Self-Augmentation via Self-Reweighting: Unlocking Intrinsic Potential of Language Models for Cross-Encoded Conditional Semantic Textual Similarity Measurement

Anonymous ACL submission

Abstract

Conditional Semantic Textual Similarity (C-STS) introduces specific limiting conditions to the traditional Semantic Textual Similarity (STS) task, posing challenges for various mainstream models. Language models employing cross-encoding demonstrate satisfactory performance in STS, yet their effectiveness significantly diminishes in C-STS. In this work, we argue that the failure of cross-encoding language models in C-STS is not due to their inability to extract effective features, but rather because they extract an excessive number of features, thereby diluting the impact of condition-relevant features. To alleviate this, we propose *Self-Augmentation via Self-Reweighting*, which does not require the introduction of any external auxiliary information. Instead, it amplifies the impact of condition-relevant features and suppresses condition-irrelevant features through model's intrinsic information. The self-reweighted outputs are used as a self-augmentation signal to enhance the model's original outputs. On the C-STS test set, our proposed method consistently improves the performance of all fine-tuning baseline models (up to around 3 points). Remarkably, it even enables models with smaller parameter scales to surpass the performance of zero-shot and few-shot prompted large language models (such as GPT-4) with substantially larger parameter scales.

1 Introduction

Semantic textual similarity (STS) has been a cornerstone task in NLP for years(Agirre et al., 2014, 2015, 2016; Cer et al., 2017; Abdalla et al., 2021), which is to measure the semantic similarity between two sentences. With the emergence of pre-trained language models, such as BERT(Devlin et al., 2018), RoBERTa(Liu et al., 2019), GPT-3(Brown et al., 2020) and T5(Raffel et al., 2020), etc., the STS task seems to have been almost solved. However, STS is an inherently ambiguous task(Wang et al., 2023), for the varying aspects that can influence sentence similarity, unconditionally measuring this similarity is irrational and unexplainable. To solve the ambiguity of STS task itself, Deshpande et al. (2023) proposed a novel task called conditional semantic textual similarity (C-STS), which incorporates specific conditions to highlight aspects of interest in sentence pair similarity assessment, enables a more grounded, precise and multi-faceted evaluation.

Given that C-STS introduces additional complexity into STS, researchers have explored various encoding strategies, including cross-encoder(Liu et al., 2019), bi-encoder(Reimers and Gurevych, 2019), and tri-encoder(Deshpande et al., 2023). However, the results obtained have been less than satisfactory. The current state-of-the-art models on STS tasks, such as SimCSE(Gao et al., 2021) can only achieve relatively low performances on C-STS, even large language models with few-shot prompts perform poorly on C-STS task.

As noted in the previous study, pre-trained language models have already gained the ability to capture most kinds of potential semantic information in sentences effectively(Rogers et al., 2021; Gessler and Schneider, 2021; Vig, 2019; Clark et al., 2019; Hewitt and Manning, 2019; Davison et al., 2019; Petroni et al., 2019; Wang et al., 2020). Accordingly, in this paper, we argue that the reason they do not perform well on C-STS is that they attend to excessive semantic information, resulting in the introduction of numerous condition-irrelevant features when measuring similarity through simple cross-encoding, which in turn dilutes the impact of condition-relevant features, namely, dilution effect.

To address this issue, we need to seek a method capable of selectively capturing salient features based on the condition. Such tasks are more common in vision and multimodal fields. Previous work(Mirza et al., 2019; Lu et al., 2017; Yang et al., 2016; Jaegle et al., 2021; Shi et al., 2023) in these domains has also yielded effective results by inte-

Method	Encoder Type	#CM	#FF	Reweight	Main Field
Vanilla LMs (Gao et al., 2021)	cross-encoder	1	1	✗	text
PerceiverIO (Jaegle et al., 2021)	cross-encoder	3	1	✓	multimodal
AbSViT (Shi et al., 2023)	bi-encoder	2	2	✓	vision
Self-Augmentation (Ours)	**cross-encoder**	**1**	**1**	✓	**text**

Table 1: Comparison of related work. "#CM" and "#FF" represent the number of computational modules required for a single feedforward pass and the number of feedforward passes needed for one prediction, respectively.

grating modules that calculate similarities between input objects and specified conditions, utilizing these scores to reweight the outputs for prediction, which effectively adjusts the distribution of salient regions in the model's attention maps to make the model focus more on specific objects with higher similarity to the conditions, thereby reducing the interference of other objects during prediction.

Inspired by the "reweighting" strategy, to alleviate the dilution effect mentioned above, we propose a method provides a stronger guide signal for fine-tuning language model, further exploiting the intrinsic potential of language models to solve the C-STS task. We combine the reweighted signal with the original output using a specific scale factor, making the condition-relevant features contribute more when predicting. Given that the correlation information used for reweighting is directly derived from the last-layer attention computed in the feedforward pass, we refer to this as *Self-Augmentation via Self-Reweighting*, eliminating the need to introduce external auxiliary information, thereby making the fine-tuning process more efficient.

Retaining an architecture that is relatively consistent with that of the pre-trained language model, our proposed method exhibits the capability to outperform the fine-tuning baselines on the C-STS test set. Remarkably, with a significantly smaller parameter scale, it also surpasses the performance of most zero-shot and few-shot prompted large language models, highlighting its significant potential in advancing C-STS measurement.

2 Related Work

Pre-trained Language Model. There is substantial evidence indicating that throughout the pre-training, language models learn not only contextualized text representations, but also a grasp of grammar(Vig, 2019), syntax(Hewitt and Manning, 2019), even commonsense(Davison et al., 2019) and world knowledge(Petroni et al., 2019; Wang et al., 2020). This multifaceted learning underscores the depth and breadth of understanding that language models achieve during pre-training.

