How to use BERT attention for data augmentation?

Overview of mixing augmentation methods for NLP.

Author: **Dominik Lewy**,
Lingaro Group, Warsaw University of Technology

Introduction

The goal of this poster is to present a recently created augmentation method for NLP area (section Attention Mix) based on the idea of mixing training observations and using attention mechanism to guide this process. The poster should additionally build understanding on where those methods originated (section Mixing for CV), what is the starting point for this research (section Mixing for NLP) and give high level understanding of the attention mechanism (section Bert attention). At the end of the poster (section Empirical evaluation on SST dataset) an empirical evaluation of the method is also presented alongside an attempt at explaining its effectiveness via part of speech analysis.

Augmentation for NLP

The high level groups of augmentation methods for Natural Language Processing (NLP) are presented in Figure 1. When it come to relatively simple rule-based augmentation methods, the selection of those for NLP area is very limited. All simple augmentation for NLP are described in [15] and visualized in left part of Figure 2. This is especially evident compared to augmentation for Computer Vision (CV) area where there is a lot of relatively simple and label preserving augmentations like rotation, cropping, color and brightness changes or blurring. Probably due to lack of options for NLP, a lot of effort of augmentation techniques for NLP development was focused on model based augmentations. Right part of Figure 2 shows one of those, back translation, which is translation of a sentence using model, to other language and back to original one in hope to get different wording with meaning preserved. Other type of model based augmentations is paraphrasing. This work will however mainly focus on Interpolation based augmentation a new group of augmentation for NLP inspired by CV area.

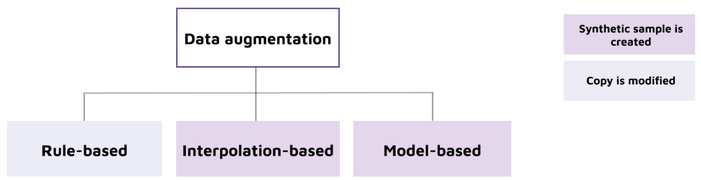

Figure 1. Different groups of data augmentation techniques for Natural Language Processing (NLP). Methods are additionally divided into those that transform copy of existing observation (green) vs methods that created new synthetic sample based on one or more existing observations (yellow).

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
SR	A ***lamentable***, superior human comedy played out on the ***backward*** road of life.
RI	A sad, superior human comedy played out on ***funniness*** the back roads of life.
RS	A sad, superior human comedy played out on ***roads*** back ***the*** of life.
RD	A sad, superior human out on the roads of life.

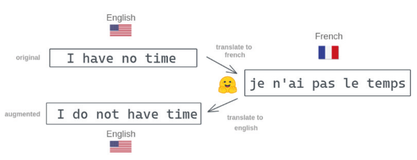

Figure 2. Left sub-figure) Examples of rule based data augmentation: SR - synonymreplacement, RI - random insertion, RS - random swap, RD - random deletion. Source: [15]. (Right sub-figure) Example of back translation a model-based data augmentation. Source: https://amitness.com/back-translation/.

Mixing for CV

Although the idea of mixing observations for structured data has been known for some time, it's application to unstructured data was popularized in CV area. A canonical method that started the research on image mixing as a form of data augmentation is Mixup [17], that linearly interpolates two images and its corresponding labels. There are other methods that perform mixing in a guided manner, i.e. by means of identifying the most relevant parts of the image, e.g. [14] which uses information coming from a CNN classifier, [12] that applies statistical approach, or [5] which, in the mixing process, utilizes the neural network gradients. A review of mixing based data augmentation techniques for image classification is presented in the recent survey paper [8]. Figure 3 presents a map of the mixing methods, indicating for each of them the publication date, certain key characteristics and relations to other methods. The key characteristic differentiating mixing augmentation methods is whether method mixes images using pixel-wise weighted average (referred to as pixel-wise mixing) or mixes images spatially by means of extracting patches from different images and joining them together (referred to as patch-wise mixing methods). Examples from both classes are presented in Figure 4.

