F[2]-Softmax: Diversifying Neural Text Generation via Frequency Factorized Softmax

Byung-Ju Choi[1]   **Jimin Hong[1]**   **David Keetae Park[2]**   **Sang Wan Lee[1,3,4]**

[1]Humelo, Republic of Korea
[2]Department of Biomedical Engineering, Columbia University, USA
[3]Department of Bio and Brain Engineering, KAIST, Republic of Korea
[4]Center for Neuroscience-inspired AI, KAIST, Republic of Korea
{bjej1123, jimin9401}@gmail.com
dkp2129@columbia.edu, sangwan@kaist.ac.kr

Abstract

Despite recent advances in neural text generation, encoding the rich diversity in human language remains elusive. We argue that the sub-optimal text generation is mainly attributable to the imbalanced token distribution, which particularly misdirects the learning model when trained with the maximum-likelihood objective. As a simple yet effective remedy, we propose two novel methods, *F[2]-Softmax* and *MefMax*, for a balanced training even with the skewed frequency distribution. *MefMax* assigns tokens uniquely to frequency classes, trying to group tokens with similar frequencies and equalize frequency mass between the classes. *F[2]-Softmax* then decomposes a probability distribution of the target token into a product of two conditional probabilities of (i) frequency class, and (ii) token from the target frequency class. Models learn more uniform probability distributions because they are confined to subsets of vocabularies. Significant performance gains on seven relevant metrics suggest the supremacy of our approach in improving not only the diversity but also the quality of generated texts.
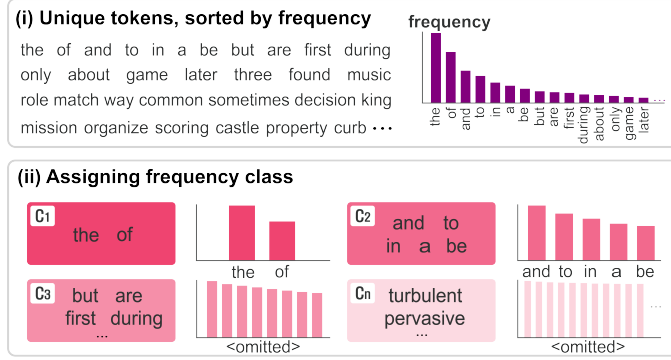
1 Introduction

Neural text generation is one of the extensively studied tasks of natural language processing (NLP), as it forms the basis for dialogue systems (Chen et al., 2017), machine translation (Chaudhary and Patel, 2018), and text summarization (Kryscinski et al., 2019). However, often monotonous or dull, texts generated from existing methods do not fully reflect the rich diversity and expression in human language (Welleck et al., 2020). In particular, models tend to overproduce words frequently appearing in the data, while hardly utilizing informative words (Dinan et al., 2020). Even pre-training techniques on large corpora fail to resolve the issue (Holtzman et al., 2019).

Possible causes for text degeneration have been illuminated, such as a defect specific to model architectures (Vig, 2018) or the discrepancy between training data and a true distribution (Holtzman et al., 2018; Jiang et al., 2019). Recently, the emphasis has been placed on investigating the flaws in the maximum likelihood objective (Holtzman et al., 2019). Concretely, the likelihood training pays little attention to the top ranks in terms of the target token probabilities (Welleck et al., 2020), or maximizing likelihood itself does not adequately reflect human language processing (Holtzman et al., 2019). Therefore, with the maximum likelihood-based training, models learn to produce tokens frequently appearing in the data more often.

We argue, however, that the primary reason behind the sub-optimal performance of the likelihood objective is essentially the imbalanced token distribution inherent in natural language. Natural language is extremely skewed in distribution, where the top hundred most frequently-used (top-100) words occupy nearly half of the total corpus (Fagan and Gençay, 2011) following the Zipf's law (Zipf, 1949). Training a classifier with the inherently imbalanced data on the maximum likelihood estimation (MLE) leads to biased classification boundaries in favor of majority classes (Khan et al., 2019). In other words, models play a difficult role in learning with the imbalanced label (i.e., token) distribution (He et al., 2008b).

We hypothesize that text generation can be enriched by balancing out the training data distribution. To this end, we introduce F[2]-Softmax (Fig. 1(B), Section 3.2), which factorizes the probability distribution of the target token into a product of two conditional probabilities of (i) frequency class, and (ii) token from the target frequency class. It ensures training over balanced data, since the frequency classes are designed to have the distribution close to uniformity, and token distributions

A MefMax: Frequency Class Assignment

(i) Unique tokens, sorted by frequency

the of and to in a be but are first during only about game later three found music role match way common sometimes decision king mission organize scoring castle property curb ···

frequency

(ii) Assigning frequency class

C_1 the   of

the   of

C_2 and   to   in   a   be

and to in a be

C_3 but   are   first   during   ...

<omitted>

C_n turbulent   pervasive   ...

