Rare Tokens Degenerate All Tokens: Improving Neural Text Generation via Adaptive Gradient Gating for Rare Token Embeddings

Anonymous ACL submission

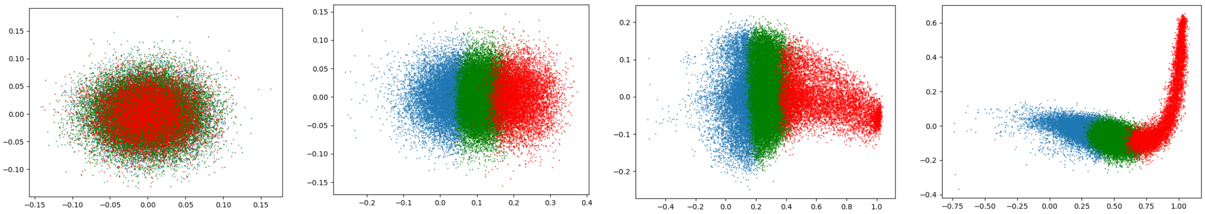
Abstract

Recent studies have determined that the learned token embeddings of large-scale neural language models are degenerated to be anisotropic with a narrow-cone shape. This phenomenon, called the representation degeneration problem, facilitates an increase in the overall similarity between token embeddings that negatively affect the performance of the models. Although the existing methods that address the degeneration problem based on observations of the phenomenon triggered by the problem improves the performance of the text generation, the training dynamics of token embeddings behind the degeneration problem are still not explored. In this study, we analyze the training dynamics of the token embeddings focusing on rare token embedding. We demonstrate that the specific part of the gradient for rare token embeddings is the key cause of the degeneration problem for all tokens during training stage. Based on the analysis, we propose a novel method called, *adaptive gradient gating* (AGG). AGG addresses the degeneration problem by gating the specific part of the gradient for rare token embeddings. Experimental results from language modeling, word similarity, and machine translation tasks quantitatively and qualitatively verify the effectiveness of AGG.

1 Introduction

Neural language models have been developed with various architectures during recent years (Graves, 2013; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). Despite the improvement in model architectures, models usually share the same process for input and output. They process token embeddings as inputs to compute contextualized features and subsequently project the features into a categorical distribution of tokens at the output softmax layer whose weight is token embedding matrix (Merity et al., 2017; Yang et al., 2018; Press and Wolf, 2017). Recent studies have determined that the learned embedding distribution is biased in a common direction, thereby resulting in a narrow cone-shaped anisotropy (Mu et al., 2018; Ethayarajh, 2019; Gao et al., 2019; Biś et al., 2021). This phenomenon, named the representation degeneration problem by Gao et al. (2019), increases the overall similarity between embeddings, and leads to a problem in which the expressiveness of the token embeddings decreases. Therefore, it is difficult for the model to learn the semantic relationship between the tokens and to generate high quality texts. Existing studies addressing this problem suggest methods that apply post-processing or regularization techniques to all token embeddings based on the observed phenomena owing to the degeneration problem (Mu et al., 2018; Gao et al., 2019; Wang et al., 2019; Wang et al., 2020; Biś et al., 2021). Although these works improve the quality of token embeddings and generated texts, it is still not clear how token embeddings become degenerate during training procedure. Also, there exists the problem of over regularization for the token embeddings whose semantic relationships are trained well because the above methods are applied for all token embeddings.

In this study, we conduct empirical studies about training dynamics of token embeddings, focusing on rare token embeddings. By observing the initial training dynamics of token embeddings grouped based on appearance frequency, we hypothesize that the degeneration of the rare token embeddings triggers the degeneration of the embeddings of the remaining tokens. We show that the entire degeneration problem is mitigated by only freezing rare tokens during training, and we demonstrate that the main cause of the entire degeneration problem is the specific part of the gradient for rare token embeddings. This gradient part roles to push away rare token embeddings from the feature vector of the non-rare targets in the current training sample. Based on the analysis, we propose a new method, *adaptive gradient gating* (AGG). With a dynamic



(a) Training step 100 (b) Training step 500 (c) Training step 1500 (d) Training step 3500

Figure 1: Visualization of token embeddings of language model trained on WikiText-103. Red, green, and blue points represent rare, medium, and frequent groups respecively. (a), (b), (c), (d) present a visualization of each training step.

grouping of rare tokens at each training step, AGG solves the entire degeneration problem by gating a specific part of the gradient that is solely about rare tokens. Because AGG is optimized to target the main cause of the degeneration problem, rare token embeddings, it can prevent the over regularization problem about frequent token embeddings which occurs in other methods addressing the degeneration problem. The proposed method is evaluated in three tasks: language modeling, word similarity, and machine translation. The AGG outperforms the baseline and other existing methods in all tasks. In addition, it shows compatibility with other method that addresses the neural text degeneration problem. Via qualitative studies, we identify a correlation between our method and the frequency bias problem of learned embeddings (Gong et al., 2018; Ott et al., 2018).

2 Background

2.1 Text Generation of Neural Language Models

Neural language generative models process text generation tasks as conditional language modeling, in which the model is typically trained by minimizing the negative log likelihood of the training data. With a vocabulary of tokens $V = \{v_1, ..., v_N\}$ and embedding vectors $\{\mathbf{w}_1, ..., \mathbf{w}_N\}$, where $\mathbf{w}_i$ corresponds to token v_i, at every training step, the model obtains a mini-batch input and target text corpus pair ($\mathbf{x}$, $\mathbf{y}$), where $x_i, y_i \in V$, and $\mathbf{y} \in V^T$. The conditional probability for the target token y_t, $P_\theta(y_t|\mathbf{h}_t)$, where $\mathbf{h}_t$ is a context feature vector of the t-th position of the generated text conditioned by ($\mathbf{x}$, $y_{<t}$), and θ denotes model parameters, which is defined as follows.

$$P_\theta(y_t|\mathbf{h}_t) = \frac{\exp\left(\mathbf{h}_t \mathbf{w}_{I(y_t)}^T\right)}{\sum_{l=1}^{N} \exp \mathbf{h}_t \mathbf{w}_l^T}, \qquad (1)$$

where $\mathbf{w}$ is the output token embedding which roles the weight of the output softmax layer, and $I(y_t)$ represents the index of token y_t. The negative log likelihood loss for an input and target pair ($\mathbf{x}$, $\mathbf{y}$), L_{NLL} is expressed as follows.

$$L_{NLL} = -\sum_{t=1}^{T} \log P_\theta(y_t|\mathbf{h}_t). \qquad (2)$$

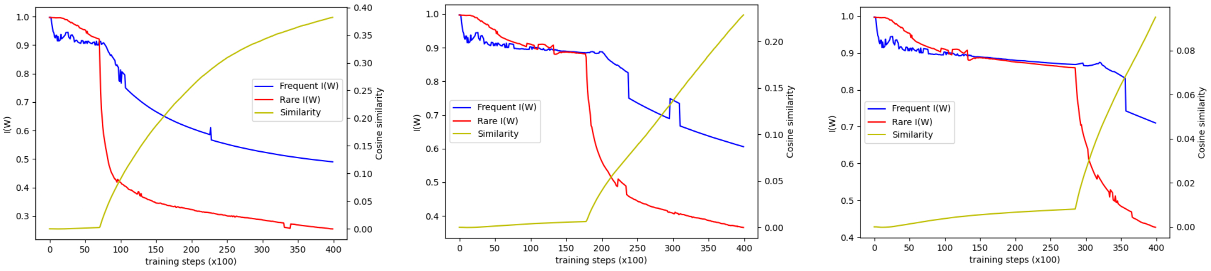
2.2 Embedding Problems in Neural Language Models

