Can Multiple Tokens Improve Sentence Embeddings? A Classification-Based Analysis

Anonymous ACL submission

Abstract

Existing representation models often utilize single-token embedding for downstream tasks, employing approaches such as first token pooling, last token pooling, average pooling, and max pooling for representation. However, these token pooling methods inevitably lead to information loss, as they either ignore or dilute important features from the rest of the sentence. So, would multiple tokens improve sentence embeddings? In this paper, we select the sentence classification task as the research foundation, as it best reflects the quality of sentence embeddings. Randomly selecting multiple tokens is unlikely to effectively improve sentence embeddings; understanding which tokens to use and how to utilize multiple tokens are critical questions that must be explored. Therefore, we propose BTMR, which stands for **B**oosted **T**oken-Level **M**atryoshka **R**epresentation, to investigate the impact of using multiple tokens on sentence embeddings. BTMR operates through two key stages: Fine-to-Coarse Token Matryoshka Learning, which generates token group representation vectors by capturing both local and global contextual information, and Token Fusion Boosting, which aggregates the correct predictions derived from these vectors to produce the final prediction. Experimental results demonstrate that leveraging multiple tokens can indeed improve sentence embeddings.

1 Introduction

In the field of natural language processing (NLP), effective representation of textual data is paramount for the success of various downstream tasks such as question answering (Rajpurkar et al., 2016), classification (Warstadt et al., 2019), sentiment analysis (Socher et al., 2013), and so on. Existing representation models employ various strategies to encapsulate overall text information. Some (Devlin et al., 2019) employ special token representation vectors, some apply average pooling (Li et al., 2023) to token representation vectors, and some (Meng et al., 2024) utilize the representation of the last token. To enhance performance, existing models have expanded parameter scales (Radford et al., 2018), increased dataset diversity (Raffel et al., 2023), and introduced various training or fine-tuning techniques (Wang et al., 2024).

Despite advancements in representation models, the token pooling methods commonly used to obtain sentence embeddings still suffer from inevitable information loss. For instance, First Token Pooling, as used in BERT (Devlin et al., 2019), captures global sequence-level information when trained properly but loses detailed context from the rest of the sentence. Last Token Pooling, employed in the GPT (Radford et al., 2018) series, fails to explicitly consider contributions from preceding tokens. Average Pooling dilutes important features by averaging irrelevant tokens. Max Pooling risks losing information by focusing solely on maximum values and ignoring subtle variations.

These limitations raise an important question: could using multiple tokens improve sentence embeddings? To explore the question, we select the sentence classification task as our research foundation, as it best reflects the quality of sentence embeddings. Randomly selecting multiple tokens is unlikely to effectively improve sentence embeddings; understanding which tokens to use and how to utilize multiple tokens are critical questions that must be explored. Therefore, we introduce BTMR (**B**oosted **T**oken-Level **M**atryoshka **R**epresentation), an approach that effectively takes the advantage of multiple sentence classification to compensate for the shortcomings of the sentence classification prediction of a single token representation vector.

Specifically, BTMR first leverages a Fine-to-Coarse Token Matryoshka Learning strategy to generate token group representation vectors that capture both local and global information. The to-

ken group, when transmitting information, extracts the locally significant information that contributes more to the right prediction, similar to applying a weight to the crucial local information. Then, by employing a Token Fusion Boosting mechanism, BTMR aggregates correct sentence classification from these token group representation vectors, thereby refining the overall prediction and reducing the likelihood of misclassification.

The extensive experiments demonstrate the effectiveness of BTMR, showing that multiple tokens can improve sentence embeddings. Additionally, we conduct a series of ablation studies, which collectively highlight the necessity of Fine-to-Coarse Token Matryoshka Learning and the Token Fusion Boosting mechanism, indicating the importance of methods for selecting and utilizing multiple tokens to improve sentence embeddings.

Our contributions can be summarized as follows: To address the limitations of token pooling methods, which inherently suffer from information loss, we investigate whether multiple tokens could improve sentence embeddings. Since randomly selecting multiple tokens is unlikely to achieve this effectively, we introduce BTMR, which combines Fine-to-Coarse Token Matryoshka Learning and a Token Fusion Boosting mechanism to better select and utilize multiple tokens. Extensive experiments validate the effectiveness of BTMR, demonstrating that multiple tokens can indeed improve sentence embeddings. Ablation studies highlight the necessity of BTMR's components, underscoring the importance of methods for selecting and utilizing multiple tokens in improving sentence embeddings.

2 Related Work

2.1 Representation Models

Representation models have a long history. In this section, we briefly review some of the most widely used and more recent models. Early models, such as Word2Vec (Google, 2013), represented words as vectors using the Skip-gram or CBOW models, making natural language computable. GloVe (Pennington et al., 2014) constructed word vectors based on co-occurrence matrices.