In this paper, we adopt this idea and argue that the poor performance of current language models on C-STS tasks can be attributed to the models' focus on excessive amount of such semantic information across multiple condition-irrelevant aspects during similarity measurement employing cross-encoding, thereby diluting the essential correlation between sentence pairs and the conditions, ultimately leading to suboptimal performance.

Conditional Reweighted Feedforward. Tasks similar to C-STS (Deshpande et al., 2023) find more common application in fields like vision (e.g., multi-object image recognition(Deng et al., 2009)) and multimodal tasks (e.g., visual question answering(Antol et al., 2015; Carrasco, 2011; Li, 2014)). In these contexts, a specific condition is essential for directing the model's focus towards objects that are relevant to the given condition.

Previous work employing such methods has yielded effective results. PerceiverIO(Jaegle et al., 2021) introduced multiple cross-attention modules to compute the relevance to reweight the output tokens, which were directly used for prediction. Conversely, AbSViT(Shi et al., 2023) proposed a feedback mechanism to feed the relevance computed during the first feedforward phase back to the preceding modules, then the second feedforward were conducted for prediction.

Inspired by previous work, we adapt the "reweighting" strategy to C-STS. As shown in Table 1, compared to PerceiverIO, our method eliminates the need for multiple modules, simplifies the workflow, and achieves a higher degree of model integration. And compared to AbSViT, our method eliminates the feedback modules and only reweights the final output, which maintains the consistency of the pre-trained language models, making the training process more efficient.

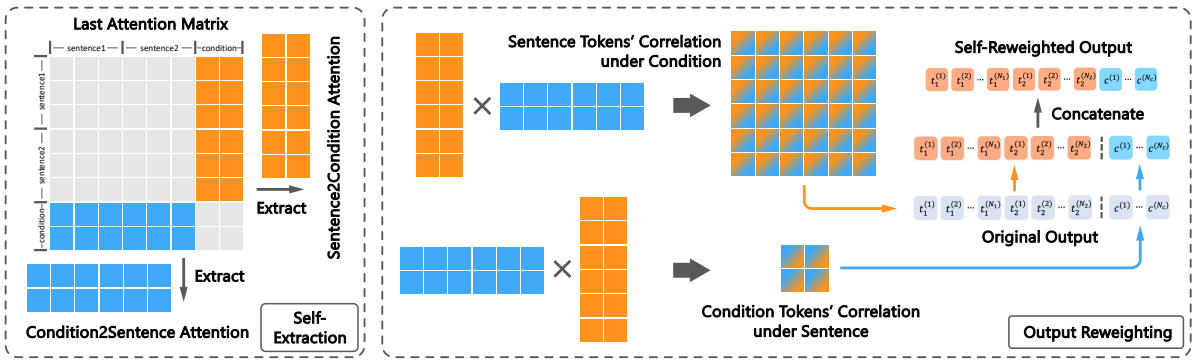


Figure 1: Self-Reweighting flow (from left to right). (i) Self-Extraction: extract own attention sub-matrix, which represents the interaction between the sentence and the condition. (ii) Output Reweighting: compute correlation matrices, serving to reweight the original output of the sentence and the condition, respectively, then concatenate them, culminating in the acquisition of a self-reweighted output.

3 Method

This section starts with *self-reweighting*, which directly extracts correlation information between sentences and the conditions to reweight the outputs (Section 3.1), then we use the reweighted outputs to enhance the original outputs in a specific proportion (Section 3.2), namely *self-augmentation*.

3.1 Self-Reweighting

When utilizing cross-encoding, we compute the attention matrix of concatenated sentence pair and the condition. The resulting attention matrix actually encapsulates multi-faceted information, encompassing both the self-attention of each input item and the cross-attention among input items.

To utilize the condition-relevant information, as shown in Fig. 1, we specifically extract the cross-attention between the sentences and the conditions from the whole attention matrix. Then we divide them into two distinct aspects of attention, namely *Sentence2Condition Attention* (abbreviated as **SCAttn**) and *Condition2Sentence Attention* (abbreviated as **CSAttn**), respectively. Here, $\mathbf{SCAttn} \in \mathbb{R}^{l_s \times l_c}$ and $\mathbf{CSAttn} \in \mathbb{R}^{l_c \times l_s}$, where l_s indicates the length of the concatenated sentence pair, and l_c indicates the condition length.

We use the extracted **SCAttn** as the condition-guided signal for the concatenated sentence pair and **CSAttn** as the sentence-guided signal for the condition. Utilizing these, we construct the reweighting matrices for the sentences and the conditions, respectively, which are computed as

$$\mathbf{W}_S = \text{softmax}(\mathbf{SCAttn} \cdot \mathbf{CSAttn}) \quad (1)$$

$$\mathbf{W}_C = \text{softmax}(\mathbf{CSAttn} \cdot \mathbf{SCAttn}), \quad (2)$$

where $\mathbf{W}_S \in \mathbb{R}^{l_s \times l_s}$ indicates the reweighting matrix for the sentence pair and $\mathbf{W}_C \in \mathbb{R}^{l_c \times l_c}$ indicates the reweighting matrix for the condition.

Applying the obtained reweighting matrices $\mathbf{W}_S$ and $\mathbf{W}_C$, we perform self-reweighting on the truncated model outputs. This allows us to obtain the reweighted outputs of both the sentence pair and the condition parts, which can be computed as

$$\mathbf{RO}_S = \mathbf{W}_S \cdot \mathbf{O}[t_1^{(1)}, ..., t_1^{(N_1)}; t_2^{(1)}, ..., t_2^{(N_2)}] \quad (3)$$

$$\mathbf{RO}_C = \mathbf{W}_C \cdot \mathbf{O}[c^{(1)}, ..., c^{(N_c)}], \quad (4)$$

where $\mathbf{O} \in \mathbb{R}^{l \times d}$ indicates the last hidden state of the language model, which we subsequently refer to as the original output in the following text. l and d represent the length of the concatenated input (comprising the sentence pair and the condition) and the dimension of the language model's hidden state, respectively. Here we represent the i-th token of sentence k ($k \in \{1, 2\}$) as $t_k^{(i)}$. $\mathbf{RO}_S \in \mathbb{R}^{l_s \times d}$ and $\mathbf{RO}_C \in \mathbb{R}^{l_c \times d}$ represent the reweighted output of the sentence pair and the condition, respectively.