Recent studies on the geometric properties of contextual embedding space have observed that the distribution of embedding vectors is far from isotropic and occupies a relatively narrow cone space(Mu et al., 2018; Liu et al., 2019; Zhou et al., 2019; Ethayarajh, 2019;). Gao et al. (2019) named this phenomenon the *representation degeneration problem*. This degeneration problem results in an increase in the overall cosine similarity between token embeddings, making it difficult for the model to learn semantic relationships between tokens. Demeter et al. (2020) demonstrated that the norm information of the token embeddings is so dominant that angle information about the feature vector is ignored when calculating the logits in the output layer. Owing to this structural weakness of the embedding space, embeddings with small norms are always assigned with a low probability, which reduces the diversity of the text generated by the model. Anisotropy of the embedding space is still problem for the pre-trained large language models, and language models with improved isotropic embedding space performs well in downstream tasks(Biś et al., 2021; Rajaee and Pilehvar, 2021).

Although the problem has been theoretically analyzed in several studies, existing methods are based on the observed phenomena as a result of the problem. To mitigate the phenomena observed from

Methods	PPL ↓				I(W) ↑			
	Freq	Med	Rare	Total	Freq	Med	Rare	Total
MLE	16.58	224.24	813.76	20.77	0.426	0.286	0.198	0.293
Freeze	16.48	233.92	3017.53	20.78	0.840	0.651	0.831	0.739

Table 1: Perplexity and $I(\mathbf{W})$ for each token groups. Lower is better for PPL and higher is better for $I(\mathbf{W})$.



(a) freeze until step 7k

(b) freeze until step 18k

(c) freeze until step 29k

Figure 2: Plot of $I(\mathbf{W})$ for rare and frequent groups and average cosine similarity between rare and frequent embeddings when freezing the training of rare tokens until specific training steps.

the problem, the post-processing of the embedding vectors(Mu et al., 2018; Biś et al., 2021) or regularization terms about the phenomena(Gao et al., 2019; Wang et al., 2019; Wang et al., 2020; Zhang et al., 2020) were introduced. These methods are applied to all token embeddings, so there is the problem of over regularization for the embeddings whose semantic relationship is trained well. Also, methodologies based on the training dynamics of the token embeddings concerning the degeneration problem remain subject to study.

Frequency bias in embedding space is another problem. Ott et al. (2018) conducted a comprehensive study on the under-estimation of rare tokens in neural machine translation. Gong et al. (2018) observed that embeddings in the language model were biased towards frequency and proposed an adversarial training scheme to address this problem.

3 Empirical Study: Token Embedding Training Dynamics led by Rare Tokens

3.1 Initial Training Dynamics of Embeddings

To analyze the training procedure of token embeddings, we train a Transformer language model at the WikiText-103 dataset from scratch. Whole vocabulary tokens are divided into three groups: frequent, medium, and rare groups. Based on the appearance frequency in the training corpus, the 30%, 50%, and 20% tokens are assigned to the frequent, medium, and rare group. We visualize the initial training dynamics of these groups via the projection of the embeddings into 2D, using singular value decomposition (SVD) projection. As illustrated in Figure 1, rare groups degenerate first, as they emerge from the entire embedding distribution. Subsequently, other groups also start to degenerate, following the degeneration of the rare group. Based on this observation, we hypothesize that *the degeneration of rare token embeddings induces the degeneration of non-rare token embeddings*.

3.2 Rare Tokens Degenerate Non-Rare Tokens

Because Transformer(Vaswani et al., 2017) is representative of the current language models, we adopt the 6-layer Transformer decoder model architecture for an empirical study on the training dynamics of embedding vectors. The model is trained in language modeling task using WikiText-103 dataset(Merity et al., 2018). Experimental details regarding the model and training hyperparameter configurations can be found in the Appendix B. To verify the hypothesis of the previous subsection, we train a model while freezing the rare group token embeddings in their initial states during training, and compare it to the baseline model, where all embeddings are trained with negative log-likelihood loss. In addition, we train the models of various settings relative to freezing steps and examine whether the degeneration of rare token embeddings depends on when training of rare embeddings begins.

The performance of the models is evaluated in two ways; the likelihood and isotropy of token embeddings. Perplexity(Bengio et al., 2003) is

Methods	**PPL** ↓				**I(W)** ↑			
	Freq	Med	Rare	Total	Freq	Med	Rare	Total
MLE	16.58	224.24	813.76	20.77	0.426	0.286	0.198	0.293
Freeze (b) & (c)	17.41	247.89	66.41	21.79	0.323	0.693	0.551	0.536
Freeze (b)	16.99	240.72	65.76	21.26	0.495	0.561	0.678	0.748
Freeze (c)	16.61	220.07	645.24	20.76	0.443	0.276	0.15	0.317

Table 2: Perplexity and $I(\mathbf{W})$ for each token group at gradient partial freezing experiment.

adopted to evaluate the performance of the likelihood of the model. To measure the isotropy of the token embedding distribution, we adopt the partition function $Z(\mathbf{a}) = \sum_{i=1}^{N} \exp\left(\mathbf{w}_i \mathbf{a}^T\right)$ defined in Arora et al. (2016), where $\mathbf{w}_i$ denotes the embedding vector of token i, and $\mathbf{a}$ represents a unit vector. Lemma 2.1. in Arora et al. (2016) demonstrate that if the embedding vectors are isotropic, $Z(\mathbf{a})$ is approximately constant. Based on this property, we measure the isotropy of an embedding matrix $\mathbf{W}$ using $I(\mathbf{W})$, which is defined as follows.

$$I(\mathbf{W}) = \frac{\min_{\mathbf{a}\in\mathbf{X}} Z(\mathbf{a})}{\max_{\mathbf{a}\in\mathbf{X}} Z(\mathbf{a})}, \tag{3}$$

where $I(\mathbf{W}) \in [0, 1]$ and $\mathbf{X}$ represents the set of eigenvectors of $\mathbf{W}^T\mathbf{W}$ (Mu et al., 2018; Wang et al., 2020; Biś et al., 2021). Furthermore, we measure the relatedness between the rare and frequent group token embeddings to verify that the degeneration of the frequent group follows the degeneration of the rare group. We calculate the average cosine similarity between the rare and frequent group embeddings to measure the relatedness.

Table 1 shows the comparison of the baseline model and the model with frozen rare tokens. We denote the baseline as "MLE" and the freezing method as "Freeze". Surprisingly, the PPL of frequent group tokens and overall $I(\mathbf{W})$ improved by simply not training the rare token embeddings. Figure 2 illustrates the change in $I(\mathbf{W})$ for the frequent and rare token embeddings, including the similarity between frequent and rare token embeddings at various freezing step settings. Whenever the rare token embeddings start to be trained, their $I(\mathbf{W})$ decreases steeply, followed by decreasing $I(\mathbf{W})$ of frequent embeddings and increasing similarities between the frequent and rare embeddings. From the analysis in this subsection, we demonstrate that the entire degeneration problem can be solved by solely handling just rare embeddings during the entire training procedure.