Subsequently, several well-known deep learning representation models emerged. For example, ELMo (Peters et al., 2018) uses bidirectional LSTMs to generate word embeddings, capturing contextual information. Models like ULMFiT (Howard and Ruder, 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and Electra (Clark et al., 2020) follow the paradigm of pre-training on large datasets and then fine-tuning on downstream tasks. GPT (Radford et al., 2018), LLaMA (Touvron et al., 2023), Mistral (Jiang et al., 2023) and other decoder-only models also follow this paradigm; however, they use a unidirectional Transformer (Vaswani et al., 2023) and pre-train in a generative way, whereas the BERT series uses a bidirectional Transformer to learn deep contextual representations. BART (Lewis et al., 2020) combines BERT's bidirectional encoding with GPT's autoregressive decoding, providing effective sequence-to-sequence modeling. T5 (Raffel et al., 2023) introduces transfer learning into NLP by converting text-based language problems into a text-to-text format. ST-MoE (Zoph et al., 2022) uses a Mixture-of-Experts (MoE) approach for stable training and improved capability. Vega v2 (Zhong et al., 2022) enhances performance through self-evolution learning.

Recently, some models represent the overall text through methods like average pooling over the representation vectors of tokens (as in GTE (Li et al., 2023)), using the last token's representation vector (as in E5-Mistral-7B-instruct (Wang et al., 2024) and SFR-Embedding-Mistral (Meng et al., 2024)), or employing a latent attention layer (as in NV-Embed (Lee et al., 2024)). Gist (Mu et al., 2024), Transformer-XL (Dai et al., 2019) and AnLLM (Pang et al., 2024) distill context information into several designed tokens.

In addition to these monolingual models, some models extend representation to multilingual contexts, such as mBERT (Devlin et al., 2019), XLM (Lample and Conneau, 2019), XLM-R (Conneau et al., 2020), mT5 (Xue et al., 2020), MASS (Song et al., 2019), and XY-LENT (Patra et al., 2022).

Besides these flat representation models, there are also models that perform hierarchical representation learning. For example, MRL (Kusupati et al., 2022) designs a flexible representation that can adapt to multiple downstream tasks with varying computational resources; AdANNS (Rege et al., 2023) leverages different stages of approximate nearest neighbor search for adaptive semantic search; and repLLaMA-rankLLaMA (Ma et al., 2024) utilizes a multi-stage text ranking pipeline to enhance a variety of retrieval tasks.

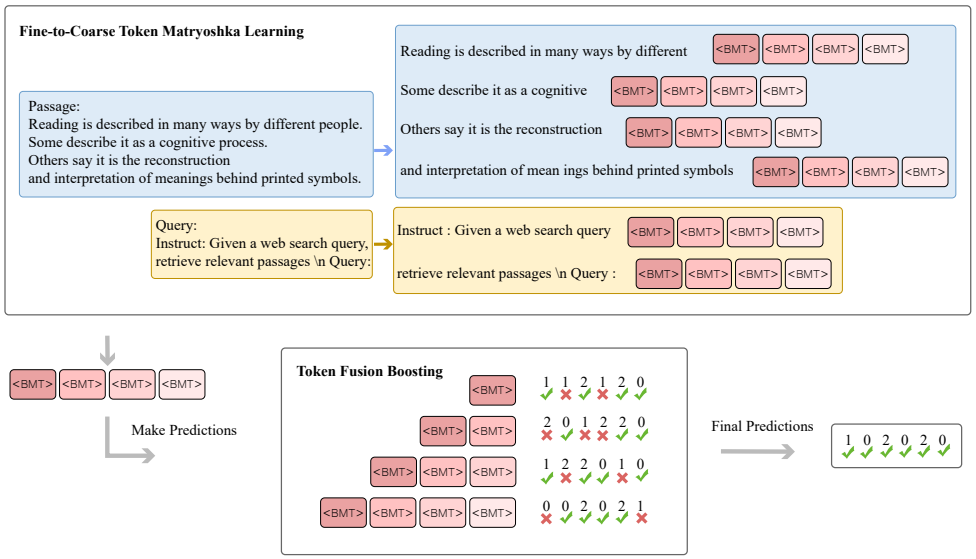

Figure 1: The overall framework of BTMR. BTMR first leverages a Fine-to-Coarse Token Matryoshka Learning strategy to generate token group representation vectors that capture both local and global information. Then, by employing a Token Fusion Boosting mechanism, BTMR aggregates correct sentence classification from these token group representation vectors, refining the overall prediction. Symbols 0, 1, 2 represent classification categories.

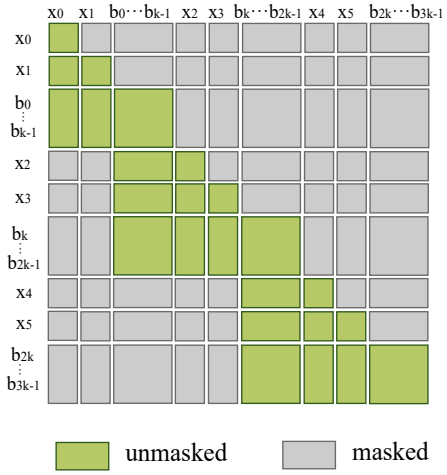

Figure 2: The attention masks utilized in Fine-to-Coarse Token Matryoshka Learning.