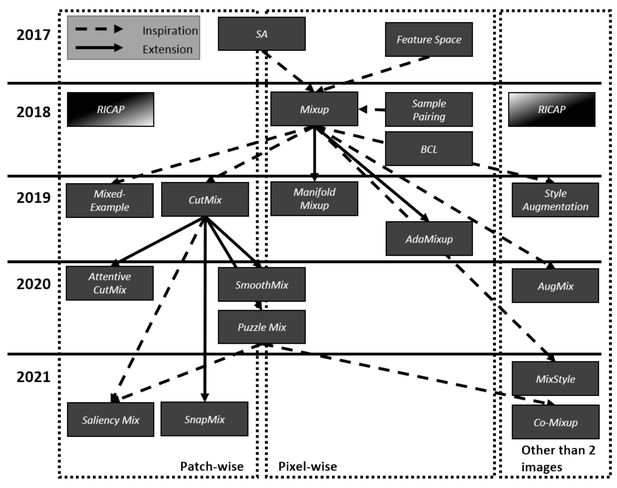

Figure 3. Image mixing DA methods presented on a time scale, with key characteristics and dependencies indicated. Dotted regions separate methods in which mixing takes pixel-wise form (Pixel-wise) from those with spatial mixing (Patch-wise), and those with mixing applied not to a pair of images, but either just one image and its transformed version or more than 2 images (Other than 2 images). Directed lines indicate inspirations (dotted lines) or direct extensions (solid lines) of the methods. Source: [8].

Figure 4. From left to right: two sample images and examples of pixel-wise and patch-wise mixing, respectively. The pixel-wise and patch-wise images present the zoomed region indicated by a red rectangle to show the detailed characteristics of the mixed images. Source: [8].

Mixing for NLP

The above idea, that originated in the CV domain, proven also useful in text classification. [4] utilizes the Mixup idea to perform augmentation of word embeddings and sentence embedding in the training process of CNN and LSTM networks. Subsequent works [6, 11] extend this research by considering BERT architecture and experimenting with Manifold Mixup [13] – a variation of Mixup, which applies mixing to hidden layers of the network. Another example is [16] which utilizes gradient based saliency information to mix the original sentence on a word level. DropMix [7], also utilizes gradient based saliency information, but additionally combines mixing with dropout mechanism to obtain a mixed sample. All the above methods verify whether the mechanisms that have already been successful in CV can be effectively applied to NLP, albeit with certain domain-specific adjustments (e.g. changing a sentence into an embedding that can be mixed [4, 6] or summing gradient based saliency information at the word level [16]). Contrary to the above approaches that rely on improvements rooted in the CV domain, our method [9] is the first to use the guidance stemming from the text-specific mechanism, i.e. attention.

BERT attention

BERT architecture [3] consists of multiple attention layers, each of them containing multiple attention heads. An attention head takes as input a sequence of embeddings $x = [x_1, ..., x_n]$ corresponding to n tokens of the input sentence. Those embeddings (x_i) are transformed into query, key, and value vectors (q_i, k_i, v_i) using Q, K and V matrices learned during the training process for each attention head separately. Each head computes attention weight between all pairs of tokens according to Eq. 1. The above stands for a softmax-normalized dot product between the query and key vectors. An exemplary calculation of attention value and embedding of the next layer is presented in Figure 5

$$\alpha_{ij} = \frac{exp(q_i^T k_j)}{\sum_{l=1}^{n} exp(q_i^T k_l)} \quad (1)$$

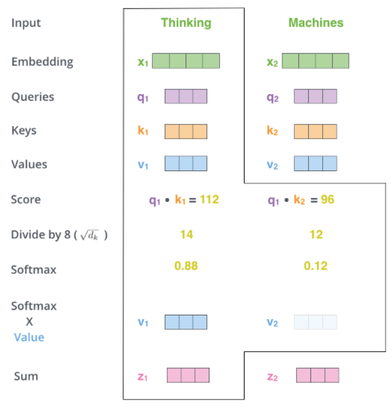

Figure 5. Detailed example of how attention is calculated and used to arrive at the embedding of the token in the next layer. Source: http://jalammar.github.io/illustrated-transformer/