3.3 Finding the Primary Cause of the Degeneration Problem: From the Gradient

With T context feature vectors $\mathbf{h}_i$ ($i \in [1, T]$) from the training sample, the negative log-likelihood loss gradient for the rare token embedding $\mathbf{w}_r$ is calculated as follows.

$$\begin{aligned} \nabla_{\mathbf{w}_r} L_{NLL} = & \underbrace{\sum_{y_i = v_r} (p_{r|i} - 1)\mathbf{h}_i}_{(a)} \\ & + \underbrace{\sum_{y_j \notin V_r} p_{r|j}\mathbf{h}_j}_{(b)} + \underbrace{\sum_{y_k \in V_r} p_{r|k}\mathbf{h}_k}_{(c)}, \end{aligned} \tag{4}$$

where y_i denotes the target token for $\mathbf{h}_i$, V_r is the rare token vocabulary group, and $p_{r|i}$ represents the conditional probability of token v_r given $\mathbf{h}_i$, which is calculated as $[\text{softmax}(\mathbf{h}_i\mathbf{W}^T)]_r$. We divide the gradient for $\mathbf{w}_r$ to 3 parts in Eq. 4. Part (a) pulls $\mathbf{w}_r$ close to the feature vectors whose target tokens are v_r. Part (b) pushes away $\mathbf{w}_r$ from the feature vectors whose target tokens are not rare. Part (c) pushes away $\mathbf{w}_r$ from the feature vectors whose target tokens are rare. As an extension of the analysis in the previous subsection, we freeze these parts of the gradient with various settings during training to identify the key cause of the degeneration problem. In other words, depending on the settings, the specific gradient parts that will not be used for embedding training is detached from the computation graph during training stage. This can be easily implemented by `detach()` function of `Pytorch` (Paszke et al., 2019). All model and training configurations are the same as in the previous sections, except those to be frozen.

Table 2 presents the results of the experiments in this subsection. We freeze the parts of the gradient for the rare tokens with three settings. Because part (a) is a key component required to train the token embedding to be aligned to the target, all settings activate part (a). We notice that when part (b) is activated (solely freezing part (c)), $I(\mathbf{W})$ decreases and PPL for rare tokens increases almost 10 times compared to when part (b) is frozen. Because activating part (c) is not seen to be negative for PPL and $I(\mathbf{W})$, we conclude that part (b) of Eq. 4 is the bedrock cause for the degeneration problem. From the analysis in this section, we demonstrate that *the entire degeneration problem can be solved by mainly addressing the part of the gradient for rare embeddings that pushes away rare token embeddings from non-rare feature vectors*.

4 Method

4.1 Dynamic Rare Token Grouping

To handle the specific part of the gradient for the rare token embeddings studied in the previous section, we need to properly group the rare tokens. A naive approach can be used to group rare tokens based on the appearance frequency of the training corpus, as described in the previous section. However, this static grouping method is suboptimal because the model is typically trained via mini-batch training. The group of rare tokens that appeared less frequently in recent batch samples is variable in the mini-batch training. Therefore, it is necessary to dynamically group rare tokens based on token appearances in recent batch samples.

To consider the token appearances in recent batch samples, we introduce the token counter memory that remembers the number of the appearances of each token during the previous K training steps. For K memories, $[\mathbf{m}_1, ..., \mathbf{m}_K]$, $\mathbf{m}_t \in \mathbb{R}^N$ represents the number of appearances of each token of N-size vocabulary at the t-th previous training step. Memories are set as zero vectors at the initial stage. At each training step, the token appearance, $\mathbf{a} \in \mathbb{R}^N$, is calculated as the sum of all K memories: $\mathbf{a} = \sum_{t=1}^{K} \mathbf{m}_t$. Based on $\mathbf{a}$, we determine whether token i is in the rare token group V_r as follows.

$$\begin{aligned} \frac{a_i}{K} < \alpha \Rightarrow v_i \in V_r \\ \frac{a_i}{K} \geq \alpha \Rightarrow v_i \notin V_r, \end{aligned} \tag{5}$$

where a_i is the i-th component of $\mathbf{a}$, and α is a hyper-parameter in our method that controls the proportion of rare tokens in the entire vocabulary. In this study, we set K to the number of iteration steps during one epoch of training stage.

4.2 Adaptive Gradient Gating for Rare Tokens

After dynamically grouping the rare tokens at each training step, we need to handle a specific part of the gradient for the rare token embeddings to solve the degeneration problem of all embeddings. To solely control the gradient for rare token embeddings, we introduce a *gradient gating* method for a parameter $\mathbf{x}$. We define $\tilde{\mathbf{x}}$ as a tensor whose value is the same as $\mathbf{x}$, but detached from the current training graph. This implies that $\tilde{\mathbf{x}}$ is considered a constant, hence, gradient about $\tilde{\mathbf{x}}$ does not exist. In practice, $\tilde{\mathbf{x}}$ can be easily obtained from $\mathbf{x}$ using the `detach()` function of `Pytorch` (Paszke et al., 2019). With $\tilde{\mathbf{x}}$, we can gate the gradient for $\mathbf{x}$ as follows.

$$\begin{aligned} \mathbf{x}_{gated} = \mathbf{g} \odot \mathbf{x} + (1 - \mathbf{g}) \odot \tilde{\mathbf{x}} \\ \nabla_{\mathbf{x}} f(\mathbf{x}_{gated}) = \mathbf{g} \odot \nabla_{\mathbf{x}} f(\mathbf{x}), \end{aligned} \tag{6}$$

where $\mathbf{x}_{gated}$ is a new parameter whose value is the same as $\mathbf{x}$, and $\mathbf{g} \in [0, 1]$ is a gate tensor. When the $\mathbf{x}_{gated}$ is fed to the function $f(\cdot)$ as input, the gradient for $\mathbf{x}$ is gated by $\mathbf{g}$.

As we described in section 3, part (b) of Eq. 4 should mainly be handled to solve the degeneration problem. To address part (b) of Eq. 4, given a context feature vector of the i-th position $\mathbf{h}_i$, we introduce a gate vector $\mathbf{g}_1 \in \mathbb{R}^N$ as follows.

$$g_{1k} = \begin{cases} a_k/K & \text{if } v_k \in V_r, v_k \neq y_i \\ 1 & \text{else ,} \end{cases} \tag{7}$$

where g_{1k} denotes a k-th component of $\mathbf{g}_1$. $\mathbf{g}_1$ controls the degree to which rare token embeddings move away from non-rare feature vectors whose targets differ from each rare token embedding. Also, each component of $\mathbf{g}_1$ is calculated based on the rarity of each rare token, a_k, so gradient gating for part (b) of Eq. 4 is adaptive for each rare tokens.

Although part (c) of Eq. 4, which pushes embeddings away from the feature vectors whose targets are other rare tokens, is not to be seen as the cause of the degeneration problem in section 3, this part also induces the degeneration problem for the certain situation when rare tokens degenerate other rare tokens. To address this, we approximate the multiple levels of rarity in the rare token group to two levels in this paper: 'less rare' and 'very rare'. We define the two rarity levels based on the average number of appearances of the entire rare tokens: if the token appearance a_k is smaller than the mean

Methods	**PPL** ↓				**Uniq** ↑				**I(W)**↑
	Freq	Med	Rare	Total	Freq	Med	Rare	Total	
MLE	**13.30**	146.47	438.67	15.51	**9107**	3945	91	13143	0.377
AGG	13.35	**146.44**	**75.39**	15.51	9105	**4287**	**345**	**13737**	**0.813**
Human	–	–	–	–	10844	7146	300	18920	–

Table 3: Experimental results for each token group in WikiText-103 language modeling task comparing MLE baseline and AGG.