<omitted>

B F^2-Softmax: Frequency Factorization

token pool: C_3 but   are   first   during   only   about   game   later

(i) Frequency class prediction

C_1 C_2 $*C_3$ C_4 C_5 C_6 ··· C_n

(ii) Token prediction

are   first   *during   only   about   game   later

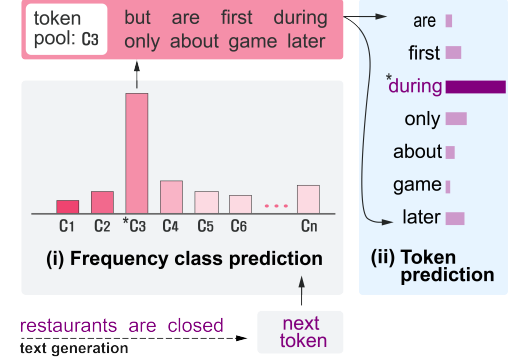restaurants are closed → next token

text generation

Figure 1: Illustrations for working mechanisms of the two proposed modules, MefMax and F^2-Softmax. (A) Once unique tokens are sorted by frequency, MefMax (Section 3.3) groups them to a set of frequency classes. The frequency distributions of the grouped tokens are more uniform than the distribution of the full vocabulary set. (B) With the frequency class assigned, token generation is decomposed into (i) predicting the frequency class, and (ii) generating the target token from the given frequency class.

within a class are confined to subsets of vocabularies grouped with similar frequencies. To this end, all unique tokens are assigned to a frequency class prior to the training, by our novel *mean efficiency maximization* (MefMax; Fig. 1(A), Section 3.3). MefMax evaluates and maximizes the class-labeling performance with the normalized entropy (i.e., efficiency), having the probability distributions to be learned as uniform as possible.

We conduct extensive performance evaluations on seven relevant metrics that quantify the diversity and quality of generated texts. In terms of the diversity of generated texts, our approach significantly outperforms not only the MLE baseline (Radford et al., 2019) but also other diversity-promoting alternatives (Welleck et al., 2020; Jiang et al., 2019). We also achieve state-of-the-art results on most of the quality performances.

2 Related Works

2.1 Diversity-promoting Text Generation

In the field of neural text generation, prior studies either take a training-based approach or a decoding-based approach to promote the diversity in the generated texts.

Training-based Approach.   In dialogue generation, stimulating models to generate texts that share high mutual information with the contexts (Li et al., 2016) has shown to improve the diversity of output tokens by adding a maximum mutual information (MMI) constraint to the standard likelihood objective. Meanwhile, FACE (Jiang et al., 2019) dynamically weights the cross-entropy losses based on target token frequencies, to prevent excessive weight-updates of some frequently used words. In another line of works for language modeling, text diversity has been promoted by a learning-to-cooperate framework in which multiple discriminators cooperate to reach a common goal (Holtzman et al., 2018). Also, the unlikelihood training strategy penalizes repetition with auxiliary loss terms (Welleck et al., 2020). Such works are orthogonal to ours since F^2-Softmax focuses on decomposing the softmax function without employing an auxiliary loss or loss re-scaling.

Decoding-based Approach.   One of the widely used decoding tactics for promoting the diversity and richness of texts is stochastic decoding. Top-k sampling stochastically samples the next token from the top-k candidates in the predicted probability distribution (Fan et al., 2018). Another pillar of stochastic decoding is nucleus sampling, which selects the next token from the top-p portion of the probability mass (Holtzman et al., 2019). Other studies include beam blocking (Paulus et al., 2017) in which the probabilities of tokens are set to zero if they were to create repeating n-grams, diverse beam search (Vijayakumar et al., 2018) which integrates dissimilarity terms into beam scores. Iterative beam search (Kulikov et al., 2019) enhances diverse beam search with multiple iterations of beam search with different search spaces. These techniques are agnostic about model architecture or training methods. Our approach can be harmonically combined with the above techniques.

2.2 Softmax Decomposition

Decomposing the softmax function has long been studied in language modeling. Goodman (2001) decomposed the softmax function using a two-level hierarchy. This idea was generalized to deeper hierarchies in a later study (Mnih and Hinton, 2009). Approaches to construct softmax hierarchies have followed, such as utilizing word clusters obtained from k-means algorithms (Le et al., 2011) or implementing Huffman coding with word frequencies (Mikolov et al., 2013). Furthermore, dynamic programming has been applied to obtain an optimal set of word classes with minimal computational costs for calculating the softmax function (Zweig and Makarychev, 2013). The same process has also been streamlined to fit into modern GPU environments (Grave et al., 2017). These techniques bear a resemblance to ours for the use of softmax decomposition. However, our goal is fundamentally different: we aim to balance the data distribution in training, whereas previous approaches share the primary goal of reducing computational costs.

2.3 Imbalanced Classification

That we assign tokens to classes of balanced distribution shares a similar goal with overcoming imbalanced classification in computer vision domains. One of the widely adopted techniques for imbalanced classification is re-sampling, which includes removing examples from the majority classes (under-sampling) and adding samples for the minority classes (over-sampling) (Buda et al., 2018). Techniques for over-sampling include interpolating samples from neighboring samples (Chawla et al., 2002) and adaptively synthesizing samples (He et al., 2008a). Cost-sensitive learning dynamically re-weights costs based on sample difficulties (Dong et al., 2017) or effective number of samples (Cui et al., 2018). Other studies for the data imbalance problem consider metric learning (Huang et al., 2016), knowledge transfer (Wang et al., 2017), and Bayesian estimation (Khan et al., 2019).