Mixing in BERT

Mixing in BERT can be applied on various levels of the network:

- At the word embedding level – Figure 6 (left)
- At the word encoding level – Figure 6 (middle)
- At the sentence embedding level – Figure 6 (right)

The AttentionMix method that we propose follows the first implementation concept

Figure 6. AttentionMix at the word-embedding level proposed in this paper (left subfigure) and two Mixup schemes: at the word-encoding level (middle) and at the sentence-encoding level (right).

Attention Mix

AttentionMix aims to utilize the information coming from attention heads (Eq. 1) to guide the mixing process. Attention information is relevant and applicable only on the word embedding and word encoding levels. Furthermore, we focus on the augmentation at the word embedding level since the working hypothesis is that utilizing attention closer to the input and prior to the encoding stage (i.e. learning the context of each token and adjusting its embedding based on that) will lead to higher model's accuracy.

Let's consider L attention layers with H heads each. Then, for each head $h \in H$ in layer $l \in L$ and each sentence S the attention weight matrix $AW_{hl}(S)$ has the form:

$$AW_{hl}(S) = [\alpha_{ij}]_{n \times n} \quad (2)$$

where n is the number of tokens in a sentence. α_{ij} represents the impact of token α_{ij} on the next layer representation of the current token. Based on $AW_{hl}(S)$, we calculate the relevance of each token in the sentence from the perspective of a single head (Eq. 3) and the mean from all heads in a single layer (Eq. 4).

$$B_{head_{hl}} = \frac{\sum_i \alpha_{ij}}{n} \quad (3)$$

$$B_{layer_l} = \frac{\sum_{h=1}^{H} B_{head_{hl}}}{H} \quad (4)$$

The above relevance of each token in a sentence is calculated for each observation (sentence) in the training dataset. For two pairs of (sentence, label): (x_1, y_1) and (x_2, y_2) the equations for creating a mixed sentence are as follows:

$$\lambda_{vector} = \frac{B^1}{B^1 + B^2} \quad (5)$$

$$\tilde{x} = \lambda_{vector} \odot x_1 + (1 - \lambda_{vector}) \odot x_2 \quad (6)$$

$$\lambda_{label} = \frac{\sum \lambda_{vector}}{|\lambda_{vector}|} \quad (7)$$

$$\tilde{y} = \lambda_{label} \cdot y_1 + (1 - \lambda_{label}) \cdot y_2 \quad (8)$$

where B_1 and B_2 are the relevance vectors, calculated using either Eq. 3 or Eq. 4, for observations (x_1, y_1) and (x_2, y_2), respectively. λ_{vector} is the mixing ratio vector used for token embedding mixing, λ_{label} is the mixing ratio used to mix one-hot-encoded labels, and $|\lambda_{vector}|$ is the number of token relevance values. λ_{vector} represents the importance of each individual token in a sentence and λ_{label} is a single value (the mean of all λ_{vector} elements) that is defines the relative degree to which each of the two one-hot-encoded vectors of labels contributes to the calculation of $\tilde{y}$ (Eq. 8).

Empirical evaluation on SST dataset

SST – is the Stanford Sentiment Treebank dataset [10] with fine-grained 5-level sentiment scale. Note that in the literature, a simplified binary version of this data set is also considered. In the experiments, we chose the original non-binary setting with a 5-point sentiment scale.