Methods	**PPL** ↓				**Uniq** ↑				**I(W)**↑
	Freq	Med	Rare	Total	Freq	Med	Rare	Total	
UL	**14.05**	**125.17**	385.6	**16.17**	9527	4402	97	14026	0.396
UL + AGG	14.17	125.93	**71.48**	16.25	**9625**	**4884**	**453**	**14962**	**0.654**
Human	–	–	–	–	10844	7146	300	18920	–

Table 4: Experimental results for each token group in WikiText-103 language modeling task comparing UL and UL+AGG.

of a_r where $r \in V_r$, corresponding token is a very rare token. For the very rare token embeddings, part (c) of the gradient about embeddings pushes them away from the feature vectors whose targets are less rare tokens that are relatively frequent compared to them. This means that part (c) roles like part (b) in the above situation, which becomes the cause of the degeneration problem. Therefore, we need to handle part (c) of Eq. 4 for very rare tokens. To address part (c) of Eq. 4 for the very rare token embeddings, we introduce another gate vector $\mathbf{g}_2 \in \mathbb{R}^N$ as follows.

$$g_{2k} = \begin{cases} min(\frac{a_k}{\bar{a}_r}, 1) & \text{if } v_k \in V_r, v_k \neq y_i \\ 1 & \text{else,} \end{cases} \tag{8}$$

where g_{2k} is the k-th component of $\mathbf{g}_2$ and $\bar{a}_r$ is the mean of a_r where $r \in V_r$. $\mathbf{g}_2$ controls the degree to which very rare token embeddings move away from less rare feature vectors whose targets differ from each very rare token embedding. Also, each component of $\mathbf{g}_2$ is calculated based on the rarity of each very rare token, a_k, so gradient gating for part (c) of Eq. 4 is adaptive for each very rare tokens.

To calculate the loss of $\mathbf{h}_i$, we calculate three logits, $\mathbf{z}_i^0, \mathbf{z}_i^1$, and $\mathbf{z}_i^2$, as follows.

$$\begin{aligned} \mathbf{z}_i^0 &= \mathbf{h}_i \tilde{\mathbf{W}}^T \\ \mathbf{z}_i^l &= \mathbf{g}_l \odot \tilde{\mathbf{h}}_i \mathbf{W}^T + (1 - \mathbf{g}_l) \odot \tilde{\mathbf{h}}_i \tilde{\mathbf{W}}^T, \end{aligned} \tag{9}$$

where $\mathbf{W}$ denotes an embedding matrix, and $l = 1, 2$. Because our method solely handles the gradient for embeddings, we calculate $\mathbf{z}_i^0$ for a gradient about $\mathbf{h}_i$, which does not need to be gated. Finally, the negative log-likelihood loss for i-th position L_i is computed as follows.

$$\begin{aligned} L_i = &- \log p^0_{I(y_i)|i} \\ &- \mathbb{1}(y_i \notin V_r) \log p^1_{I(y_i)|i} \\ &- \mathbb{1}(y_i \in V_r) \log p^2_{I(y_i)|i}, \end{aligned} \tag{10}$$

where $p^m_{I(y_i)|i}$ = $[\text{softmax}(\mathbf{z}_i^m)]_{I(y_i)}$ with m=$0, 1, 2$ and $\mathbb{1}(\cdot)$ denotes the Indicator function. Derivation of the gradient for rare token embeddings, $\nabla_{\mathbf{w}_r} L_i$, is provided in Appendix A.

5 Experiments

We evaluate our method on various tasks including language modeling, word similarity, and machine translation. In the language modeling task, we focus on verifying the diversity of the generated texts. We test the learning of the semantic relationships between tokens on the word similarity task. Finally, we evaluate the quality of generated texts on the machine translation task. For all the experimental results below, we adopt the state-of-the-art model architecture as a baseline to properly demonstrate the effectiveness of our method. Every detail on the experiment, such as model hyper-parameters and training configurations, regard the reproducibility are provided in Appendix B.

5.1 Language Modeling

Setting We conduct experiments using WikiText-103 dataset, which is a significantly large dataset for language modeling task with approximately

103M words and 260K vocabulary size (Merity et al., 2018). Texts in the dataset are preprocessed based on the byte-pair encoding(Sennrich et al., 2016). We adopt the GPT-2 medium architecture(Radford et al., 2019), which comprises 24 Transformer decoder layers as a baseline model. Because our method is about learning token embeddings, we train the models from scratch for a maximum of 50k iterations and evaluate them based on the perplexity of the validation set. For hyper-parameter searching, we select $\alpha \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$ for AGG method on the language modeling task. The hyper-parameter sensitivity for the AGG are given in Appendix E.

We use three quantitative metrics to evaluate our method: Perplexity, Uniq, and $I(\mathbf{W})$. Related to the likelihood of generated texts, Perplexity quantifies the prediction difficulty over the next token. Uniq (Welleck et al., 2020) quantify the number of unique next-token predictions, measuring the token diversity. As described in section 3, $I(\mathbf{W})$ measures the isotropy of the token embedding space.

Results We present our results for the testset in Table 3. We denote the baseline method as ’MLE’ and our method as ’AGG’. We measure Perplexity and Uniq for each token group defined in Section 3. As presented in Table 3, AGG improves the overall metrics for the medium and rare groups while maintaining performance for the frequent token group. This shows that our method not only improves the quality of rare token embeddings, but also the quality of non-rare token embeddings. In particular, for the rare group, the Perplexity score decrease significantly and the number of unique predictions surpasses the human distribution. The $I(\mathbf{W})$ for all token embeddings increased over 2 times the baseline. Experimental results of $I(\mathbf{W})$ for the embeddings of each frequency groups can be found in Appendix C. We also show examples of generated texts in Appendix G.

Compatibility Neural text degeneration problem is another problem in neural text generative models, where the model generates texts that are less likely to match human word distributions. Existing methods for this problem focus on the diversity of the generated texts by adding an auxiliary loss to the original negative log-likelihood loss (Welleck et al., 2020). Although Welleck et al. (2020) and AGG attempts to address the same problem about diversity, AGG can be compatible with the existing method in the text degeneration problem because AGG does not alter the form of the loss function in MLE training. Table 4 presents the results of the experiments about fusion of unlikelihood training(Welleck et al., 2020) and AGG. We denote the unlikelihood training as UL. From table 4, we notice that when UL and AGG are fused, it produces a synergistic effect that exceeds the gain of each for the baseline. This indicates that AGG is compatible with methods that address other problems in text generation.

Datasets	MLE	AGG
MEN	33.57	**55.13**
WS353	47.51	**56.54**
RG65	35.48	**65.45**
RW	32.13	**36.36**

Table 5: Performance(Spearman’s $\gamma \times 100$) of the models on the four word similarity datasets.