3 Methods

3.1 Maximum Likelihood

The goal of language modeling is to learn a model $\hat{p}(\mathbf{x})$ which best describes a joint probability distribution $p(\mathbf{x})$, where $\mathbf{x} = [x_1, \ldots, x_T]$ is a sequence of tokens and $x_i \in \mathcal{V}$ is a token from a vocabulary set. In an auto-regressive manner, $p(\mathbf{x})$ can

be factorized into a product of conditional probabilities of tokens; $p(\mathbf{x}) = \prod_t p(x_t|\mathbf{x}_{<t})$. A conventional approach for the training is to maximize log-likelihood of a sequence $\mathbf{x}$ as the following:

$$\mathcal{L}_{\text{MLE}}(\hat{p}, \mathbf{x}) = \sum_{t=1}^{T} \log \hat{p}(x_t|\mathbf{x}_{<t}). \quad (1)$$

3.2 F²-Softmax

We propose to factorize the posterior $\hat{p}(x_t|\mathbf{x}_{<t})$ into a product of two conditional probabilities:

$$\hat{p}(x_t|\mathbf{x}_{<t}) = \hat{p}_1(c_t|\mathbf{x}_{<t}) \times \hat{p}_2(x_t|c_t, \mathbf{x}_{<t}), \quad (2)$$

where $c_t \in \mathcal{C}$ denotes a frequency class label assigned to the token x_t given the global frequency of the token in a corpus, belonging to a set of frequency classes $\mathcal{C}$. Following Eq. (2), the updated objective $\mathcal{L}_{F^2}(\hat{p})$ is then formulated as:

$$\begin{aligned} &\mathcal{L}_{F^2}(\hat{p}, \mathbf{x}) \\ &= \sum_{t=1}^{T} [\log \hat{p}_1(c_t|\mathbf{x}_{<t}) + \log \hat{p}_2(x_t|c_t, \mathbf{x}_{<t})]. \end{aligned} \quad (3)$$

The objective is thus learning how to classify the target frequency of the token and selecting the exact token given the target frequency class. The factorized probabilities $\hat{p}_1(c_t|\mathbf{x}_{<t})$ and $\hat{p}_2(x_t|c_t, \mathbf{x}_{<t})$ are defined empirically using softmax functions:

$$\begin{aligned} \hat{p}_1(c_t|\mathbf{x}_{<t}) &= \frac{\exp(\mathbf{h}_{t-1} \cdot \mathbf{u}^{c_t})}{\sum_{m \in \mathcal{C}} \exp(\mathbf{h}_{t-1} \cdot \mathbf{u}^m)} \\ \hat{p}_2(x_t|c_t, \mathbf{x}_{<t}) &= \frac{\exp(\mathbf{h}_{t-1} \cdot \mathbf{o}^{x_t})}{\sum_{n \in \mathcal{V}_{c_t}} \exp(\mathbf{h}_{t-1} \cdot \mathbf{o}^n)}, \end{aligned} \quad (4)$$

where $\mathbf{h}_{t-1}$ is a hidden state of the context $\mathbf{x}_{<t}$; $\mathbf{o}^i$ and $\mathbf{u}^j$ can be viewed as output embedding vectors for $i \in \mathcal{V}_{c_t}$ and $j \in \mathcal{C}$, respectively, while $\mathcal{V}_{c_t}$ is a subset of vocabularies assigned to the class c_t. Note that $\hat{p}_2(x_t|c_t, \mathbf{x}_{<t})$ is computed from the narrowed pool of tokens $\mathcal{V}_{c_t}$ rather than the full vocabularies set $\mathcal{V}$. Since classes are differentiated based on the token frequency, tokens with the same class have similar frequencies. It ensures within-class frequency distribution of tokens is closer to uniform than that of the full vocabulary set.

3.3 MefMax for Class Optimization

The more uniform a label distribution is, the less likely decision boundaries are biased in favor of frequent classes. Therefore, we aim to maximize

the degree of uniformity of frequency distributions for both (i) tokens within each class and (ii) classes themselves (i.e., the sum of token frequencies within each class), to avoid the class imbalance problem (Buda et al., 2018) over the course of training. It is formalized as follows:

$$\mathcal{C}' = \arg\max_{\mathcal{C}}[\mathcal{U}(\mathcal{C}) + \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}}\mathcal{U}(\mathcal{V}_i)], \quad (5)$$

where $\mathcal{U}$ is a function that measures the uniformity of the frequency distribution of a given set. While any tests of uniformity can be used as $\mathcal{U}$, we adopt *Shannon's entropy* (Shannon, 1948). The entropy is a decent proxy for measuring the uniformity (Dudewicz and Van Der Meulen, 1981).