2.2 Ensemble Learning

We primarily discuss boosting in this section. Boosting is a powerful ensemble learning technique that has significantly influenced the field of machine learning. The concept was first introduced by Schapire with the proposal of the Weak Learnability framework (Schapire, 1990), leading to the development of the AdaBoost (Schapire et al., 1999) algorithm. AdaBoost, short for Adaptive Boosting, combines weak learners sequentially, adjusting the weights of misclassified examples to focus the learning on harder cases. Following AdaBoost, many variants have been proposed. Gradient Boosting Machines (GBM) (Friedman, 2001, 2002; Mason et al., 1999) extend the idea of boosting to optimize any differentiable loss function. This has been further popularized by implementations like XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2019), which focus on improving computational efficiency, scalability, and handling of categorical features. Moreover, recent advances have incorporated boosting techniques into deep learning. For instance, (Schwenk and Bengio, 1997, 2000; Han et al., 2016) combine the strengths of boosting and neural networks to tackle complex tasks.

3 Multi-Token Sentence Representation Learning

3.1 Preliminary

We limit the representation learning discussed in this paper to the field of NLP. Given a sequence of tokens $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ representing a text of length n, the objective of text representation learning is to learn a mapping function $f : \mathbf{X} \rightarrow \mathbf{H}$, where $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$ denotes the set of hidden representation vectors corresponding to each token x_i in the input sequence. Each hidden representation vector $\mathbf{h}_i \in \mathbb{R}^d$ is a d-dimensional vector that captures semantic information of the token x_i within context of the entire sequence $\mathbf{X}$.

The goal is to ensure that the learned representations $\mathbf{H}$ effectively capture both the local and

global semantic information of the text. To this end, we can define the problem as minimizing a loss function $\mathcal{L}(\mathbf{H})$, which typically involves tasks such as predicting the next token, reconstructing the input sequence, or fine-tuning on specific downstream tasks. Formally, the problem can be expressed as:

$$\mathbf{H} = f(\mathbf{X}; \theta) \tag{1}$$

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\mathbf{H}, \mathbf{Y}), \tag{2}$$

where θ represents the parameters of the mapping function f, and $\mathbf{Y}$ denotes the ground truth labels or targets used for supervision, depending on the specific task at hand. The learned representation $\mathbf{H}$ should not only be robust and informative for the input sequence $\mathbf{X}$ but also be generalizable across various downstream tasks.

Token pooling methods used in single-token representation models to obtain the sentence embedding z can be represented as follows. **First Token Pooling** selects the hidden representation of the first token in the sequence: $z = \mathbf{h}_1$. **Last Token Pooling** selects the hidden representation of the last token in the sequence: $z = \mathbf{h}_n$. **Average Pooling** computes the average of all token representations across the sequence: $z = \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}_i$. **Max Pooling** selects the maximum value across the token representations for each dimension: $z_j = \max_{i=1}^{n} \mathbf{h}_{i,j}, \quad j = 1, 2, \ldots, d$, where z_j is the j-th dimension of z.

To address the limitations of traditional single-token representation models, we introduce **B**oosted **T**oken-Level **M**atryoshka **R**epresentation (BTMR) to investigate whether multiple tokens can improve sentence embeddings. BTMR operates by initially applying Fine-to-Coarse Token Matryoshka Learning, which generates token group representation vectors that capture both local and global information. These vectors are then utilized through a boosting mechanism that aggregates the accurate sentence classification to produce a final prediction. We design a special token <BMT> and the overall framework is in Figure 1.

3.2 Fine-to-Coarse Token Matryoshka Learning

Given text token sequence $\mathbf{X}$, to capture not only detailed local information but also global information, BTMR divides $\mathbf{X}$ into multiple fixed-length segments, inserting a fixed number of learnable special tokens <BMT> after each segment. By using a specific attention mask, BTMR condenses the information of each segment s and the information contained in the <BMT> token group before s into the <BMT> token group after s. By continuously transferring information from the previous <BMT> token group to the subsequent <BMT> token group, the information contained in the final <BMT> token group of $\mathbf{X}$ gradually shifts from fine-grained local details to coarse-grained global information.

In this process, BTMR utilizes Matryoshka representation learning (Kusupati et al., 2022) and adds the losses generated by the first $1, 2, \ldots, k$ tokens in the <BMT> token group during training to the overall loss. This concentrates the information density in the leading tokens of the <BMT> token group, so that even with fewer <BMT> tokens, critical information can still be obtained, while adding more <BMT> tokens provides richer information.