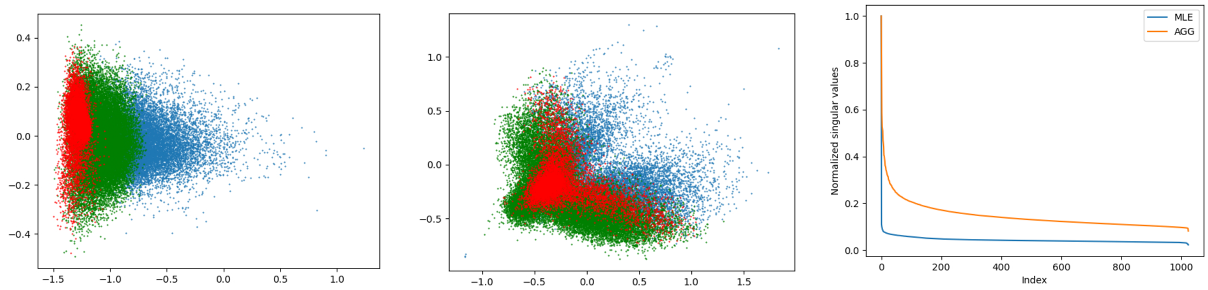
Methods	BLEU ↑	
	Base	Big
Transformer (Vaswani et al., 2017)	27.30	28.40
CosReg (Gao et al., 2019)	28.38	28.94
Adv MLE (Wang et al., 2019)	28.43	29.52
SC (Wang et al., 2020)	28.45	29.32
AGG	**28.70**	**29.81**

Table 6: Comparison of different methods in terms of BLEU scores.

5.2 Word Similarity

Setting We evaluate the semantic relationship between tokens for AGG and the baseline with four word similarity datasets: MEN, WS353, RG65, and RW(Bruni et al., 2014; Agirre et al., 2009; Rubenstein and Goodenough, 1965; Luong et al., 2013). Methods are tested whether the similarity between the given two words in the embedding space is consistent with the ground truth, in terms of Spearman’s rank correlation. We adopt cosine distance to compute the similarity between embeddings. We use the same models trained on language modeling tasks with the WikiText-103 dataset for the word similarity task.

Results Table 5 presents the result obtained from the evaluation of the word similarity task. From this table, it can be observed that our method outperforms the baseline on overall datasets. Although AGG handles only training of rare tokens, the semantic relationships between all tokens are also well learned. Qualitative studies on semantic align-



(a) MLE

(b) AGG

(c) Singular value decay

Figure 3: (a), (b) Token embedding visualization for the baseline model and AGG on the language modeling task with WikiText-103; (c) Normalized singular value for MLE and AGG.

ment between tokens are provided in Appendix F.

5.3 Machine Translation

Setting We utilize a dataset from standard WMT 2014 containing 4.5M English→German sentence pairs. The source and target sentences are encoded by 37K shared tokens based on byte-pair encoding(Sennrich et al., 2016). We adopt the two version of Transformer(Vaswani et al., 2017) as the baseline model for applying our method: `base` and `big`. The model configuration is the same as that proposed in Vaswani et al. (2017). To evaluate the quality of the generated texts, we measure BLEU score (Papineni et al., 2002), which is standard metric for machine translation task.

Results Table 6 presents a comparison of our method and other methods in terms of the BLEU score. Our method achieves 1.4 and 1.41 BLEU score improvements on the machine translation task for the `base` and `big` baseline models. In addition, our method is better than all other previous works in handling the representation degeneration problem that reported BLEU scores in the same tasks. These results demonstrate the effectiveness of AGG in the quality of the generated texts. While other methods addressing the degeneration problem targets all token embeddings, target of AGG, rare token embeddings, are optimized based on the analysis about the training dynamics of token embeddings. Due to this difference, our method can prevent the over regularization problem for frequent token embeddings, which is the main advantage of AGG compared to other works. Qualitative study about cross-lingual semantic alignment between tokens of the source and target languages is provided in Appendix F.

6 Analysis of AGG

Figure 3 (a) and (b) present the visualizations of the embedding space of baseline MLE and our method. In the figure, applying the AGG method restores the isotropy of the token embedding space. In addition, we observe that the regions occupied by each token group are not disjoint when applying AGG. For baseline, the regions occupied by rare group and the frequent group are disjoint, which is refered as the frequency bias problem of embeddings (Gong et al., 2018). From the analysis of the visualization of the embedding space, we notice that the manipulating the training of the rare token embeddings can alleviate the frequency bias problem. Figure 3 (c) presents the plot of the normalized singular value of embedding matrix for MLE and AGG. Slowly decaying singular values of AGG demonstrate an isotropic distribution of the embedding space. Ablation studies about the gating terms and dynamic rare token grouping can be found in Appendix D.

7 Conclusion

In this study, we analyzed the training dynamics of the token embeddings concerning the representation degeneration problem of the learned embeddings, focusing on the rare tokens. Based on the analysis, we propose an adaptive gradient gating method that solves the problem by solely handling the training for rare token embeddings. Experiments and qualitative studies in various tasks of text generation demonstrate the effectiveness of our method. Beyond the two-level approximation of rarity of rare tokens which is applied to our study, addressing multiple levels of rarity can be an interesting region to study for the future work.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. 2021. Too much in common: Shifting of embeddings in transformer language models and its implications. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*.

David Demeter, Gregory Kimmel, and Doug Downey. 2020. Stolen probability: A structural weakness of neural language models. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Jun Gao, Di He, X. Tan, Tao Qin, L. Wang, and T. Liu. 2019. Representation degeneration problem in training natural language generation models. *ArXiv*, abs/1907.12009.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Chengyue Gong, Di He, X. Tan, Tao Qin, L. Wang, and T. Liu. 2018. Frage: Frequency-agnostic word representation. *ArXiv*, abs/1809.06858.

A. Graves. 2013. Generating sequences with recurrent neural networks. *ArXiv*, abs/1308.0850.

Tianlin Liu, L. Ungar, and João Sedoc. 2019. Unsupervised post-processing of word vectors via conceptor negation. In *AAAI*.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, Sofia, Bulgaria.

Stephen Merity, N. Keskar, and R. Socher. 2018. Regularizing and optimizing lstm language models. *ArXiv*, abs/1708.02182.

Stephen Merity, Caiming Xiong, James Bradbury, and R. Socher. 2017. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

Jiaqi Mu, S. Bhat, and P. Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. *ArXiv*, abs/1702.01417.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. *ArXiv*, abs/1803.00047.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *ArXiv*, abs/1912.01703.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *EACL*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

S. Rajaee and Mohammad Taher Pilehvar. 2021. A cluster-based approach for improving isotropy in contextual embedding space. In *ACL/IJCNLP*.

Herbert Rubenstein and John Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8:627–633.

Rico Sennrich, B. Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6555–6565. PMLR.

L. Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020. Improving neural language generation with spectrum control. In *ICLR*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. *ArXiv*, abs/1908.04319.

Z. Yang, Zihang Dai, R. Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank rnn language model. *ArXiv*, abs/1711.03953.

Z. Zhang, Chongming Gao, Cong Xu, Rui Miao, Qinli Yang, and Junming Shao. 2020. Revisiting representation degeneration problem in language modeling. In *EMNLP*.

Tianyuan Zhou, João Sedoc, and Jordan Rodu. 2019. Getting in shape: Word embedding subspaces. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5478–5484. International Joint Conferences on Artificial Intelligence Organization.