Normalized Entropy. Since the number of samples affects the entropy, entropy cannot be directly used. To marginalize the effect of the sample size, we use *efficiency*, which is also known as the normalized entropy (Wijesekera and Dillon, 1997), defined as:

$$\mathcal{U}(\mathbf{k}) = -\sum_{k_i\in\mathbf{k}}\frac{p(k_i)\log(p(k_i))}{\log(|\mathbf{k}|)}. \quad (6)$$

It is equivalent to the ratio of the entropy to the maximum entropy, if the data were perfectly uniform. By applying the efficiency to Eq. (5), our objective is to find a set of classes and their vocabularies such that their *mean efficiency* is maximized.

Greedy Approach. The remaining issue is the computational overhead since the cost for exploring all possible class boundaries grows exponentially with the vocabulary size, not to mention the challenge of finding the optimal number of classes. To improve computational efficiency, we employ a straightforward greedy mechanism. It is based on the assumption that the mean efficiency is maximized when each class has approximately the same total frequency size. This assumption allows us to reduce our objective to optimizing the number of classes. Given a sorted vocabulary set $\mathcal{V}'$ and a candidate number of classes K, we divide classes so that each class has the same $1/K$ of total frequency. The optimal number of classes is the one that maximizes the mean efficiency. Algorithm 1 shows the complete pseudo-code. Computation time is linear to the vocabulary size.

Algorithm 1 Pseudo-code for MefMax

> **Inputs :** Array $\mathcal{V}'$ of length n sorted by the decreasing order of token frequency
> **Outputs :** Number of classes, class boundary tokens

1: $\mathcal{V}' \leftarrow \mathcal{V}' / \text{sum}(\mathcal{V}')$ ▷ get relative frequencies
2: maxMeanEfficiency $\leftarrow$ 0
3: maxClassNum $\leftarrow 1/\mathcal{V}'[0]$
4: **for** K in [1, 2, ..., maxClassNum] **do**
5:      $B \leftarrow$ empty list   ▷ lists for candidate boundaries
6:      tar $\leftarrow 1/K$ ▷ target frequency
7:      cum, idx $\leftarrow$ 0, 0   ▷ cumulative frequency & index
8:      **while** tar $\leq$ 1 **do**   ▷ compute candidate boundaries
9:          cum $\leftarrow$ cum + $\mathcal{V}'$[idx++]
10:          **if** cum $\geq$ tar **then**
11:             tar $\leftarrow$ tar + $1/K$
12:             B.append(idx)
13:      meanEfficency $\leftarrow$ mean efficiency based on B
14:      **if** maxMeanEfficiency $<$ meanEfficency **then**
15:          maxMeanEfficiency $\leftarrow$ meanEfficency
16:          $Out \leftarrow B$
17: **return** len(Out), Out

3.4 Decoupled Decoding

For the decoding stage, we decouple $\hat{p}_1$ from $\hat{p}$ in Eq. (2) by first selecting a single frequency class from $\hat{p}_1$ and then generating the next token based on the selected class. For the target class $c_t = i$ sampled from the distribution $\hat{p}_1(c_t|\mathbf{x}_{<t})$, the probability for the next token is defined as:

$$\hat{p}'(x_t|\mathbf{x}_{<t}) = \begin{cases} \hat{p}_2(x_t|c_t = i, \mathbf{x}_{<t}) & \text{if } x_t \in \mathcal{V}_i \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The target class can be sampled in both deterministic or stochastic manners, depending on decoding strategies. We found that the advantages of training balanced data distributions can be fully leveraged by sequentially performing tasks of frequency class prediction and token generation from the selected class.

4 Experiments

4.1 Training and Evaluation Details

In this section, experimental details are illustrated. Exact hyperparameter settings and data statistics are described in Appendix.

4.1.1 Datasets

Two datasets that differ in language and text types are selected for the implementations.
Wikitext-103 [1] is a collection of English articles extracted from Wikipedia. Containing more than

---

[1] https://s3.amazonaws.com/research.metamind.io/wikitext/wikitext-103-v1.zip

100 million words, it is widely regarded as a benchmark dataset for language modeling.

Melo-Lyrics is a Korean lyrics dataset we crawled from multiple music streaming websites, including Soribada[2], Genius[3], etc. Tokens in lyrics show a distribution largely different from general articles; for instance, repeated phrases are abundant in lyrics. Therefore it provides an additional unique angle for model evaluations and comparisons. It includes approximately 478 thousand songs with 51 million words in total.

4.1.2 Model Architecture

We use the Transformer (Vaswani et al., 2017), an architecture well-suited for neural text generation (Lewis et al., 2019; Welleck et al., 2020). Specifically, we apply the Transformer decoder used in the GPT-2 model (Radford et al., 2019). Input texts are tokenized with the byte pair encoding (Sennrich et al., 2016).

4.1.3 Baseline Models

For the baseline, we consider maximum likelihood estimation (MLE), a standard approach for text generation. Also compared are alternative models for promoting text diversities, including recently proposed FACE[4] (Jiang et al., 2019) and unlikelihood training[5] (UL) (Welleck et al., 2020). FACE improves text diversity by dynamically scaling losses, while the latter employs auxiliary losses.