After acquiring the reweighted outputs for both the sentence pair and the condition, we then concatenate them to form the concatenated reweighted output, as shown below:

$$\mathbf{RO} = [\mathbf{RO}_S; \mathbf{RO}_C], \quad (5)$$

where $\mathbf{RO} \in \mathbb{R}^{l \times d}$ indicates the concatenated reweighted output, which is of the same size with the original output $\mathbf{O}$. Then, we utilize the obtained concatenated reweighted output $\mathbf{RO}$ as an augmentation signal to perform the self-augmentation as described in Section 3.2.

Furthermore, it is important to note that the reweighting matrices are derived directly from the attention matrices returned by the last layer of the language model. Since this does not introduce an external information, we refer to this process as *self-reweighting*.

3.2 Self-Augmentation

We consider the multi-head self-attention mechanism of the language model, which ultimately yields H attention matrices, where H is the number of attention heads. Here, we refer to the reweighted output obtained after applying the reweighting matrices constructed from the attention matrix returned by the i-th attention head as $\mathbf{RO}_i$. Following a method similar to that used in Transformers for processing outputs from multiple attention heads(Vaswani et al., 2017), we concatenate these H reweighted outputs. Subsequently, they are projected through a projection matrix to match the dimension of a single reweighted output, which can be computed as

$$\mathbf{RO} = [\mathbf{RO}_1; \mathbf{RO}_2; ...; \mathbf{RO}_H] \cdot \mathbf{W}_o, \quad (6)$$

where $\mathbf{W}_o \in \mathbb{R}^{Hd \times d}$ indicates the projection matrix. To be more specific, the $\mathbf{RO}$ here indicates the projected reweighted output. Each $\mathbf{RO}_i$ is computed through Eq. 5, where it should be noted that the $\mathbf{RO}$ in Eq. 5 denotes the case for a single attention head.

We utilize the final reweighted output $\mathbf{RO}$ as an augmentation signal, aimed at enhancing parts of the original output $\mathbf{O}$ where there is a significant semantic association between the sentence pair and the condition. To achieve this, we perform a weighted addition of the augmentation signal $\mathbf{RO}$ with the original output $\mathbf{O}$. This results in the self-augmented output, which is then utilized for predicting similarity, which can be computed as

$$\mathbf{AO} = \mathbf{RO} + \alpha \mathbf{O}, \quad (7)$$

where $\mathbf{AO} \in \mathbb{R}^{l \times d}$ indicates the self-augmented output and $\alpha \geq 0$ denotes the hyperparameter that controls the ratio between the weight of reweighted output $\mathbf{RO}$ and the original output $\mathbf{O}$, which is discussed in detail in Section 4.2.

The overall architecture of the model is as depicted in Fig. 2, where the final regressor connected behind the pre-trained language model is a single-hidden-layer MLP structure. It is important to note

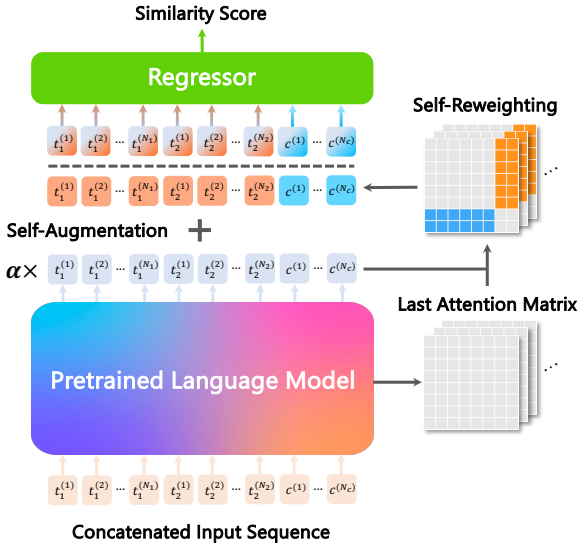


Figure 2: Overall architecture of our Self-Augmentation model. A self-augmented output is derived through the addition of the self-reweighted output to the original output(scaled by a factor of α). This self-augmented output is subsequently fed into a regressor (a single-hidden-layer MLP), predicting the semantic similarity.

that since the augmentation signal is directly derived from the attention matrix computed by the language model itself, and no external augmentation information is introduced in this process, we refer to this as *self-augmentation*.

4 Experiments

In this section, we first demonstrate that the self-reweighting operation can be conceptualized as a *soft mask* mechanism, which amplifies the parts of the output where the sentence pair and the condition are highly related, while suppressing the parts where the relevance is low (Section 4.1). Then we provide a comprehensive quantitative analysis, discussing how the combined augmentation signal and original signal at varying ratios influence the model's final predictive behavior (Section 4.2).