Algorithm 1 Token Fusion Boosting Algorithm

Input: Training data $\{(x_i^{:1}, y_i), (x_i^{:2}, y_i), \ldots, (x_i^{:k}, y_i)\}_{i=1}^{n}$, where $x_i^{:k}$ is the sentence classification prediction of the first k <BMT> tokens' representation vectors in the <BMT> token group of the i-th data item, and y_i is the label; weak learner $b_f(x; \gamma)$, where γ is its parameters; loss function $L(y, f(x))$; number of boosting rounds T

Parameter: Weak learner coefficient β; Weak learner parameters γ

Output: Final prediction model $f(x)$

1: Initialize the model with a constant: $f_0(x) = \arg\min_c \sum_{i=1}^{n} L(y_i, c)$

2: **for** $t = 1$ to T **do**

3: let $x \leftarrow Concat([x^{:1}, x^{:2}, \ldots, x^{:k}])$

4: Minimize the loss function:

$$\arg\min_{\beta,\gamma} \sum_{i=1}^{n} L\left(y_i, f_{t-1}(x_i) + \beta b_f(x_i; \gamma)\right)$$

to obtain the parameters β_t, γ_t.

5: Update the model:

$$f_t(x) = f_{t-1}(x) + \beta_t b_f(x; \gamma_t)$$

6: **end for**

7: **return** the final prediction model:

$$f(x) = f_T(x) = f_0(x) + \sum_{t=1}^{T} \beta_t b_f(x; \gamma_t)$$

Model Name	E5-Mistral-7B-instruct	<BMT>-Num-1	<BMT>-Num-2	<BMT>-Num-3	<BMT>-Num-4	BTMR
MTOPDomain	96.12	96.09	96.09	96.02	95.99	96.01
Banking77	88.23	87.88	87.95	87.94	87.90	**88.24**
AmazonCounterfactual	78.69	73.36	73.21	73.31	74.16	**78.72**
Emotion	49.77	45.95	46.37	47.13	47.72	**52.20**
ToxicConversations	69.59	66.94	67.91	68.16	68.78	**73.66**
MassiveScenario	82.39	82.06	81.94	81.70	81.22	**82.40**
TweetSentimentExtraction	63.72	62.51	62.09	61.84	61.78	**64.74**
AmazonPolarity	95.91	95.19	95.05	95.70	95.90	**95.92**
AGNews	86.27*	86.98	**87.67**	87.58	87.36	87.53
RottenTomatoes	91.51*	91.38	90.97	90.75	90.69	91.24
DBpedia14	95.40*	**95.52**	94.94	94.62	94.25	94.77
ClimateSentiment	72.22*	**72.75**	71.44	68.66	68.78	70.94
Financial	53.83*	55.32	57.81	59.25	**59.62**	59.25
EnvironmentalClaim	82.34*	82.23	81.92	81.43	81.13	**82.53**
Average	79.00	78.15	78.24	78.15	78.23	**79.87**

Table 1: Results compared to the baseline E5-Mistral-7B-instruct. The results marked with * are the performance we obtain using E5-Mistral-7B-instruct model.

The attention masks used is shown in the Figure 2. The idea is that the segment s and the <BMT> token group after it can only see the previous <BMT> token group, meaning the conditional probability of predicting the next token x_i is $p(x_i|b_1, \ldots, b_k, x_{i-t}, \ldots, x_{i-1})$, where b represents the <BMT> token, k is the number of <BMT> tokens in the <BMT> token group, and t is the number of text tokens in the segment s before x_i.

Pre-training We use the MS-MARCO BM25 dataset processed by SimLM (Wang et al., 2023) to pre-train the <BMT> token group representation vectors for 1 epoch. For each query-passage pair (q, d) in the dataset, we segment both q and d and then insert <BMT> token groups. We use standard language modeling cross-entropy loss:

$$L_{lm} = -\sum_{i=1}^{n} \log P(y_i \mid y_{1:i-1}; \theta) \quad (3)$$

In addition to the standard language modeling loss, we also perform contrastive learning by comparing the representation vectors of the first $1, 2, \ldots, k$ <BMT> tokens following each segment of q with those following each segment of d. For positive samples, we aim to increase the matching score, while for negative samples, we aim to decrease it. The contrastive loss used is as follows:

$$l_{neg} = \sum_{n \in \mathbb{N}} (\phi(b_{s_q}^{1:i}, b_{s_n}^{1:i}) + \phi(b_{s_d}^{1:i}, b_{s_n}^{1:i})) \quad (4)$$

$$l_i = -\log \frac{\phi(b_{s_q}^{1:i}, b_{s_d}^{1:i})}{\phi(b_{s_q}^{1:i}, b_{s_d}^{1:i}) + l_{neg}} \quad (5)$$

$$L_{cons} = \sum_{s_q \in \mathbb{Q}} \sum_{s_d \in \mathbb{D}} \sum_{i=1}^{k} l_i \quad (6)$$

where $b_{s_q}^{1:i}, b_{s_d}^{1:i}, b_{s_n}^{1:i}$ represent the vector formed by concatenating the representation vectors of the first i <BMT> tokens in the <BMT> token group following the query segment s_q, passage segment s_d, and negative passage segment s_n; k represents the number of <BMT> tokens in the <BMT> token group; $\mathbb{N}$ denotes all the negatives; $\mathbb{Q}$ and $\mathbb{D}$ represent segments set of query q and passage d; and $\phi(q, d)$ is a function to compute the matching score between the representation vector of q and d. We use temperature-scaled cosine similarity function: $\phi(q, d) = \exp(\frac{1}{\tau} \cos(\mathbf{h}_q, \mathbf{h}_d))$, where $\mathbf{h}_q$ and $\mathbf{h}_d$ denote representation vector of q and d; τ is a temperature hyper-parameter and set to a constant 0.02 in pre-training.