4.1.4 Training

Training is carried out on a single GPU environment with 24GB of memory. We set all hyperparameters equal for all approaches by tuning them based on the validation losses of the MLE baseline for fair comparisons. We additionally optimize approach-specific hyperparameters of diversity-promoting baselines.

4.1.5 Generation

We generate texts for the evaluation by completing sequences from prefixes. Specifically, we batchify a test set, select the first 50 tokens from each batch as prefixes, and guide models to generate a continuation of 100 tokens from the prefixes. The experiments include both deterministic and stochastic

---

[2] https://www.soribada.com
[3] https://genius.com
[4] https://github.com/ShaojieJiang/FACE
[5] https://github.com/facebookresearch/unlikelihood_training

decoding. We apply greedy search for deterministic decoding, and use top-k sampling for stochastic decoding.

4.1.6 Evaluation Metrics

From seven total quantitative metrics we adopt to evaluate our model, Perplexity (Bengio et al., 2003), KL-Divergence (Kullback, 1997), and MS-Jaccard (Alihosseini et al., 2019) are closely related to the *likelihood* of generated texts. The other four metrics, namely Self-BLEU (Zhu et al., 2018), Distinct-n (Li et al., 2016), Repetition (Holtzman et al., 2019), and Uniq (Welleck et al., 2020) measure the text *diversity*.

Perplexity quantifies the prediction difficulty over the next token. It is regarded as a general performance metric for text generation.

KL-Divergence measures the difference between two probability distributions. We use unigram distributions of the generated texts and the test data.

MS-Jaccard computes the similarity between the model's output and the ground truths by matching n-grams.

Self-BLEU evaluates the inter-text diversity by computing BLEU (Papineni et al., 2002) score for each generated text by regarding other outputs as reference.

Distinct-n quantifies the intra-text diversity based on distinct n-grams in each text.

Repetition examines whether texts are stuck in repetitive loops.

Uniq quantifies the richness of models using the number of unique generated tokens.

4.2 Quantitative Comparisons

In this section, we report the scores computed from fully-trained models on the two benchmarks, Wikitext-103 and Melo-Lyrics, compared against baselines. Table 1 shows the results of stochastic decoding, while the results of deterministic decoding are reported in Table 2.

4.2.1 Stochastic Decoding

Wikitext-103. The desired qualities we aim for a text generation model is to generate human-like texts with a wide spectrum of token choices. Coupled with top-k sampling, our F^2-Softmax achieves both goals by outperforming baselines with nearly all metrics compared, and closely approaching the human gold standard. As shown in Table 1(a), our model is particularly effective in capturing the token diversity in the corpus. Notably, F^2-Softmax

Metrics / Models	PPL	KLD	MS-Jaccard			Self-BLEU			Distinct			Rep	Uniq
			$n=1$	$n=2$	$n=3$	$n=1$	$n=2$	$n=3$	$n=1$	$n=2$	$n=3$		
MLE	**24.7**	1.51	52.1	35.6	24.3	93.4	83.2	69.7	45.1	71.9	83.0	0.67	8.48k
FACE	29.7	1.26	53.3	33.2	21.7	92.3	76.5	57.6	53.4	77.1	85.5	2.1	10.3k
UL-token	25.8	1.28	54.3	36.7	24.8	93.7	82.4	68.1	50.0	77.3	87.1	0.39	10.2k
UL-token+seq	27.5	1.33	50.6	35.4	23.5	**95.3**	83.4	66.9	57.6	86.6	94.2	**0.11**	10.6k
F^2-Softmax	25.6	**0.62**	**67.4**	**42.4**	**26.4**	93.3	**71.9**	**48.1**	65.7	89.7	94.4	0.33	**15.7k**
Human	-	-	-	-	-	95.2	74.1	50.9	69.1	92.1	95.8	0.11	15.2k

(a) Wikitext-103

Metrics / Models	PPL	KLD	MS-Jaccard			Self-BLEU			Distinct			Rep	Uniq
			$n=1$	$n=2$	$n=3$	$n=1$	$n=2$	$n=3$	$n=1$	$n=2$	$n=3$		
MLE	**13.1**	0.34	67.6	47.3	32.4	**97.5**	83.1	62.2	56.3	75.5	84.0	1.1	22.4k
FACE	13.9	0.39	60.3	41.5	28.3	**97.5**	82.0	60.2	58.6	**78.1**	86.5	0.9	22.7k
UL-token	13.8	0.39	62.4	43.6	29.6	98.2	84.0	64.2	61.0	78.3	84.7	0.3	22.8k
UL-token+seq	16.6	0.45	58.3	39.4	25.7	98.7	79.1	57.1	67.2	90.7	95.0	0.07	22.3k
F^2-Softmax	13.2	**0.13**	**78.8**	**52.4**	**34.4**	**97.5**	76.1	50.3	64.0	78.7	**83.9**	**2.5**	**25.2k**
Human	-	-	-	-	-	97.5	76.6	53.2	62.5	77.5	82.5	2.1	27.9k