We compare AttentionMix with three baselines: (1) standard BERT training without Mixup, referred to as vanilla approach, (2) adaptation of wordMixup [4], and (3) a special case of TMix [1], which we refer to as MixupEncoding. The main difference compared to AttentionMix is that both reference Mixup-like augmentation methods do not use the guidance coming from the attention mechanism. Additionally, [4] uses LSTM or CNN architecture instead of BERT, and the embeddings are utilized at the word level, not the token level. MixupEncoding compared to TMix [1] performs mixing after the BERT entire encoder, not at a randomly chosen hidden layer.

Table 1. SST dataset. Comparison of the average results of 3 benchmark methods and AttentionMix. For AttentionMix, two attention levels are considered: the layer level – (all heads from a layer) and the head level – (a single head within all layers). In each case, the results of the 3 best performing configurations are presented. The details are depicted in Figure 7. All experiments were repeated 3 times.

Approach	Attention		Accuracy	
	layer	head	mean	std
standard training	–	–	51.17	0.97
wordMixup	–	–	51.60	0.18
MixupEncoding	–	–	51.30	1.13
AttentionMix	10	all	52.05	0.86
AttentionMix	0	all	51.78	0.23
AttentionMix	2	all	51.45	0.21
AttentionMix	10	3	52.76	0.58
AttentionMix	0	8	52.62	0.21
AttentionMix	0	0	52.37	0.26

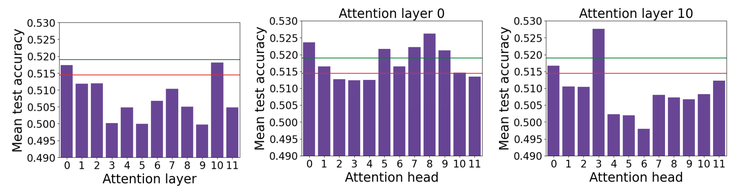

Figure 7. SST dataset. The average test accuracy for experiments utilizing mean attention from: all heads in a layer (Eq. 4) – left subfigure, single heads (Eq. 3) in layer 0 – middle subfigure, single heads (Eq. 3) in layer 10 – right subfigure. The red line indicates the vanilla BERT accuracy, and the green one, the best accuracy achieved by the Mixup benchmark methods. All experiments were repeated 3 times.

The results for the SST dataset are presented in Table 1. The vanilla BERT method reached 51.17% and was inferior to both other benchmarks that utilize Mixup in the training process (wordMixup and MixupEncoding). In all AttentionMix experiments presented in Table 1, higher mean accuracy than the vanilla approach was achieved, and in all but one of them the AttentionMix results exceeded all 3 benchmark approaches. More detailed results are depicted in Figure 7. The left subfigure presents the average accuracy when all heads in a given layer are utilized, and the middle and right subfigures are deep dives into the average accuracy when single heads in the top performing layers (0 and 10, respectively) are considered.

The results presented in Figure 7 show that the use of other than top-3 attention layers in the augmentation process clearly deteriorates the accuracy on the test set. Furthermore, when looking at the middle and right subfigures, the highest result among individual heads is achieved by a head from layer 10, the learning in layer 0 is more uniform and there are more "strong" heads in this layer. In layer 0 the use of any individual head results in higher accuracy than the standard BERT training and for 6 out of 12 heads it excels all competitive methods.

On the contrary, a closer look at layer 10 (Figure 7 (right)) shows that there are only 2 heads (0 and 3) with the results higher than all benchmarks and just 3 (0, 3 and 11) with the accuracy higher than the standard BERT training. We further investigated why certain information coming from attention weight matrices results in higher accuracy boost. We hypothesized that certain parts of speech may have higher impact on sentiment classification than others. Specifically, our assumption was that adjectives, adverbs, and verbs could possibly be more indicative for the sentiment class prediction, since the sentiment is usually reflected by the statements like:

- love, like, hate – verbs
- fantastic, disappointing – adjectives
- quite, very, extremely – adverbs

For the SST dataset, this hypothesis was confirmed only for some relevance vectors derived from attention information. Figure 8 shows the mean attention given to a certain part of speech by attention head 8 in layer 0 and attention head 3 in layer 10, whose usage resulted in the two best performing models. For head 8 in layer 0 indeed high attention is given to adjectives, adverbs, and verbs, but for attention head 3 in layer 10 very high attention is given to punctuation. This phenomenon of high punctuation-related attention has been previously observed by [2].