Therefore, the total loss for pre-training is:

$$\mathcal{L} = L_{lm} + L_{cons} \quad (7)$$

3.3 Token Fusion Boosting

After obtaining the pre-trained <BMT> token group representation vectors, Token Fusion Boosting mechanism is applied to utilize the sentence classification of <BMT> token group representation vectors, resulting in improved performance. The algorithm of Token Fusion Boosting is as in Algorithm 1.

By combining the sentence classification of multiple weak learners, each of which focuses on different aspects of the sentence classification of the <BMT> tokens' representation vectors, Token Fusion Boosting reduces the error iteratively.

4 Experiments

4.1 Datasets

We select a total of 14 datasets, including MTOP-Domain (Li et al., 2020), Banking77 (Casanueva et al., 2020), AmazonCounterfactual (O'Neill et al., 2021), Emotion (Saravia et al., 2018), ToxicConversations (cjadams et al., 2019), MassiveScenario (FitzGerald et al., 2022), TweetSentimentExtraction (Maggie, 2020), AmazonPolarity (Zhang et al., 2015), AGNews (Zhang et al., 2015), RottenTomatoes (Pang and Lee, 2005), DBpedia14 (Zhang et al., 2015), ClimateSentiment (Bingler et al., 2024), Financial (nickmuchi, 2022), and EnvironmentalClaim (Stammbach et al., 2023). We use accuracy as the reported metric.

4.2 Baseline

We choose E5-Mistral-7B-instruct (Wang et al., 2024) as the baseline model for comparison. After initializing our model with the parameters from E5-Mistral-7B-instruct, we used the average embedding weight and bias of tokens from its vocabulary to initialize the weight and bias of the <BMT> tokens. Following the pre-training phase (where we set the segment length to 64 tokens and select 9 negative samples for each data pair), we evaluate the BTMR model using the checkpoint that achieved the best Mean Reciprocal Rank (MRR) metric. The MRR is a measure used to evaluate the quality of ranking predictions.

4.3 Evaluation

During evaluation, given a data pair (d, y), where d is the sentence to be classified and y is the label, we follow the same approach as E5-Mistral-7B-instruct by prepending the task-specific instruction to d, resulting in the input d^+. We then append a <BMT> token group—specifically k <BMT> tokens—only to the end of each input d^+. In our experiments, unless otherwise noted, k is set to 4.

For each label type, we use only 8 samples as training data. We transform the input d^+ into representation vectors, using the representation vectors of the <BMT> token group as the feature x for the training data, with the corresponding ground-truth sentence classification type as the label. Consistent with E5-Mistral-7B-instruct, we fit the Logistic Regression classifier for the classification. We fit k Logistic Regression classifiers, with the predictions from these classifiers representing the sentence classification for the first $1, 2, \ldots, k$ <BMT> tokens' representation vectors, employ these Logistic Regression classifiers as weak learners, and apply the Token Fusion Boosting Algorithm to achieve the final performance of the BTMR model.

Model Name	BTMR	BTMR$_{coarse}$
MTOPDomain	**96.01**	93.81
Banking77	**88.24**	84.69
AmazonCounterfactual	**78.72**	75.49
Emotion	**52.20**	51.19
ToxicConversations	**73.66**	70.37
TweetSentimentExtraction	**64.74**	64.61
AGNews	**87.53**	85.33
Financial	**59.25**	58.95
Average	**75.04**	73.06

Table 2: Ablation results of Coarse Token Matryoshka Learning. Coarse Token Matryoshka Learning only extracts global information without emphasizing local information.

4.4 Main Results

The main results are in the Table 1. As observed, the BTMR method outperforms E5-Mistral-7B-instruct on almost all datasets, with only slight differences of 0.03 and 0.13 on the MTOPDomain and RottenTomatoes datasets. This demonstrates the effectiveness of BTMR, implying that multiple tokens can improve sentence embeddings. Although BTMR's performance when considering individual sentence classification of the first $1, \ldots, k$ <BMT> tokens' representation vectors may not always surpass that of E5-Mistral-7B-instruct, BTMR generally exceeds it after applying Token Fusion Boosting, indicating the importance of the method used to utilize multiple tokens in improving sentence embeddings, and that effectively utilizing multiple tokens can better enhance sentence embeddings.