(b) Melo-Lyrics

Table 1: Quantitative comparisons of F^2-Softmax with baseline models. Top-k sampling strategy with the k size of 3 and 20 are used for Wikitext-103 and Melo-Lyrics, respectively, across all models. UL-token imposes a token-level penalty, and UL-token+seq considers both token- and sequence-level penalties. PPL, KLD, and Rep are abbreviated notations for perplexity, KL-Divergence, and Repetition, respectively. Numbers $n \in (1, 2, 3)$ in column headers under MS-Jaccard, Self-BLEU, and Distinct refer to n-grams. Boldface scores denote the performances closest to humans. F^2-Softmax outperforms compared baselines in most of the cases. Results on various k sizes are reported in Appendix.

significantly improves both Self-BLEU and Distinct performances, having relative gaps to the human gold standard of 3.4% and 3%, respectively. The performance gaps of the second-best scores are 6.5% (FACE) and 8.1% (UL-token+seq), respectively. A surprising result is that F^2-Softmax improves Rep performance by 50% over MLE, without an explicit penalty on repeating tokens. Another seminal contribution is the 30% relative increase in unique tokens used for the generation, from the previously state-of-the-art level of 10.6k to 15.7k, as shown by the Uniq metric. This level closely reflects the human use of 15.2k tokens.

In PPL, which reflects the likelihood of the generated texts, the diversity-promoting baselines tend to perform worse than MLE, presumably due to the trade-offs between the diversity and the likelihood of texts. In contrast, F^2-Softmax maintains the smallest performance drop on PPL. F^2-Softmax also improves KLD and MS-Jaccard by 59% and 19% over MLE, respectively, which are large margins compared to the other comparatives.

Melo-Lyrics. Significant performance gains of F^2-Softmax are also observed in lyrics generation in Table 1(b). The diversity-promoting baselines display severer degradation in PPL, KLD, and MS-Jaccard compared to the Wikitext-103 dataset. Especially, their repetition levels are significantly different from that of the ground truth data. We attribute this observation to the distinctive characteristics of lyrics, in which the same phrases are rhythmically repeated throughout the songs in the form of chorus or hook. Thus, for lyrics dataset, forcing models to discourage reusing previously used tokens may adversely affect the likelihood of the generated texts. Since F^2-Softmax helps models to diversify the output without an explicit regularization, models learn to generate well-thought-out tokens from the diversified token pool of 25.2k (Uniq), with state-of-the-art performances in KLD, MS-Jaccard, Self-BLEU, Distinct, and Rep.

4.2.2 Deterministic Decoding

In deterministic decoding, there is no clear method that outperforms the others in all of the metrics. For example, UL-token+seq exhibits the best performance in Distinct and Rep, while presenting the worst score in MS-Jaccard. Similarly, FACE improves Self-BLEU in exchange for performance loss on PPL and MS-Jaccard. Since we have seven metrics to compare, we conduct pair-wise evalu-

Metrics / Models	PPL	KLD	MS-Jaccard			Self-BLEU			Distinct			Rep	Uniq
			n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3		
MLE	**24.7**	2.17	45.6	29.9	20.5	92.8	83.6	73.2	31.4	48.8	57.8	18.3	6.23k
FACE	29.7	1.67	47.9	29.9	19.8	89.6	**73.6**	**57.3**	38.4	54.3	62.0	21.0	8.03k
UL-token	25.8	1.88	47.2	30.7	20.9	92.9	83.7	73.3	37.1	56.3	65.6	12.8	7.66k
UL-token+seq	27.5	2.06	41.5	26.9	18.4	**95.6**	86.6	74.2	**49.9**	**78.1**	**89.8**	**0.3**	8.33k
F^2-Softmax	25.6	**1.63**	**49.0**	**31.2**	**21.0**	90.2	78.7	66.3	36.3	54.6	63.8	12.5	**9.08k**
Human	-	-	-	-	-	95.2	74.1	50.9	69.1	92.1	95.8	0.11	15.2k

(a) Wikitext-103

Metrics / Models	PPL	KLD	MS-Jaccard			Self-BLEU			Distinct			Rep	Uniq
			n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3		
MLE	**13.1**	0.46	64.5	44.2	31.6	91.4	71.8	51.5	22.0	26.7	28.5	67.7	20.4k
FACE	13.9	0.51	57.7	39.6	22.8	92.3	72.5	55.6	25.3	32.0	36.1	39.1	**21.7k**
UL-token	13.8	0.51	62.8	42.7	30.7	92.3	**73.0**	53.5	21.8	26.4	28.5	66.6	20k
UL-token+seq	16.6	0.74	50.6	34.1	23.9	**95.7**	78.7	56.3	**29.6**	**52.2**	**57.1**	**25.1**	19.9k
F^2-Softmax	13.2	**0.38**	**67.4**	**45.1**	**31.7**	90.4	66.6	43.1	21.5	26.0	28.0	66.9	21.3k
Human	-	-	-	-	-	97.5	76.6	53.2	62.5	77.5	82.5	2.1	27.9k

(b) Melo-Lyrics

Table 2: Evaluation results on the greedy sampling. The abbreviations are the same as Table 1.

ations between the compared methods, in which a method outperforms the other when a majority of metrics record higher. Our approach beats compared methods seven out of eight times (Table 9). This result supports the supremacy of our approach regardless of the choice of decoding strategies.