Dataset. In this study, we employ C-STS-2023 dataset collected by Deshpande et al. (2023) for training and testing, which consists of quadruples, formatted as (`sentence1`, `sentence2`, `condition`, `label`). In which `sentence1`, `sentence2` and `condition` are all presented in natural language form, and `label` represents the level of similarity between `sentence1` and `sentence2` under `condition`, converted into a Likert scale(Likert, 1932) with values ranging from 1 to 5, which is common with semantic tex-

Sentence 1	Sentence 2	Condition	Output
A boy is in midair doing a skateboard trick at a skate park while two women and a toddler walk behind him.	A boy in yellow pants and a blue shirt is rollerblading on the side of his black skates.	The type of skating.	w/o: 4.00 w/ : 1.46 Label: 1.00
Two people are near a wooden building wearing backpacks.	A couple of people working around a pile of rocks.	The number of people.	w/o: 2.60 w/ : 4.62 Label: 5.00

Table 2: Two cases from the C-STS validation set. "Output" refers to the predicted and the ground-truth similarity, where the notation "w/o" represents the prediction from the baseline model, and "w/" denotes the prediction from our proposed Self-Augmentation model (based on RoBERTa-base). More cases are available in Appendix A.1.

tual similarity tasks(Agirre et al., 2013).

Experimental Setup. We conduct a comparative analysis between various baselines and our proposed method, which can be categorized into:

(i) **Fine-tuning** baselines, which are fine-tuned on the entire training partition. We select RoBERTa(Liu et al., 2019) and SimCSE(Gao et al., 2021) as our baselines, encompassing both the `base` and `large` scale models.

(ii) **Prompting** baselines, which refer to general-purpose large language models, are recognized for their zero-shot or few-shot learning capabilities. For a comprehensive performance analysis, we select Flan-T5(Wei et al., 2021) (in both `base` and `large` configurations), GPT-J(Wang and Komatsuzaki, 2021), GPT-3.5(Brown et al., 2020), and GPT-4(Achiam et al., 2023) as our baselines.

It is important to note that due to observed variances in experimental results across different models of GPUs, to ensure reproducibility, all experiments were conducted on a single RTX A5000. More details are available in Appendix A.2.

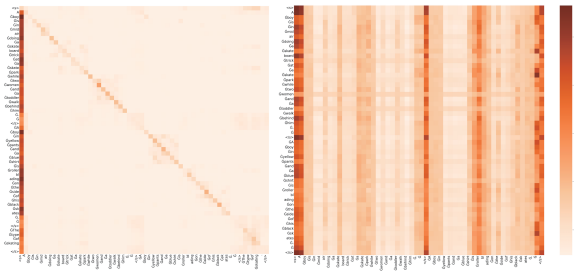
4.1 Correlation Dilution Effect and Self-Reweighting Alleviation

From the Table 2, it is observed that, moving from top to bottom, for the first case, the predictions made by the baseline model are higher in comparison to the ground-truth. This intuitively suggests that, within the relevant features captured by the baseline model, `sentence1` and `sentence2` exhibit a higher semantic similarity under the `condition`. Conversely, for the second case, the baseline model's predictions are lower relative to the ground-truth. However, the predictions from our proposed Self-Augmentation method align more closely with the ground-truth.

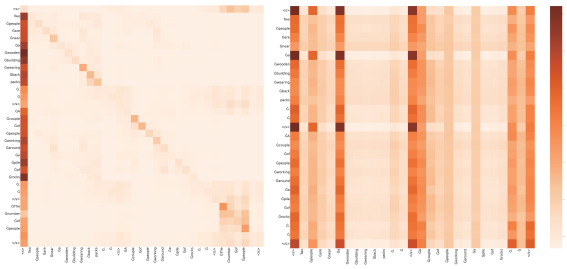
To elucidate the feature capture mechanism of the baseline model in this task, and to understand the reasons behind the baseline model's prediction failures as well as the success of our Self-Augmentation model, we extracted and averaged the multi-headed attention matrices from the last layer of the baseline model and the self-reweighting weights for the sentence part in the Self-Augmentation model. Subsequently, these were visualized for analysis. As illustrated in Fig. 3, this allows us to more intuitively analyze the differing feature capture modes of the models.

It is important to note that, as our objective is to discuss and analyze the model's feature capture patterns, since SimCSE is a model fine-tuned on the STS task, its feature capture preferences might significantly deviate from those of a pre-trained model directly fine-tuned on the C-STS dataset. To avoid ambiguity, we have chosen RoBERTa-base as the model for our case study. This selection allows for a more equitable and lucid analysis of the feature capture patterns of the baseline model and our Self-Augmentation model.

From the average attention matrix of the last layer of the baseline model shown in Fig. 3(a) (left), it is observable that the attention map of the fine-tuned baseline model does not contain any specific salient regions. However, previous studies(Clark et al., 2019) have confirmed that the pre-trained language model should possess the capability to capture multifaceted features. While it is acknowledged that the fine-tuning process may somewhat



(a)



(b)

Figure 3: Average attention matrix(left: obtained from the baseline model) and self-reweighting weight(right: obtained from our proposed Self-Augmentation model) of the first-row case(a) and the second-row case(b) presented in Table 2. The darker the color, the larger the corresponding value.

impair this capability, it appears to be markedly diminished in these cases.

We argue that the reason for the baseline model's predictive failure does not lie in its inability to capture relevant features, but rather due to its excessive capture of condition-irrelevant features, which, after being normalized by the softmax function, dilute the impact of condition-relevant features on the final prediction. This correlation dilution effect, leading to the baseline model's predictive failure, which is also observable in Fig. 3(b) (left).

After applying our proposed Self-Augmentation method, we observe from Fig. 3(a)(right) and Fig. 3(b)(right) that the reweighting weights derived from Self-Reweighting exhibit distinct salient regions (darker in color) and suppressed areas (lighter in color). Notably, the formation of such salient regions is condition-relevant. For instance, for the first case in Table 2, the salient reweighting regions of the reweighting weights concentrate on tokens related to "`the type of skating`", such as "`rollerblade`"; for the second case, the salient regions focus on tokens related to "`the number of people`", such as "`a`" and "`couple`".