In some datasets, such as AGNews and Financial, the performance after Token Fusion Boosting shows a slight decline compared to the performance before Token Fusion Boosting. This is because that while Token Fusion Boosting typically leverages the correct sentence classification from the <BMT> token group's representation vectors, it can occasionally be influenced by the erroneous sentence classification of a majority within the <BMT> token group for a particular data instance. However, this interference is quite limited.

4.5 Ablation Study

Regarding Fine-to-Coarse Token Matryoshka Learning, we conduct two ablation studies to indi-

Ablation	Model	Banking77	AmazonCounterfactual	MTOPDomain	ToxicConversations	Average
BTMR	<BMT>-Num-1	87.88	73.36	96.09	66.94	81.07
	<BMT>-Num-2	87.95	73.21	96.09	67.91	81.29
	<BMT>-Num-3	87.94	73.31	96.02	68.16	81.36
	<BMT>-Num-4	87.90	74.16	95.99	68.78	81.71
	boosted	**88.24**	**78.72**	96.01	**73.66**	84.16
BTMR$_{w/o\ Matryoshka}$	<BMT>-Num-1	85.40	73.84	92.90	68.98	80.28
	<BMT>-Num-2	84.87	73.60	92.90	68.28	79.91
	<BMT>-Num-3	84.07	73.57	92.79	68.59	79.75
	<BMT>-Num-4	83.81	73.78	92.74	68.70	79.76
	boosted	82.81	76.15	92.34	67.54	79.71

Table 3: Ablation results of Fine-to-Coarse Token Learning. The pre-training loss of Fine-to-Coarse Token Learning includes only the standard language modeling loss and the contrastive learning loss for the <BMT> token group.

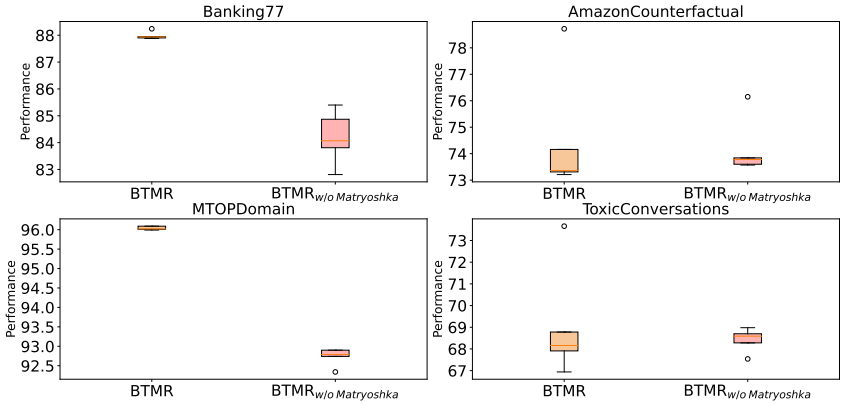

Figure 3: The boxplot of Fine-to-Coarse Token Learning results. It shows the median, quartile range, and distribution of the results. The ○ markers indicate values that are far from the main data distribution.

cate the importance of effectively selecting multiple tokens for improving sentence embeddings as follows.

Coarse Token Matryoshka Learning: In this ablation, we do not segment the text and insert the <BMT> token groups. Instead, we only append the <BMT> token group at the end, extracting only global information without emphasizing local information. The resulting performance is referred to as BTMR$_{coarse}$, and the results are shown in the Table 2. The results indicate that by extracting more detailed local information, the accuracy of the predictions improves, highlighting that multiple tokens should combine both detailed local and global information, rather than focusing solely on global information.

Fine-to-Coarse Token Learning: In this ablation, we omit the contrastive learning loss for the first $1, 2, \ldots, k-1$ <BMT> tokens. The pre-training loss includes only the standard language modeling loss and the contrastive learning loss for the <BMT> token group, resulting in a loss function of:

$$\mathcal{L}_{w/o\ Matryoshka} = L_{lm} + L_{cons\ w/o\ Matryoshka} \tag{8}$$

$$= L_{lm} + \sum_{s_q \in \mathbb{Q}} \sum_{s_d \in \mathbb{D}} l_k \tag{9}$$

From the results in Table 3 and Figure 3, it is evident that the performance deteriorates without Matryoshka Learning. Additionally, on datasets where Token Fusion Boosting performs well, such as Banking77 and ToxicConversations, the boosted performance of BTMR$_{w/o\ Matryoshka}$ does not improve, indicating that the absence of Matryoshka Learning leads to more dispersed information. In such cases, the sentence classification from multiple <BMT> tokens' representation vectors become quite similar, and the effectiveness of Token Fusion Boosting is not fully realized. Therefore, multiple tokens should contain more focused information, meaning that the representation vectors of the tokens should be more diverse.