However, deterministic decoding does not see the same amount of benefits obtained from stochastic decoding. We empirically find from our analyses that argmax operation in deterministic settings may harm the diversity when target class probabilities are nearly evenly distributed. We plan to delve deeper into our approach to improve our approach further.

4.3 Learning Balanced Distribution

The characteristic markers of monotonous texts are an overproduction of frequently used tokens and under-representation of rare tokens. To compare how models differentially generate tokens from frequent and rare tokens, we count the number of generated tokens corresponding to four defined categories of *frequent*, *medium*, *rare* and *very rare*. Tokens in each category are predefined from the Wikitext-103 training set. Fig. 2 plots the distribution results. MLE produces *frequent* tokens 34% more than human, while under-producing *rare* and *very rare* tokens by 40%. The unlikelihood training baselines (UL-token, UL-token+seq) improve the diversity against MLE, but their results are relatively far from the real distribution. FACE man-
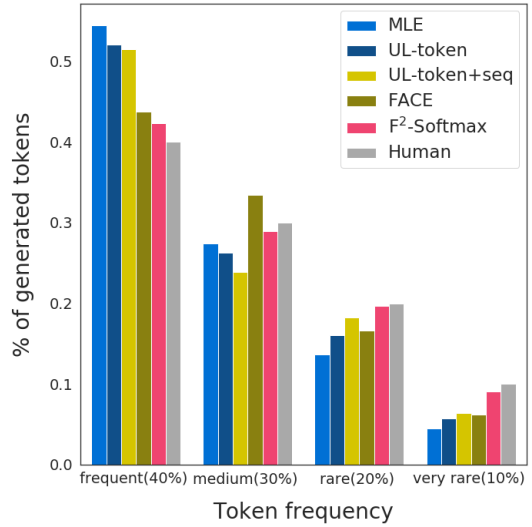

Figure 2: Frequency distribution comparisons on the Wikitext-103 test set. Tokens in each group are defined based on the frequency mass of the training set. Tokens occupying the top 40% of the frequency mass are assigned to *frequent*, while those corresponding to the bottom 10% are classified to *very rare*.

ages to regulate a disproportionate use of *frequent* tokens, but it fails to generate adequate amount of *very rare* tokens. Generation results of our F^2-Softmax are closest to the gold standard, with the differences in *frequent* and *rare* tokens falling within 6%.
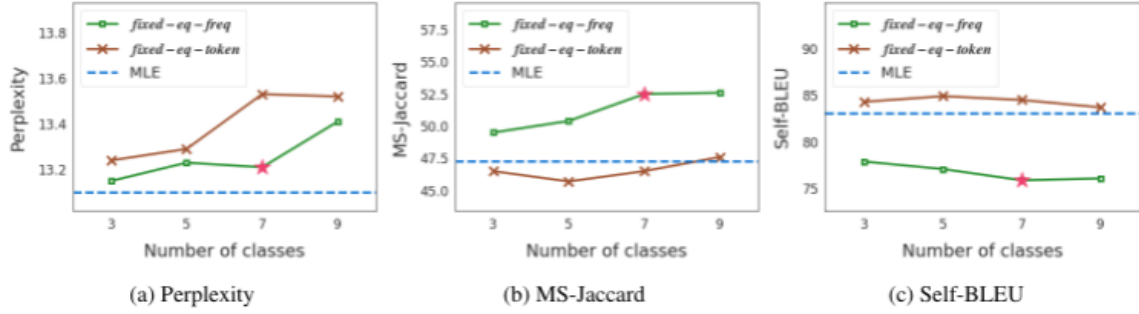
Figure 3: Bigram performance on the Melo-lyrics test set with different numbers of classes. Green and brown lines indicate models with tokens are distributed based on frequency mass and token size, respectively. Blue dotted lines and star marks represent the MLE baseline and the choice of MefMax, respectively.

4.4 Ablation Studies

In this section, we justify the pivotal roles of Mef-Max (Section 3.3) and the decoupled decoding strategy (Section 3.4). In order to assess contributions toward the final performances, we conduct a series of ablation tests. Stochastic decoding is used for the ablation studies.

4.4.1 Ablation on MefMax

MefMax finds a desirable number of classes, intending to balance the frequency distribution of tokens between classes. Does MefMax help achieve better generation results than possible variants of class assignment? We answer this question by comparing the final performances against two simpler variants of MefMax. We name the first variant as *fixed-eq-token* in which tokens are distributed in *equal* numbers to a *fixed* number of classes. The second variant, *fixed-eq-freq*, also assumes a fixed number of classes, but tokens are assigned to minimize the difference in the frequency distribution between classes.

