based on the LSTM-RNN [34] and convolutional neural network (CNN) [35] which are primarily used in text classification. In addition, through the BERT, we confirmed whether the proposed augmentation method improves performance in the CNN, RNN and PLM. We conducted an experiment with the "bert-base-cased" option provided by huggingface [36].

C. Experimental Results

Table I presents the comparison of the text augmentation benchmark data before and after applying the proposed augmentation. Table I reveals that the proposed augmentation method improves the accuracy from about 0.2%p to 2%p. In particular, performance improved by 2.98%p in the SST5 dataset, which contains multi-labeled data. This result confirms that the proposed method accomplished the objective to improve performance. All results in the table represent accuracy.

Regardless of the sentence length, text augmentation should completely preserve the semantics. We tested the proposed method on the IMDB, DBpedia, and Yelp Full datasets to check that the semantics were completely preserved for long sentences. Table II presents the results of the model using the BERT model.

The proposed method improved performance by 0.31%p for the IMDB data and 1.83%p for the Yelp Full dataset, as listed in Table II. The proposed method even works when the sentence length is increased.

D. Ablation Study

We used the VAE instead of the WAE to determine whether posterior collapse actually affects text generation. Table III indicates that the method using the VAE does not perform as well as the proposed method. The VAE method had lower accuracy than when no augmentation was performed on some data. A sentence with distorted label information was generated due to posterior collapse.

Moreover, using the BERT, we conducted an experiment with the "bert-base-cased" option provided by huggingface [36]

	PC	IMDB
Ours	95.23	92.26
w/ VAE	95.16	91.65
w/o Aug	93.65	91.95

TABLE III
PERFORMANCE COMPARING THE SUGGESTED AND VARIATIONAL AUTOENCODER (VAE) MODELS.

	SST2	SST5	PC	MR	IMDB
w/o Aug	89.61	50.42	93.65	84.05	91.95
w/ Transformer (w/o PLM)	90.01	51.59	95.23	85.69	92.26
w/ BERT	90.64	50.61	94.41	84.16	92.00
w/ T5	89.03	49.67	94.91	84.16	92.24
w/ BERT + T5*	90.68	49.34	94.37	84.42	91.73
w/ BART	91.25	51.13	95.56	85.52	92.27

TABLE IV
PERFORMANCE COMPARING THE MODEL USING DIFFERENT PLM AS THE ENCODER AND DECODER. USED BERT, BART, T5, EXPERIMENT WITH THE 'BERT-BASE-CASED', 'FACEBOOK/BART-BASE' AND 'T5-SMALL' OPTION PROVIDED BY HUGGINGFACE REPECTIVELY. *: THE ENCODER IS BERT AND THE DECODER IS T5.

In addition, we evaluated whether the performance changes according to the type of PLMs. We introduced three transformer-based PLMs (i.e., BERT, BART, and T5 [37]) to the proposed method of the encoder and decoder and evaluated whether a change in performance occurred when different PLMs were used as the encoder and decoder. Table IV presents the results of the experiment.

Table IV indicates that the proposed method works well even when various PLMs are introduced. In some tasks, such as in the SST5, models have slightly degraded performance. However, for a fair experiment, this is considered a slight error caused by training with the same epoch in the same environment regardless of the model size. Even when a transformer was used rather than the PLM, good performance was obtained, which proves that the proposed method has sufficient scalability.

V. CONCLUSION

In this paper, we propose using WAE structure for text augmentation. This method is simple but prevents performance degradation of the model due to posterior collapse. And it can be seen that the performance is improved not only when PLM is used, but also when a simple model is used. We improved performance using the proposed method for seven benchmark datasets and demonstrated that this method is scalable by confirming good performance obtained using various PLMs. Future work should use this scalability to conduct research on combining more diverse PLMs in more various way to improve text augmentation.

ACKNOWLEDGMENT

This work was partly supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-01341, Artificial Intelligence Graduate School Program(ChungAng university)).