The aforementioned further substantiates our hypothesis: the application of our proposed Self-Augmentation method, which successfully enhances condition-relevant feature regions and suppresses condition-irrelevant ones, improves the predictive capability of the model compared to the baseline model. Importantly, since our Self-Reweighting approach for obtaining reweighting weights does not introduce any external enhancement information, it indicates that the model, through pre-training and fine-tuning, has already acquired the capability to extract multifaceted features. However, the simultaneous extraction of an excessive amount of condition-irrelevant features diluted the effectiveness of valid condition-relevant features. The application of our proposed Self-Augmentation method can effectively mitigate this issue, thereby enhancing the performance and stability of the model's predictions.

4.2 Quantitative Results and Analysis

We initially conduct fine-tuning experiments on the entire training partition of the C-STS dataset, utilizing prominent sentence encoders: RoBERTa and SimCSE. We set the range of the scaling factor α in Eq. 7 from 0 to 3, to observe the impact on the overall model performance under different ratios of the self-augmentation signal combined with the original output. The detailed quantitative results of fine-tuning are shown in the Table 3.

RoBERTa has been fine-tuned directly on the C-STS dataset following pre-training. In contrast, before being further fine-tuned on the C-STS dataset, SimCSE has already been fine-tuned on unconditional STS datasets after the pre-training phase. As shown in Table 3, by adjusting α to suit different models, our proposed Self-Augmentation method can bring stable performance improvements to each model of different scales.

As RoBERTa has not been fine-tuned on other STS datasets, it largely retains the multifaceted feature extraction capability acquired during its pre-training phase. Therefore, for the `base` scale RoBERTa model, solely using the self-augmentation signal for prediction (i.e., setting α to 0) can yield its optimal result. Introducing varying degrees of the original output may, to some extent, impair this, leading to suboptimal performance. Conversely, the `large` scale RoBERTa, compared to the `base` scale, further enhances its feature extraction ability. With the increased depth of extracted features, some features suppressed in the self-augmentation signal can positively in-

Model	Scale	Spear. ↑	Pears. ↑
RoBERTa (Deshpande et al., 2023)	Base (125M)	39.07	39.05
Self-Augmented RoBERTa w/o original (Ours)	**Base (132M)**	**41.36**$_{+2.29}$	**41.05**$_{+2.00}$
Self-Augmented RoBERTa w/ 1*original (Ours)	Base (132M)	39.93$_{+0.86}$	39.83$_{+0.78}$
Self-Augmented RoBERTa w/ 2*original (Ours)	Base (132M)	40.44$_{+1.37}$	40.35$_{+1.30}$
Self-Augmented RoBERTa w/ 3*original (Ours)	Base (132M)	38.83$_{-0.24}$	38.91$_{-0.14}$
RoBERTa (Deshpande et al., 2023)	Large (355M)	40.40	40.78
Self-Augmented RoBERTa w/o original (Ours)	Large (372M)	43.16$_{+2.76}$	43.20$_{+2.42}$
Self-Augmented RoBERTa w/ 1*original (Ours)	Large (372M)	40.69$_{+0.29}$	40.56$_{-0.22}$
Self-Augmented RoBERTa w/ 2*original (Ours)	**Large (372M)**	**43.45**$_{+3.05}$	**43.60**$_{+2.82}$
Self-Augmented RoBERTa w/ 3*original (Ours)	Large (372M)	39.35$_{-1.05}$	39.28$_{-1.50}$
SimCSE (Deshpande et al., 2023)	Base (125M)	38.56	39.00
Self-Augmented SimCSE w/o original (Ours)	Base (132M)	37.16$_{-1.40}$	36.92$_{-2.08}$
Self-Augmented SimCSE w/ 1*original (Ours)	Base (132M)	38.48$_{-0.08}$	38.08$_{-0.92}$
Self-Augmented SimCSE w/ 2*original (Ours)	**Base (132M)**	**39.59**$_{+1.03}$	**39.30**$_{+0.30}$
Self-Augmented SimCSE w/ 3*original (Ours)	Base (132M)	39.18$_{+0.62}$	39.24$_{+0.24}$
SimCSE (Deshpande et al., 2023)	Large (355M)	42.28	42.40
Self-Augmented SimCSE w/o original (Ours)	Large (372M)	43.06$_{+0.78}$	43.01$_{+0.61}$
Self-Augmented SimCSE w/ 1*original (Ours)	Large (372M)	42.47$_{+0.19}$	42.52$_{+0.12}$
Self-Augmented SimCSE w/ 2*original (Ours)	Large (372M)	43.70$_{+1.42}$	43.47$_{+1.34}$
Self-Augmented SimCSE w/ 3*original (Ours)	**Large (372M)**	**43.83**$_{+1.55}$	**43.81**$_{+1.41}$

Table 3: Fine-tuning results in Spearman and Pearson correlation coefficient (scaled by 100) on the C-STS test set. Highlighted rows indicate the highest performance achieved within the same model and scale. "Self-Augmented [MODEL NAME] w/ α*original" denotes the addition of the self-augmentation signal to the original output (scaled by a factor of α), and "w/o" is equivalent to the scenario where $\alpha = 0$. More details are available in Appendix A.3.

fluence the prediction (due to increased learned semantic complexity; intuitively, some features may appear condition-irrelevant individually but become condition-relevant in combination), thus introducing a certain degree of the original output (i.e., setting α to 2) can achieve its optimal result.

While SimCSE has already been fine-tuned on unconditional STS datasets, we believe this slightly impairs the model's ability to extract general features. However, SimCSE also acquires effective task-specific features for measuring sentence similarity. There exists a certain trade-off between the negative and positive impacts brought by fine-tuning on the unconditional STS datasets. Intuitively, we suspect this is related to the model's scale. The `base` size SimCSE is more likely to be negatively influenced by fine-tuning on the unconditional STS datasets compared to the `large` scale, resulting in the optimal performance of the `base` size SimCSE model being lower than that of the same scaled RoBERTa. In contrast, the `large` scale SimCSE seems to gain more positive benefits than negative impacts from the unconditional STS fine-tuning process, thereby further enhancing its capability to extract semantic features and achieving higher optimal performance.