Regarding Token Fusion Boosting, we also conduct two ablation studies to present the importance

Ablation	Model	MTOPDomain	Banking77	AmazonCounterfactual	ClimateSentiment	DBpedia14	RottenTomatoes
BTMR	<BMT>-Num-1	96.09	87.88	73.36	72.75	95.52	91.38
	<BMT>-Num-2	96.09	87.95	73.21	71.44	94.94	90.97
	<BMT>-Num-3	96.02	87.94	73.31	68.66	94.62	90.75
	<BMT>-Num-4	95.99	87.90	74.16	68.78	94.25	90.69
	boosted	96.01	**88.24**	**78.72**	70.94	94.77	91.24
BTMR$_{non-nested}$	<BMT>-1th	96.09	87.88	73.36	72.75	95.52	91.38
	<BMT>-2nd	95.82	87.86	71.28	65.34	93.49	88.75
	<BMT>-3rd	95.51	87.89	72.85	65.34	92.91	87.54
	<BMT>-4th	95.42	87.39	72.31	63.00	91.88	85.39
	boosted	96.01	88.22	76.97	70.94	94.77	91.22

Table 4: Ablation results of Token Fusion Boosting without Nested Representation Vectors. Token Fusion Boosting without Nested Representation Vectors only applies Token Fusion Boosting to the 1-th,. . . ,k-th <BMT> token.

of effectively utilizing multiple tokens for improving sentence embeddings as follows.

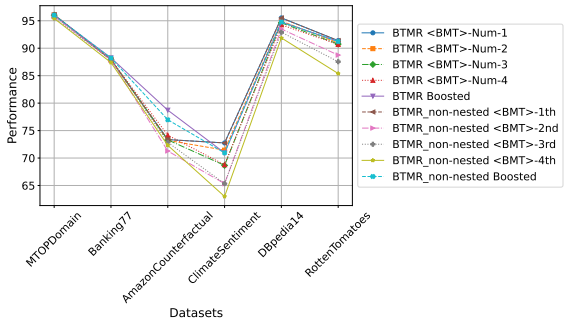

Figure 4: The line chart of Token Fusion Boosting without Nested Representation Vectors Results.

Model Name	<BMT>-4	<BMT>-3	<BMT>-2
MTOPDomain	**96.01**	95.98	95.79
Banking77	**88.24**	88.22	88.06
AmazonCounterfactual	78.72	76.73	**79.66**
Emotion	**52.20**	50.08	50.53
MassiveScenario	**82.40**	81.72	81.88
TweetSentimentExtraction	**64.74**	62.99	62.82
AGNews	87.53	**88.50**	88.49
RottenTomatoes	91.24	91.11	**91.27**
Financial	**59.25**	58.00	56.82
EnvironmentalClaim	**82.53**	81.81	81.92
Average	**78.29**	77.51	77.72

Table 5: The ablation results of Token Fusion Boosting with Different Token Numbers.

Token Fusion Boosting without Nested Representation Vectors: We explore the impact of not using nested representation by applying Token Fusion Boosting to the 1-th, . . . , k-th <BMT> token, with the results shown in the Table 4 and Figure 4. From the results, when nested representation is not used, the performance remains fairly consistent, further demonstrating that Token Fusion Boosting effectively leverages the correct sentence classification from different <BMT> tokens' representation vectors, even when the performance of the 2-nd, . . . , k-th <BMT> tokens is noticeably worse than that of the first $2, \ldots, k$ <BMT> tokens. In other words, as long as multiple tokens are effectively utilized, whether or not nested representation vectors are used, the improvement in sentence embeddings is similar.

Token Fusion Boosting with Different Token Numbers: We investigate the effects of using different numbers of <BMT> tokens for Token Fusion Boosting, with the results presented in the Table 5. The reason the number of <BMT> tokens is limited to 4 is that increasing the number of tokens further is unlikely to yield substantial improvement, and it also increases memory usage. To balance performance and memory consumption, we select 4 tokens as the upper bound. Based on the results, in general, using more <BMT> tokens with Token Fusion Boosting tends to lead to better performance, which is consistent with our conclusion that multiple tokens can improve sentence embeddings. However, in some cases, more <BMT> tokens might introduce some noise, and the performance may not necessarily be better than boosting with fewer <BMT> tokens. In other words, the choice of the number of multiple tokens should balance both efficiency and benefits.

5 Conclusion

In this paper, we explore whether using multiple tokens can improve sentence embeddings to address the limitations of traditional single-token representation models. We propose BTMR, which leverages Fine-to-Coarse Token Matryoshka Learning and Token Fusion Boosting. Experiments show that by appropriately selecting and utilizing multiple tokens, sentence embeddings can be improved.

Limitations

Our paper focuses on the Sentence Classification task. Sentence classification differs from tasks such as clustering, reranking, retrieval and so on. In Sentence Classification, each data entry can be directly classified into its category independently, whereas other tasks require batch-level operations like clustering or similarity ranking. Our approach aggregates the correct predictions to enhance performance and is particularly well-suited for the sentence classification task. So we select the classification datasets from MTEB. Other MTEB datasets are designed for clustering, reranking, retrieval, etc., which are beyond the scope of our current research.