A Derivation of the gradient of AGG loss *w.r.t.* rare token embedding

We follow the same notation as in the main paper. Before we write the derivation of the gradient about rare token embedding $\mathbf{w}_r$, we write the gradient of $f(\tilde{\mathbf{w}}_j)$ and $(z_i^l)_j$ about $\mathbf{w}_r$, where $f(\tilde{\mathbf{w}}_j)$ is the function of $\tilde{\mathbf{w}}_j$ with $j = 1, ..., N$ and $(z_i^l)_j$ is a j-th component of $\mathbf{z}_i^l$ with $l = 0, 1, 2$ as follows.

$$\begin{aligned}
\nabla_{\mathbf{w}_r} f(\tilde{\mathbf{w}}_j) &= \nabla_{\tilde{\mathbf{w}}_j} f(\tilde{\mathbf{w}}_j) \odot \nabla_{\mathbf{w}_r} \tilde{\mathbf{w}}_j \\
&= \nabla_{\tilde{\mathbf{w}}_j} f(\tilde{\mathbf{w}}_j) \odot 0 \\
&= 0 \text{ for all } j \\
&(\because \tilde{\mathbf{w}}_j \text{ is treated as a constant.})
\end{aligned} \tag{11}$$

$$\begin{aligned}
\nabla_{\mathbf{w}_r} (z_i^l)_j &= \nabla_{\mathbf{w}_r} [g_{lj} \cdot \tilde{\mathbf{h}}_i \mathbf{w}_j^T + (1 - g_{lj} \cdot \tilde{\mathbf{h}}_i \mathbf{w}_j^{\tilde{T}})] \\
&= g_{lj} \nabla_{\mathbf{w}_r} \tilde{\mathbf{h}}_i \mathbf{w}_j^T + 0 \\
&= \begin{cases} g_{lj} \tilde{\mathbf{h}}_i & \text{if } j = r \\ 0 & \text{else} \end{cases} \\
&= \begin{cases} g_{lj} \mathbf{h}_i & \text{if } j = r \\ 0 & \text{else} \end{cases} \\
&(\because \mathbf{h}_i = \tilde{\mathbf{h}}_i \text{ in terms of value.})
\end{aligned} \tag{12}$$

Considering the case of $y_i \notin V_r$, AGG negative log-likelihood loss for the i-th position of token generation, L_i^{AGG} is written as follows.

$$L_i^{AGG} = -\log p^0_{I(y_i)|i} - \log p^1_{I(y_i)|i} \tag{13}$$

Then gradient of L_i^{AGG} about $\mathbf{w}_r$ is written as follows.

$$\begin{aligned}
&\nabla_{\mathbf{w}_r} L_i^{AGG} \\
&= -\nabla_{\mathbf{w}_r} \log p^0_{I(y_i)|i} - \nabla_{\mathbf{w}_r} \log p^1_{I(y_i)|i} \\
&= -\nabla_{\mathbf{w}_r} \log p^1_{I(y_i)|i} - 0 \\
&(\because \log p^0_{I(y_i)|i} \text{ is a function of } \tilde{\mathbf{w}}_r.) \\
&= -\frac{1}{p^1_{I(y_i)|i}} \nabla_{\mathbf{w}_r} p^1_{I(y_i)|i} \\
&= -\frac{1}{p^1_{I(y_i)|i}} \sum_{j=1}^{N} \nabla_{(z_i^1)_j} p^1_{I(y_i)|i} \cdot \nabla_{\mathbf{w}_r} (z_i^1)_j \\
&(\because p^1_{I(y_i)|i} \text{ is a function of } (z_i^1)_j, j = 1, ..., N.) \\
&= -\frac{1}{p^1_{I(y_i)|i}} \nabla_{(z_i^1)_r} p^1_{I(y_i)|i} \cdot \nabla_{\mathbf{w}_r} (z_i^1)_r \\
&(\text{By Eq. } 12.)
\end{aligned} \tag{14}$$

As $p^1_{I(y_i)|i} = [\text{softmax}(\mathbf{z}_i^1)]_{I(y_i)|i}$,

$$\nabla_{(z_i^1)_r} p^1_{I(y_i)|i} = -p^1_{I(y_i)|i} p^1_{r|i}. \tag{15}$$

Thus, $\nabla_{\mathbf{w}_r} L_i^{AGG}$ is computed as follows.

$$\begin{aligned}
&\nabla_{\mathbf{w}_r} L_i^{AGG} \\
&= -\frac{1}{p^1_{I(y_i)|i}} \nabla_{(z_i^1)_r} p^1_{I(y_i)|i} \cdot \nabla_{\mathbf{w}_r} (z_i^1)_r \\
&(\text{By Eq. } 14.) \\
&= p^1_{r|i} \cdot \nabla_{\mathbf{w}_r} (z_i^1)_r \\
&= g_{1r} p^1_{r|i} \mathbf{h}_i \\
&(\text{By Eq. } 12.)
\end{aligned} \tag{16}$$

Considering the case of $y_i \in V_r$ but $y_i \neq v_r$, L_i^{AGG} is written as follows.

$$L_i^{AGG} = -\log p^0_{I(y_i)|i} - \log p^2_{I(y_i)|i} \tag{17}$$

Then $\nabla_{\mathbf{w}_r} L_i^{AGG}$ is written as follows.

$$\begin{aligned}
&\nabla_{\mathbf{w}_r} L_i^{AGG} \\
&= -\nabla_{\mathbf{w}_r} \log p^0_{I(y_i)|i} - \nabla_{\mathbf{w}_r} \log p^2_{I(y_i)|i} \\
&= -\nabla_{\mathbf{w}_r} \log p^2_{I(y_i)|i} - 0 \\
&(\because \log p^0_{I(y_i)|i} \text{ is a function of } \tilde{\mathbf{w}}_r.) \\
&= -\frac{1}{p^2_{I(y_i)|i}} \nabla_{\mathbf{w}_r} p^2_{I(y_i)|i} \\
&= -\frac{1}{p^2_{I(y_i)|i}} \sum_{j=1}^{N} \nabla_{(z_i^2)_j} p^2_{I(y_i)|i} \cdot \nabla_{\mathbf{w}_r} (z_i^2)_j \\
&(\because p^2_{I(y_i)|i} \text{ is a function of } (z_i^2)_j, j = 1, ..., N.) \\
&= -\frac{1}{p^2_{I(y_i)|i}} \nabla_{(z_i^2)_r} p^2_{I(y_i)|i} \cdot \nabla_{\mathbf{w}_r} (z_i^2)_r \\
&(\because \text{Eq. } 12.)
\end{aligned} \tag{18}$$

As $p^2_{I(y_i)|i} = [\text{softmax}(\mathbf{z}_i^2)]_{I(y_i)|i}$,

$$\nabla_{(z_i^2)_r} p^2_{I(y_i)|i} = -p^2_{I(y_i)|i} p^2_{r|i}. \tag{19}$$

Thus, $\nabla_{\mathbf{w}_r} L_i^{AGG}$ is computed as follows.

$$\begin{aligned}
&\nabla_{\mathbf{w}_r} L_i^{AGG} \\
&= -\frac{1}{p^2_{I(y_i)|i}} \nabla_{(z_i^2)_r} p^2_{I(y_i)|i} \cdot \nabla_{\mathbf{w}_r} (z_i^2)_r \\
&(\text{By Eq. } 18.) \\
&= p^2_{r|i} \cdot \nabla_{\mathbf{w}_r} (z_i^2)_r \\
&= g_{2r} p^2_{r|i} \mathbf{h}_i \\
&(\text{By Eq. } 12.)
\end{aligned} \tag{20}$$

Considering the remained case of $y_i = v_r$, since $y_i \in V_r$, L_i^{AGG} is same as the second case, and derivation process of $\nabla_{\mathbf{w}_r} L_i^{AGG}$ shares the same process with Eq. 18. As $I(y_i) = r$,

$$\nabla_{(z_i^2)_r} p^2_{I(y_i)|i} = p^2_{I(y_i)|i}(1 - p^2_{I(y_i)|i}) \tag{21}$$

Thus, $\nabla_{\mathbf{w}_r} L_i^{AGG}$ is computed as follows.