To further analyze the impact of different ratios of self-augmentation signal combined with the original output on model performance, we visualized the trend of model performance under various settings of α, as shown in Fig. 4. Both the `base` and `large` scales of the RoBERTa model exhibited similar trends: a significant decrease in performance upon the initial introduction of the original output, followed by a pattern of first increasing and then continuing to decrease as α increases. However, a distinction between the `base` and `large` scales of the RoBERTa model is observed in the performance peak upon increasing the degree of the original output's inclusion: the `large` scale

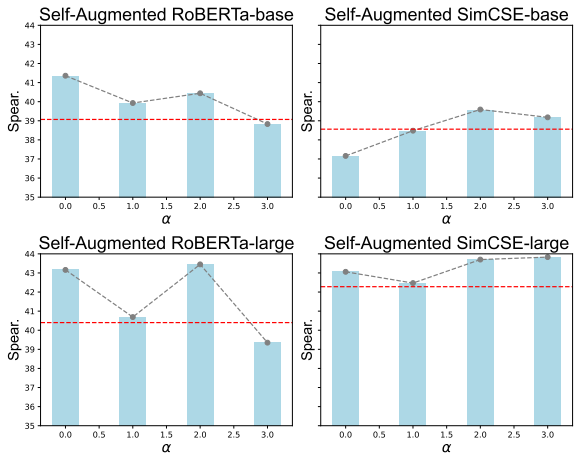


Figure 4: Trends in the Spearman's correlation coefficient of our Self-Augmentation model under different settings of α. The red dashed line represents the performance of the corresponding baseline model.

Model	0-shot ↑	2-shot ↑	4-shot ↑
Flan-T5-base	11.3	9.1	10.7
Flan-T5-large	11.1	12.3	12.8
GPT-J	7.4	1.1	2.0
GPT-3.5	15.0	16.6	15.5
GPT-4	39.3	42.6	43.6
†Self-Augmented SimCSE-large w/ 3*original			
43.8			

Table 4: Zero-shot and few-shot prompted results on the C-STS test set using Spearman's correlation coefficient. † indicates fine-tuning on the entire training partition.

of RoBERTa surpasses the performance of using solely the self-augmentation signal for prediction, whereas the `base` scale does not. The `base` size SimCSE model shows a trend where performance continuously grows to a peak and then declines as α increases. The performance trend of the `large` size SimCSE model is similar to that of RoBERTa, but the peak performance appears to be shifted to the right. It is also observable that at this point, the performance improvement has begun to converge.

The aforementioned trends confirm that, apart from the features directly related to the condition, other features also play a non-negligible role in the overall semantic similarity measurement in most cases. Therefore, this is the rationale for using the self-reweighted output as an augmentation signal to the original output, rather than as the sole component utilized in the final similarity prediction. However, it is also important to note that α increases, the overall performance of the model gradually degrades back to the unenhanced state. The ratio of the self-augmentation signal to the original output also represents a form of trade-off.

Additionally, we compared the performance of our proposed Self-Augmentation method with that of zero-shot and few-shot prompted large language models on the C-STS test set. The performances of the zero-shot and few-shot prompted large language models, as presented in Table 4, represent the best results obtained after prompting using various prompts as applied by Deshpande et al. (2023).

As shown in Table 4, it is evident that despite a substantial difference in the number of parameters between our selected model (372M) and large language models such as GPT-J (6B), GPT-3.5 (175B), and GPT-4 (even larger than GPT-3.5), the best performance of SimCSE-large with our proposed Self-Augmentation method, still surpasses the optimal performance achieved by zero-shot and few-shot prompted large language models. Furthermore, as the process of zero-shot and few-shot prompting in large language models also constitutes cross-encoding, this further confirms the superiority of our proposed method in cross-encoding models.

5 Conclusion

In this work, we argue that language models employing cross-encoding have already acquired the capability to capture multifaceted features during the pre-training phase. The reason for their subpar performance in the C-STS task is attributed to the dilution effect: the multitude of learned features dilutes the impact of condition-relevant features. However, mitigating this dilution through mere fine-tuning is challenging. To address this, we propose *Self-Augmentation via Self-Reweighting*, which does not require the introduction of any external information. Instead, it amplifies the impact of condition-relevant features and suppresses the influence of condition-irrelevant features through model's intrinsic information. The self-reweighted results are then used as an augmentation signal to enhance the model's original output, achieving self-augmentation. On the C-STS test set, our method consistently improves the performance of all fine-tuning baseline models. Notably, it even allows smaller-scale models to surpass the performance of zero-shot and few-shot prompted large language models with substantially larger parameter scales.

Limitations

Although the application of our proposed Self-Augmentation method can bring stable performance improvements to models using cross-encoding, proving its feasibility, due to concerns about the method's complexity, the Self-Augmentation method only involves extracting relevant attention scores from the last layer of the language model and calculating the semantic correlation between sentences and conditions. This results in the extracted relevance reflecting more on the independent semantic features of the last layer, which does not significantly enhance performance.

In this study, experiments have demonstrated that small models applying our proposed method can achieve performance surpassing that of zero-shot and few-shot prompted large language models. However, due to limitations in computational resources, we did not apply our method to larger scale models. And our focus is solely on the text field, without extending it to other fields.

Future work can focus on the comprehensive utilization of semantic features captured in other layers of the model, as well as the combined semantic features of the last layer and other layers. Furthermore, the adoption of a learned adaptive approach to amplify important semantic features of each layer can be considered. This would enable adaptive amplification of a certain number of semantic features according to the complexity of different sentences, thereby achieving more efficiency and satisfactory performance improvements. Additionally, future work should consider extending this method to larger scale models and additional tasks (e.g. multimodal, vision and audio tasks) to explore more of the method's potential.