References

Julia Anna Bingler, Mathias Kraus, Markus Leippold, and Nicolas Webersinke. 2024. How cheap talk in climate disclosures relates to climate initiatives, corporate emissions, and reputation risk. *Journal of Banking & Finance*, 164:107191.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *Preprint*, arXiv:2003.04807.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

cjadams, Daniel Borkan, inversion, Jeffrey Sorensen, Lucas Dixon, Lucy Vasserman, and nithum. 2019. Jigsaw unintended bias in toxicity classification.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

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

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *Preprint*, arXiv:2204.08582.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378.

Google. 2013. word2vec. *Google Code Archive*.

Shizhong Han, Zibo Meng, Ahmed-Shehab Khan, and Yan Tong. 2016. Incremental boosting convolutional neural network for facial action unit recognition. *Advances in neural information processing systems*, 29.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Preprint*, arXiv:1901.07291.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *Preprint*, arXiv:1909.11942.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880, Online. Association for Computational Linguistics.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *Preprint*, arXiv:2308.03281.

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. *Preprint*, arXiv:1907.11692.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

Wei Chen Maggie, Phil Culliton. 2020. Tweet sentiment extraction.

Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. 1999. Boosting algorithms as gradient descent. *Advances in neural information processing systems*, 12.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral: Enhance text retrieval with transfer learning. *Salesforce AI Research*.

Jesse Mu, Xiang Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36.

nickmuchi. 2022. financial-classification.

James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. I wish i would have loved this one, but i didn't–a multilingual dataset for counterfactual detection in product reviews. *arXiv preprint arXiv:2104.06893*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Jianhui Pang, Fanghua Ye, Derek F Wong, and Longyue Wang. 2024. Anchor-based large language models. *arXiv preprint arXiv:2402.07616*.

Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav Chaudhary, and Xia Song. 2022. Beyond english-centric bitexts for better multilingual language representation learning. *Preprint*, arXiv:2210.14867.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *Preprint*, arXiv:1802.05365.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2019. Catboost: unbiased boosting with categorical features. *Preprint*, arXiv:1706.09516.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *Technical report, OpenAI*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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

Aniket Rege, Aditya Kusupati, Alan Fan, Qingqing Cao, Sham Kakade, Prateek Jain, Ali Farhadi, et al. 2023. Adanns: A framework for adaptive semantic search. *Advances in Neural Information Processing Systems*, 36:76311–76335.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*,

pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Robert E Schapire. 1990. The strength of weak learnability. *Machine learning*, 5:197–227.

Robert E Schapire et al. 1999. adaboost. In *Ijcai*, volume 99, pages 1401–1406. Citeseer.

Holger Schwenk and Yoshua Bengio. 1997. Training methods for adaptive boosting of neural networks. In *Advances in Neural Information Processing Systems*, volume 10. MIT Press.

Holger Schwenk and Yoshua Bengio. 2000. Boosting neural networks. *Neural computation*, 12(8):1869–1887.

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

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Dominik Stammbach, Nicolas Webersinke, Julia Anna Bingler, Mathias Kraus, and Markus Leippold. 2023. Environmental claim detection. *Preprint*, arXiv:2209.00507.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. *Preprint*, arXiv:2401.00368.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Qihuang Zhong, Liang Ding, Yibing Zhan, Yu Qiao, Yonggang Wen, Li Shen, Juhua Liu, Baosheng Yu, Bo Du, Yixin Chen, Xinbo Gao, Chunyan Miao, Xiaoou Tang, and Dacheng Tao. 2022. Toward efficient language model pretraining and downstream adaptation via self-evolution: A case study on superglue. *Preprint*, arXiv:2212.01853.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *Preprint*, arXiv:2202.08906.

Model Name	SFR-Embedding-Mistral	<BMT>-Num-1	<BMT>-Num-2	<BMT>-Num-3	<BMT>-Num-4	BTMR
AmazonCounterfactual	77.93	75.87	75.85	76.09	76.09	77.40
Emotion	50.24	51.76	51.66	51.53	51.36	**55.48**
TweetSentimentExtraction	63.64	64.13	64.10	63.90	64.31	**65.35**
RottenTomatoes	91.59*	91.03	91.14	91.23	91.40	91.50
ClimateSentiment	72.78*	73.69	73.41	73.69	73.59	**73.91**
Financial	53.87*	**58.44**	57.75	57.29	57.96	57.85
EnvironmentalClaim	82.64*	82.87	82.68	82.87	**83.70**	83.13
Average	70.38	71.11	70.94	70.94	71.20	**72.09**

Table 6: Results compared to the baseline SFR-Embedding-Mistral. The results marked with * are outcomes for datasets where SFR-Embedding-Mistral did not report results, which we measured directly under the same environment.

A Further Experiments

We test our method based on the new SFR-Embedding-Mistral model, with results shown in Table 6. Overall improvement can be observed on the tested datasets, with significant gains on some, such as a 2.69 % improvement on the TweetSentimentExtraction dataset, an 8.48% increase on the Financial dataset, and a 10.43% increase on the Emotion dataset.