$$\begin{aligned}
&\nabla_{\mathbf{w}_r} L_i^{AGG} \\
&= -\frac{1}{p^2_{I(y_i)|i}} \nabla_{(z_i^2)_r} p^2_{I(y_i)|i} \cdot \nabla_{\mathbf{w}_r} (z_i^2)_r \\
&\text{(By Eq. 21.)} \\
&= -(1 - p^2_{I(y_i)|i}) \cdot \nabla_{\mathbf{w}_r} (z_i^2)_r \\
&= -g_{2r}(1 - p^2_{I(y_i)|i})\mathbf{h}_i \\
&\text{(By Eq. 12.)} \\
&= (p^2_{r|i} - 1)\mathbf{h}_i \\
&(\because I(y_i) = r \text{ and } g_{2r} = 1 \text{ if } I(y_i) = r.)
\end{aligned} \tag{22}$$

As $p_{r|i} = p^m_{r|i}$ with $m = 0, 1, 2$ in terms of value, we finally write $\nabla_{\mathbf{w}_r} L_i^{AGG}$ as follows.

$$\nabla_{\mathbf{w}_r} L_i = \begin{cases} (p_{r|i} - 1)\mathbf{h}_i & \text{if } y_i = v_r \\ g_{1r} p_{r|i} \mathbf{h}_i & \text{if } y_i \notin V_r \\ g_{2r} p_{r|i} \mathbf{h}_i & \text{else,} \end{cases} \tag{23}$$

B Experimental Details

In this section, we present the details of the experiments in main page. All the experiments were conducted with a single GPU on our machine (GPU: NVIDIA A40) and from single run. For each task in the experiments, we use the same model architecture and train it with different objectives(*i.e.,* MLE, AGG, UL). The hyper-parameters used for different training methods in the same task are exactly same. The detailed hyper-parameters are described in Table 11.

C Experimental Results of I(W) for each frequency groups

In this section, we present the experimental results about $I(\mathbf{W})$ for the embeddings of each frequency groups. Table 7 shows the $I(\mathbf{W})$ comparing MLE baseline and AGG. Table 8 shows the $I(\mathbf{W})$ comparing UL baseline and the fusion of UL and AGG. As presented in Table 7 and 8, AGG improves isotropy of the embedding space for all frequency groups, indicating that our method solves the whole degeneration problem.

Methods	$\mathbf{I}(\mathbf{W})\uparrow$		
	Freq	Med	Rare
MLE	0.51	0.33	0.278
AGG	**0.702**	**0.714**	**0.813**

Table 7: Experimental results about $\mathbf{I}(\mathbf{W})$ for each token group in WikiText-103 language modeling task comparing MLE baseline and AGG.

Methods	$\mathbf{I}(\mathbf{W})\uparrow$		
	Freq	Med	Rare
UL	0.533	0.351	0.293
UL + AGG	**0.731**	**0.626**	**0.696**

Table 8: Experimental results about $\mathbf{I}(\mathbf{W})$ for each token group in WikiText-103 language modeling task comparing UL baseline and UL + AGG.

D Ablation Study about AGG

In this section, we present the ablation studies about AGG. In our method, AGG, we introduce two gate vectors, $\mathbf{g}_1$, and $\mathbf{g}_2$, to handle the gradient for rare and very rare token embeddings. We conduct experiments on these gate vectors. Table 9 presents the results of the ablation studies compared with the MLE and AGG. When $\mathbf{g}_1$ is excluded from AGG (denoted as 'no $\mathbf{g}_1$'), Uniq and $I(\mathbf{W})$ decreased significantly, because $\mathbf{g}_1$ is the key component for the gradient gating. When $\mathbf{g}_2$ is excluded from AGG (denoted as 'no $\mathbf{g}_2$'), Uniq and $I(\mathbf{W})$ slightly decrease. Accordingly, we notice that $\mathbf{g}_2$ is important for the gating of gradients fort the very rare token embeddings.

Also, we present the analysis about rare token grouping method of AGG. Figure 4 presents the size of the rare token group during initial 1k training steps when the model is trained with WikiText-103 dataset. As presented in the figure, rare group size fluctuate wildly at the initial training stage. We expect for this grouping method to determine an optimal rare token group for the current training step. Table 10 presents the results of ablation study about dynamic grouping. To except dynamic grouping from AGG, we fixed the rare token group after 1 epoch. For this static grouping AGG method, Next-token diversity(Uniq) and the isotropy of the token embedding space($I(\mathbf{W})$) perform worse than dynamic grouping AGG.

Method	PPL↓	Uniq↑	I(W)↑
MLE	15.51	13143	0.377
AGG	15.51	13737	0.813
no $\mathbf{g}_1$	15.48	13018	0.367
no $\mathbf{g}_2$	15.51	13682	0.701

Table 9: Ablation study on gating vector of AGG.

Method	PPL↓	Uniq↑	I(W)↑
MLE	15.51	13143	0.377
AGG	15.51	13737	0.813
static AGG	15.55	13614	0.752

Table 10: Ablation study about dynamic grouping of AGG.

E Hyperparameter Sensitivity

In this sections we show how the metrics used on language modeling task change with the hyperparameter α in Figure 5. We observed an interesting phenomenon about the non-rare token group when rare token group size increases over a specific threshold. For the rare token group, Uniq and I(W) metrics have a positive correlation. They increase together up to a certain alpha value and decrease together as alpha increases over that value. However, for the non-rare token group, Uniq increases as alpha increases over that certain value while there are negative effects where I(W) decreases and Ppl increases. Because non-rare tokens are a major group, Figure 5 (b) and (c) present the above phenomenon about the non-rare token group although they present metrics for overall tokens. We consider this phenomenon to be another degeneration problem, as the increase of Uniq with negative impacts on isotropy and likelihood does not imply improvement of text quality, implying just generation of unproper tokens. This problem which occurs when rare token group size increases over a certain threshold can be handled in future work.

F Qualitative Study about Semantic Alignments between Tokens

In this section, we present qualitative studies about semantic alignments between tokens for language modeling and machine translation tasks. We select three rare token from each datasets: "homepage", "Werewolf", and "policymakers" for WikiText-103 dataset, and "optimum", "criminal", and "happiness" for WMT14 En→De dataset. For each rare token, we extract the top-5 nearest neighbor token predicted by the cosine distance between token embeddings. Compared with baseline MLE method, AGG shows significant improvement to train semantic alignments for rare tokens. From Table 12, we notice that the rare tokens trained with AGG are semantically well aligned and not biased about token frequency. Table 13 demonstrates that token embeddings trained with AGG also learn the cross-lingual semantic alignments between target language tokens.