Ethical Considerations

It is widely acknowledged that language models are capable of generating predictions that exhibit bias. This issue becomes especially pronounced when the input sentences possess sensitive characteristics. While strategies such as data cleaning can alleviate these problems, they do not offer a complete solution. This study advocates for usage under research purposes. Appropriate care should thus be taken when applying such approaches for any non-research purpose (e.g. in user-oriented applications).

In this study, our use of existing artifacts is consistent with their intended purposes. All the datasets and models used in this work are publicly available. RoBERTa-* models have MIT license[1]. Flan-T5-* models have Apache-2.0 license[2]. The remaining open-source models and datasets used, due to the lack of explicit licensing declarations, have all been credited with their sources in Appendix A.2 in this paper.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and

[1] https://choosealicense.com/licenses/mit

[2] https://www.apache.org/licenses/LICENSE-2.0

Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

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. *Advances in neural information processing systems*, 33:1877–1901.

Marisa Carrasco. 2011. Visual attention: The past 25 years. *Vision research*, 51(13):1484–1525.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. 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*, pages 276–286.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Ameet Deshpande, Carlos E Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. 2023. Csts: Conditional semantic textual similarity. *arXiv preprint arXiv:2305.15093*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL).

Luke Gessler and Nathan Schneider. 2021. Bert has uncommon sense: Similarity ranking for word sense bertology. *arXiv preprint arXiv:2109.09780*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. 2021. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*.

Zhaoping Li. 2014. *Understanding vision: theory, models, and data*. Oxford University Press (UK).

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

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*.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383.

M Berk Mirza, Rick A Adams, Karl Friston, and Thomas Parr. 2019. Introducing a bayesian model of selective attention based on active inference. *Scientific reports*, 9(1):13915.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Baifeng Shi, Trevor Darrell, and Xin Wang. 2023. Top-down visual attention from analysis by synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2102–2112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Yuxia Wang, Shimin Tao, Ning Xie, Hao Yang, Timothy Baldwin, and Karin Verspoor. 2023. Collective human opinions in semantic textual similarity. *Transactions of the Association for Computational Linguistics*, 11:997–1013.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29.

A Appendix

A.1 Correlation Dilution Effect and Self-Reweighting Alleviation

Additional cases, along with their corresponding attention matrices and self-reweighting weights, are provided in Table 6 and Fig. 5, respectively. This enables a broader and deeper understanding of the correlation dilution effect and self-reweighting alleviation mentioned in Section 4.1.

It must be reiterated that the self-reweighting weights computed here reflect the modulation of different features' intensities. That is, to enhance condition-relevant features and suppress condition-irrelevant features, it is necessary to adjust the intensity of the original features. Therefore, in the heatmap of self-reweighting weights, there may be instances where the weights of features that are supposed to be enhanced are not as salient. This can occur not only due to the intrinsic learning quality of the model but also because the original intensity of certain features is already relatively strong, thus requiring less enhancement, and vice versa.

A.2 Implementation Details

The hyperparameter settings shown in Table 5 were determined to yield the best performance when evaluating our proposed Self-Augmentation models on the C-STS validation set. To maintain higher consistency with the baseline proposed by Deshpande et al. (2023), and to maximize the reproducibility of our experimental results, we set the torch seed to 42 in all our experiments.

As mentioned by Deshpande et al. (2023), the C-STS-2023 dataset used in this paper comprises a training set (11,342 examples), a validation set (2,834 examples), and a test set (4,732 examples), all consisting of English sentence examples.

All pre-trained parameters of the language models involved in the experiments are directly available on Hugging Face: RoBERTa-base[3], RoBERTa-large[4], SimCSE-base[5], SimCSE-large[6]. For GPT-3.5 and GPT-4, consistent with the experimental setup described by Deshpande et al. (2023), the related test results were obtained using the OpenAI API with the static model versions `gpt-3.5-turbo-0301` and `gpt-4-0314` during the experiments.

Configuration	Base	Large
Batch Size	64	64
Learning Rate	3e-5	1e-5
Weight Decay	0.1	0.1
Seed	42	42
Loss	MSE	MSE

Table 5: Hyperparameter sweep done for C-STS validation for our proposed Self-Augmentation models. "Base" and "Large" represent the scale of our proposed Self-Augmentation models.

A.3 Model Parameter Discussion

In Table 3, we can observe that the parameter count of our Self-Augmentation model has increased slightly compared to the similar scale baseline. This increase is due to the application of a projection matrix that maps the concatenated multi-headed vector dimensions back to the model dimension (the slight increase in parameters corresponds to the introduction of this projection matrix). However, since no external information is introduced and the transformation is applied only to the information originally extracted by the model, our proposed Self-Augmentation method still maintains a relatively high degree of consistency with the original baseline model.