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

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

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[4] Y. Zhang, A. Warstadt, X. Li, and S. R. Bowman, "When do you need billions of words of pretraining data?," 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)*, pp. 1112–1125, Association for Computational Linguistics, 2021.

[5] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[6] N. Malandrakis, M. Shen, A. Goyal, S. Gao, A. Sethi, and A. Metallinou, "Controlled text generation for data augmentation in intelligent artificial agents," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, Association for Computational Linguistics, 2019.

[7] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," in *Proceedings of ICLR*, 2019.

[8] J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi, "Understanding posterior collapse in generative latent variable models," in *DGS@ICLR*, 2019.

[9] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf, "Wasserstein auto-encoders," in *6th International Conference on Learning Representations (ICLR)*, 2018.

[10] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling, "Do not have enough data? deep learning to the rescue!," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7383–7390, 2020.

[11] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *International Conference on Computer Vision (ICCV)*, pp. 6023–6032, 2019.

[12] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6438–6447, PMLR, 2019.

[13] C. Chadebec, E. Thibeau-Sutre, N. Burgos, and S. Allassonnière, "Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder," *arXiv preprint arXiv:2105.00026*, 2021.

[14] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *International Conference on Learning Representations*, 2018.

[15] S. Y. Feng, V. Gangal, D. Kang, T. Mitamura, and E. Hovy, "GenAug: Data augmentation for finetuning text generators," in *Proceedings of Deep Learning Inside Out (DeeLio): The First workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, Association for Computational Linguistics, 2020.

[16] J. Wei and K. Zou, "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 Inthernational Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6383–6389, Association for Computational Linguistics, 2019.

[17] S. Luu, K. Nguyen, and N. Nguyen, "Empirical study of text augmentation on social media text in vietnamese," in *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pp. 462–470, Association for Computational Linguistics, 2020.

[18] W. Wang, Z. Gan, H. Xu, R. Zhang, G. Wang, D. Shen, C. Chen, and L. Carin, "Topic-guided variational auto-encoder for text generation," 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)*, pp. 166–177, Association for Computational Linguistics, 2019.

[19] J. Jorge, J. Vieco, R. Paredes, J.-A. Sánchez, and J.-M. Benedí, "Empirical evaluation of variational autoencoders for data augmentation.," in *VISIGRAPP (5: VISAPP)*, pp. 96–104, 2018.

[20] H. Bahuleyan, L. Mou, H. Zhou, and O. Vechtomova, "Stochastic wasserstein autoencoder for probabilistic sentence generation," in *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

[21] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, "Conditional bert contextual augmentation," in *International Conference on Computational Science*, pp. 84–95, Springer, 2019.

[22] R. Liu, G. Xu, C. Jia, W. Ma, L. Wang, and S. Vosoughi, "Data boost: Text data augmentation through reinforcement learning guided conditional generation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9031–9041, Association for Computational Linguistics, 2020.

[23] V. Kumar, A. Choudhary, and E. Cho, "Data augmentation using pre-trained transformer models," in *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pp. 18–26, Association for Computational Linguistics, 2020.

[24] M. Bayer, M.-A. Kaufhold, B. Buchhold, M. Keller, J. Dallmeyer, and C. Reuter, "Data augmentation in natural language processing: A novel text generation approach for long and short text classifiers," *arXiv preprint arXiv:2103.14453*, 2021.

[25] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[26] Y. Goldberg, "Assessing bert's syntactic abilities," *arXiv preprint arXiv:1901.05287*, 2019.

[27] A. Yang, Q. Wang, J. Liu, K. Liu, Y. Lyu, H. Wu, Q. She, and S. Li, "Enhancing pre-trained language representations with rich knowledge for machine reading comprehension," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2346–2357, 2019.

[28] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Association for Computational Linguistics, July 2020.

[29] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[30] M. Ganapathibhotla and B. Liu, "Mining opinions in comparative sentences," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 241–248, 2008.

[31] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," pp. 79–86, 2002.

[32] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

[33] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015.

[34] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, p. 2873–2879, 2016.

[35] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Association for Computational Linguistics, 2014.

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

[37] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.