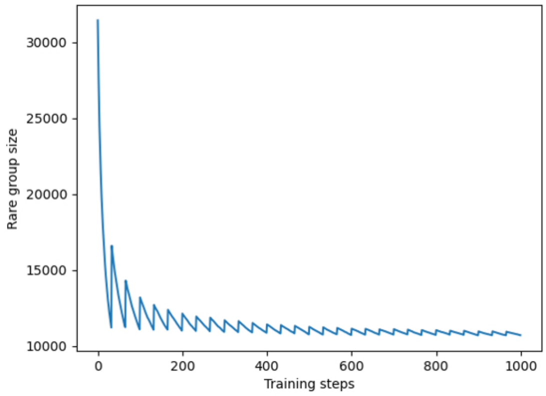


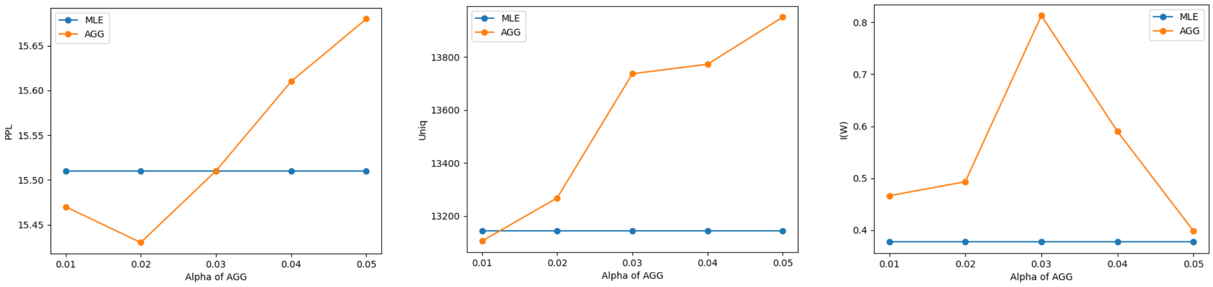
Figure 4: Size of the rare token group during initial 1k steps of training with WikiText-103 dataset.

G Examples

We present additional generated text samples from the model trained on language modeling task in Table 14. From the table, we notice that the model trained with AGG generates more diverse and high quality text than the baseline.

Hyperparameter	Empirical Study	Language Modeling	Machine Translation	
			Base	Big
# of layers	6	24	6-6	6-6
Hidden dimension	512	1024	512	1024
Projection dimension	2048	4096	2048	4096
# of heads	8	16	8	16
Dropout	0.1	0.1	0.1	0.3
Vocabulary size	44256	44256	40624	40624
# of parameters	42M	358M	65M	218M
Learning rate	$7 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
Max tokens per batch	32k	32k	64k	64k
Maximum training steps	40k	50k	190k	190k
Warmup steps	4k	4k	4k	4k
Optimizer	Adam	Adam	Adam	Adam
Weight decay	0.01	0.01	0.01	0.01
α for AGG	–	0.03	0.08	0.08
α for UL	–	1.0	–	–

Table 11: Model configurations and training hyper-parameters for all experiments conducted in the main page. For word similarity task, the model trained on language modeling task are evaluated for word similarity datasets.



(a) Perplexity

(b) Uniq

(c) $I(\mathbf{W})$

Figure 5: Hyper-parameter(α) sensitivity of AGG in the language modeling task on Wikitext-103 dataset.

homepage		Werewolf		policymakers	
MLE	AGG	MLE	AGG	MLE	AGG
BOX	website	ASUS	Creature	Steam	politicians
inbox	webpage	riet	Nightmare	death	environmentalists
livestream	blog	480	Bride	Venezuel	activists
namespace	Tumblr	nuclear	Sneak	includ	planners
hashes	websites	ATCH	Sniper	reason	economists

Table 12: Top-5 nearest neighbors of each rare tokens in WikiText-103 dataset. Performance of AGG method is compared with the baseline MLE method. Red color denotes the rare tokens among neighbors.

optimum		criminal		happiness	
MLE	AGG	MLE	AGG	MLE	AGG
therto	optimal	Criminal	criminals	juries	happy
ratory	optimale*	criminals	Criminal	enness	joy
consultan@@	optimalen*	perpetr@@	krimi@@*	ocopying	happ@@
sofar	maximum	secution	kriminellen*	ratory	Glück*
protection@@	Optim@@	xious	crime	sacri@@	pleasure

Table 13: Top-5 nearest neighbors of each rare source tokens in WMT14 En→De dataset. Performance of AGG method is compared with the baseline MLE method. The symbol @@ stands for sub-word tokenization of the dataset. The symbol * denotes the synonym token of the target language.

Method	Texts	Uniq ↑
Prefix	A Company , 2nd Engineer Combat Battalion , moved to the south side of the Yongsan - Naktong River road ; D Company of the 2nd Engineer Battalion was on the north side of the road . Approximately 2 miles (3	
MLE	. 2 km) north of the river , the 2nd Engineer Combat Battalion was on the south side of the road . The 2nd Engineer Combat Battalion was on the south side of the road .	22
AGG	. 2 km) north of the river , the 2nd Engineer Combat Battalion was attacked by a small force of North Korean People 's Volunteer Army (NK) troops . The North Koreans were able to penetrate the UN lines and capture the town .	43

Method	Texts	Uniq ↑
Prefix	The twelve 45 - calibre 35 . 6 cm (14 . 0 in) Type 41 guns of the Ise class were mounted in three pairs of twin - gun , superfiring turrets . Numbered one through	
MLE	two , the turrets were placed in superfiring pairs , one ahead of the other . The turrets were placed in superfiring pairs , with the turrets superfiring over the wing turrets . The turrets were placed in superfiring pairs , with the turrets superfiring over the wing turrets . The turrets were placed in superfiring pairs , with the turrets superfiring over the wing turrets . The turrets were placed in superfiring pairs , with the turrets superfiring over the wing turrets . The turrets were placed in superfiring	19
AGG	five from front to rear , the guns had a firing arc of 260 ° . They fired a 45 . 3 kg (99 lb) projectile at a muzzle velocity of 835 m / s (2 , 740 ft / s) to a range of 16 , 000 m (52 , 000 yd) . They had a maximum elevation of 30 degrees and fired a 36 . 3 kg (77 lb) projectile at a muzzle velocity of 835	55

Method	Texts	Uniq ↑
Prefix	Despite the expensive reconstructions , both vessels were considered obsolete by the eve of the Pacific War , and neither saw significant action in the early years of the war . Following the loss of most of the IJN 's large aircraft carriers during the Battle	
MLE	of Midway , the IJN was forced to rely on the United States Navy for aircraft carriers and aircraft carriers to support operations in the Pacific . The IJN 's aircraft carriers were the first to be equipped with the new Mark 4 torpedo and the IJN 's aircraft carriers were the first to be equipped with the Mark 4 torpedo . The IJN 's aircraft carriers were the first to be equipped with the Mark 4 torpedo , and the IJN 's aircraft carriers	37
AGG	of Midway , the IJN decided to modernize its fleet and modernize its fleet . The IJN 's new ships were designed to be capable of operating at speeds of up to 30 knots (56 km / h ; 35 mph) , and were fitted with a new bow section . The ships were designed to carry 1 , 000 t (980 long tons ; 1 , 100 short tons) of fuel oil , and were fitted with a pair of aircraft catap	63

Method	Texts	Uniq ↑
Prefix	Amos was born in Macclesfield , Cheshire and was a member of the Crewe Alexandra academy until he was released at the age of 10 . While at Crewe , he also played for another local team , Bollington United , as a centre	
MLE	- back . He was a member of the team that won the FA Youth Cup in 1989 , and was a member of the team that won the FA Youth Cup in 1990 . He was a member of the team that won the FA Youth Cup in 1990 , and was a member of the team that won the FA Youth Cup in 1992 .	24
AGG	- back . He was signed by Crewe Alexandra in July 2006 , and made his debut for the club in a 2 - 1 win over Rotherham United in the League Cup on 18 August 2006 . He was loaned out to Rotherham for the rest of the 2006 - 07 season , before being released at the end of the season .	52

Table 14: Generated texts on the Wikitext-103 test set and uniq tokens for each texts. 50 bpe tokens are given as prefix and the models are to generate the continuation of 100 next bpe tokens.