[3] https://huggingface.co/FacebookAI/roberta-base

[4] https://huggingface.co/FacebookAI/roberta-large

[5] https://huggingface.co/princeton-nlp/sup-simcse-roberta-base

[6] https://huggingface.co/princeton-nlp/sup-simcse-roberta-large

Sentence 1	Sentence 2	Condition	Output
Two martial artists compete before a referee and onlookers.	Two people are fighting in full protective gear and helmets.	The number of participants.	w/o: 2.90 w/ : 4.61 Label: 5.00
A man in a black wetsuit rides a surfboard on a wave.	Surfer in black wetsuit falling off his board into the water.	The color of clothing.	w/o: 2.75 w/ : 4.75 Label: 5.00
A man dressed in red dives for a shuttlecock with a racket on a court.	A Japanese man in a red shirt, at the olympics playing tennis.	The name of the color.	w/o: 2.35 w/ : 4.08 Label: 5.00
At a rodeo and a cowboy is riding a bull and other men are standing by.	A man dressed as a cowboy walks away from a brown horse.	The type of animals.	w/o: 3.35 w/ : 1.54 Label: 1.00
A youth on a skateboard is doing flips and tricks over a metal bar.	Young kid in a blue shirt is doing a trick on his rollerblades.	What the person is wearing on their feet.	w/o: 3.07 w/ : 1.28 Label: 1.00
A man with a blue harness climbing a climbing wall.	A young girl wearing a safety harness climbs a rock wall.	The sex of the person.	w/o: 3.37 w/ : 1.66 Label: 1.00
A guy in red shirt is rock-clibbing on a dangerous mountain wall.	A man in a red jacket mountain climbing an icy rock mountain.	The color of clothing.	w/o: 2.18 w/ : 4.12 Label: 5.00
A brown and white dog running fast in a fenced yard.	A dog is running while catching a tennis ball in its mouth.	The action.	w/o: 2.73 w/ : 4.47 Label: 5.00
A boy wearing a green shirt rides a scooter down the sidewalk.	A little boy in a green jacket is crying on his tricycle.	The color of the clothing.	w/o: 2.25 w/ : 4.10 Label: 5.00
A woman in an oversized black shirt plays a black and red guitar in a musky room.	A bass player girl, who is performing at a concert one of the bands songs.	The sex of the musician.	w/o: 2.58 w/ : 4.20 Label: 5.00

Table 6: 10 additional cases from the C-STS validation set. "Output" refers to the predicted and the ground-truth similarity, where the notation "w/o" represents the prediction from the baseline model, and "w/" denotes the prediction from our proposed Self-Augmentation model (based on RoBERTa-base).

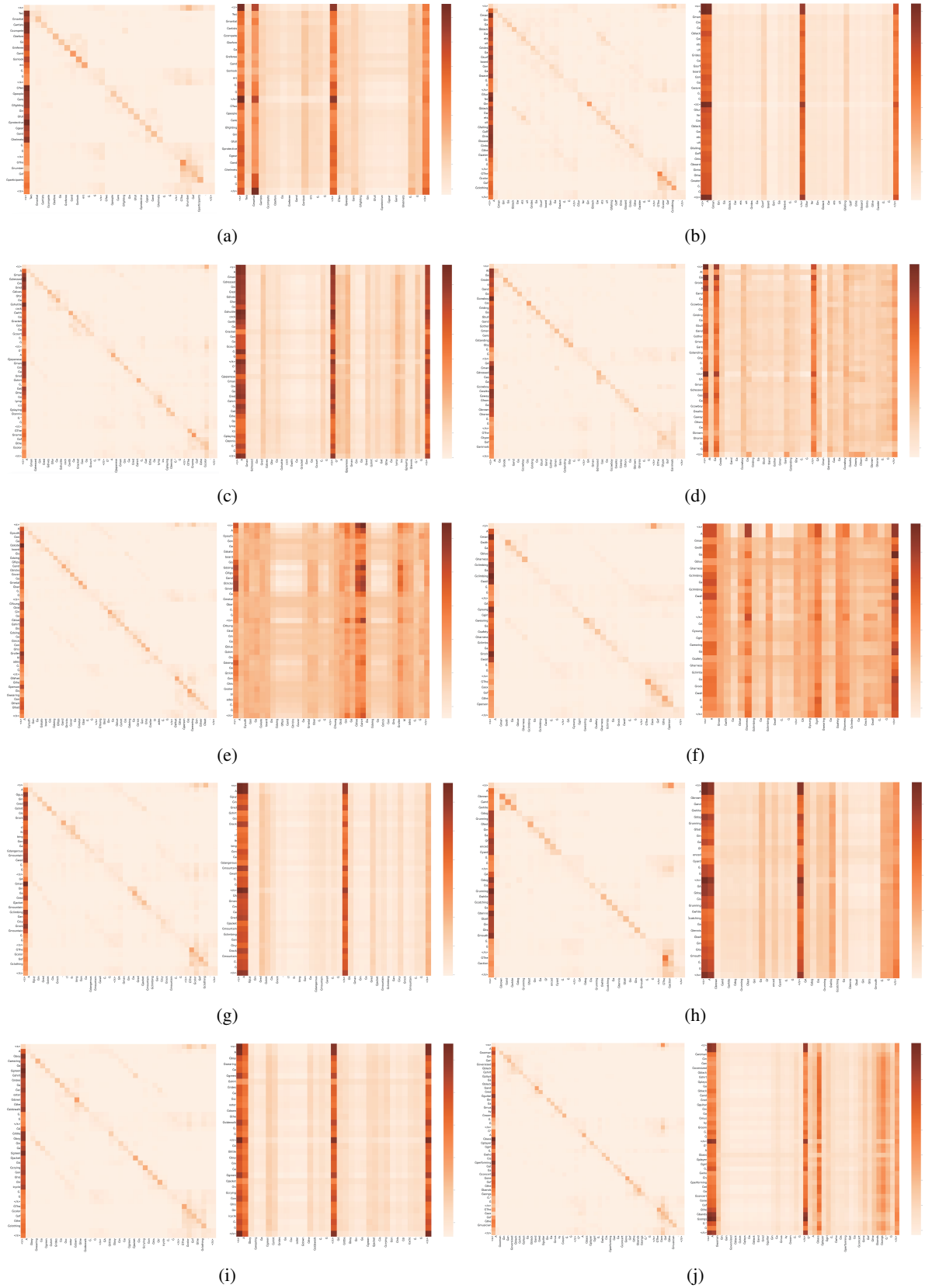


Figure 5: Average attention matrix(left: obtained from the baseline model) and self-reweighting weight(right: obtained from our proposed Self-Augmentation model) of each row case ((a) for the first row, (b) for the second row, etc) presented in Table 6. The darker the color, the larger the corresponding value.