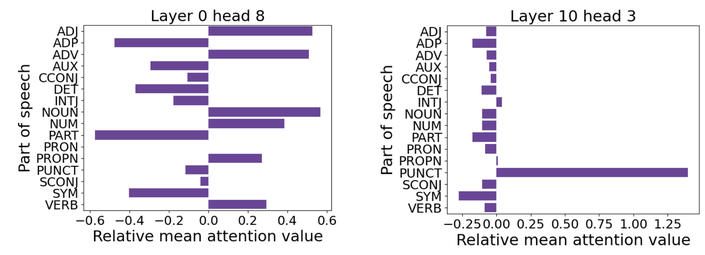

Figure 8. SST dataset. The mean attention value assigned to each part of speech for a given attention head, relative to the mean attention value of the head after training on SST. The abbreviations stand for: ADJ – adjective, ADP – adposition, ADV – adverb, AUX – auxiliary, CONJ - conjunction, CCONJ - coordinating conjunction, DET – determiner, INTJ – interjection, NOUN – noun, NUM - numeral, PART – particle, PRON - pronoun, PROPN - proper noun, PUNCT - punctuation, SCONJ - subordinating conjunction, SYM - symbola and VERB - verb.

References

[1] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 2147–2157, 2020.

[2] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of bert's attention. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@ACL 2019, Florence, Italy, pages 276–286, 2019.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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, NAACL-HLT 2019, Minneapolis, MN, USA, pages 4171–4186, 2019.

[4] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. CoRR, abs/1905.08941, 2019.

[5] Shaoli Huang, Xinchao Wang, and Dacheng Tao. Snapmix: Semantically proportional mixing for augmenting fine-grained data. CoRR, abs/2012.04846, 2020.

[6] Amit Jindal, Dwaraknath Gnaneshwar, Ramit Sawhney, and Rajiv Ratn Shah. Leveraging BERT with mixup for sentence classification student abstract). In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, pages 13829–13830, 2020.

[7] Fanshuang Kong, Richong Zhang, Xiaohui Guo, Samuel Mensah, and Yongyi Mao. Dropmix: A textual data augmentation combining dropout with mixup. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 890–899. Association for Computational Linguistics, 2022.

[8] Lewy D. Mańdziuk J. An overview of mixing augmentation methods and augmentation strategies. Artif Intell Rev, 2022.

[9] Lewy D. Mańdziuk J. Attentionmix: Data augmentation method that relies on bert attention mechanism. 2023.

[10] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, Seattle, Washington, USA, pages 1631–1642, 2013.

[11] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S. Yu, and Lifang He. Mixup-transformer: Dynamic data augmentation for NLP tasks. In Donia Scott, Núria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 3436–3440. International Committee on Computational Linguistics, 2020.

[12] A. F. M. Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. Saliencymix: A saliency guided data augmentation strategy for better regularization. CoRR, abs/2006.01791, 2020.

[13] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better Representations by Interpolating Hidden States. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA, volume 97, pages 6438–6447, 2019.

[14] Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. Attentive Cutmix: An Enhanced Data Augmentation Approach for Deep Learning Based Image Classification. In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, pages 3642–3646, 2020.

[15] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 6381–6387. Association for Computational Linguistics, 2019.

[16] Soyoung Yoon, Gyuwan Kim, and Kyumin Park. Ssmix: Saliency-based span mixup for text classification. In Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, pages 3225–3234, 2021.

[17] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 2018.