Fig. 3 presents the results. Clearly, *fixed-eq-freq* outperforms *fixed-eq-token*. It indicates that a decomposition of the softmax function without consideration of the data distribution (i.e., frequency distribution) aggravates both the likelihood and token diversity performances, regardless of the number of classes. For *fixed-eq-token*, we find that models tend to overclassify the target class to the first class, which contains most of the total frequency, having most tokens generated from a fraction of the total vocabulary. <mark>This finding also justifies the hypothesis that balanced data distribution is an important factor in text generation.</mark>

Assigning classes based on the frequency (i.e., *fixed-eq-freq*) continues to improve MS-Jaccard

and Self-BLEU until the number of classes reaches the class choice of MefMax. With a larger number of classes than the choice, performances either plateau or decrease, demonstrating that MefMax is capable of selecting the optimal class size. Interestingly, perplexity significantly deteriorates when the number of classes exceeds the optimal number decided by MefMax.

4.4.2 Ablation on Decoupled Decoding

Decoupled decoding formalized in Eq. (7) fully leverages the benefits of F^2-Softmax by sequentially performing the frequency class prediction and token generation tasks. Table 3 reports the results from an ablation test on the decoupled decoding. To ablate the decoupled decoding, we use a full posterior in Eq. (2). We observe significant performance gaps, meaning that the decoupled decoding is an indispensable component of our model. Notably, even without the decoupled decoding, our model maintains better performances than the MLE baseline.

4.5 Qualitative Comparisons

To further examine the generation quality, we sample texts from the trained models. Table 4 compares the generated texts from the same prefix. The results suggest that all trained models are capable of

Model	Decoupled Decoding	KLD	MSJ	SB	Uniq
F^2-Softmax	O	0.13	52.4	76.1	25.2k
	×	0.31	47.4	81.9	22.6k
MLE	×	0.34	47.3	83.1	22.4k

Table 3: Ablation study on the decoupled decoding. MSJ and SB stand for MS-Jaccard and Self-BLEU, respectively. Scores are measured on bigram level.

Model	Texts	VR (%)
Prefix	= Robert Boulter = Robert Boulter is an English film, television and theatre actor. He had a guest-starring role on the television series The Bill in 2000. This was followed by a starring role in the play Herons written	
MLE	for the television series The Bill in 2001. **Boul**ter was cast in the role of Robert **Boul**ter on the television series The Bill in 2002. The character was first cast in the role of Robert **Boul**ter in the 2003 film The Bill in a series of short stories about the life of a man. = = Personal life = = **Boul**ter is a **vegetarian**	6.7
FACE	by John **Landis** in 2001 . His role in the play was also played by John Lloyd , who played a fictional character in The Bill . **Boul**ter played a leading role in The Bill in 2002 and played a leading role in The Bill in 2004 . In 2006 he starred as a supporting actor in The Bill , a film adaptation of John Wayne 's play The **Nut**ty Professor in which	4
UL-token	in 2002 by the BBC. **Boul**ter was cast by the BBC, playing a young Englishmen in a **pub** in the fictional town of Bristol. He appeared in the 2006 BBC drama series, The Bill, in which he was cast as the lead character. He played the role of Peter **Gabriel**, an Englishman who is portrayed in a television serial killer. **Boul**ter is	5.3
UL-token+seq	and directed by David Lloyd. **Wins**let was a director for **Wins**let 's comedy-drama, The New **Yorker**. He starred in several of his films, and was also the first actor to appear in multiple films. He played the role in the film 's second part in the 2002 **HBO** series The **Adventures** of Bob **Marley** (which premiered in 2001) and was written by David	8
F^2-Softmax	by Philip **Glass** in 2003. He co-starred as the male fictionalised father with a woman named Anne **Hutchinson**, an **eccentric** lawyer who discovers that his family had **vanished** in the previous film. **Boul**ter's performance in A Day in Yellow **Cell** is considered a turning point in his career. He received an Academy Award nomination for Best Actor in 2004. **Boul**ter appeared in several films	**9.3**

Table 4: Generated texts on the Wikitext-103 test set. A prefix from the first batch was selected to avoid cherry-picking. **VR** denotes the ratio of *very rare* tokens (see Section 4.3 for the definition) against the text length. While all colored and bold-faced tokens indicate *very rare* tokens, **green** color denotes repeated tokens, and **red** color is reserved for non-pronoun words.

generating texts semantically coherent to the prefix. However, they differ in rare token usage patterns. While our F^2-Softmax exhibits the highest usage of rare tokens, we observe two issues from the baselines. The first is that models tend to repeat the same rare token across all sentences after its first appearance (MLE). The other issue is that generated rare tokens are mostly pronouns (UL-token-seq). Unlike the baselines, F^2-Softmax utilizes the broadest range of rare tokens with significantly less, but more likely, repetitions. Further, F^2-Softmax is shown to be adept at utilizing non-pronoun rare tokens, such as 'eccentric' or 'vanished'.

5 Conclusion

In this paper, we proposed F^2-Softmax, a simple but effective method for better learning the rich diversity in text. F^2-Softmax encourages models to diversify text generation by readjusting class formation and motivating models to learn a more balanced token distribution. Quantitative and qualitative analyses validate the diversity-promoting performances of our approach. Since it can be quickly adopted to replace the traditional likelihood objective, we believe in broader applicability of F^2-Softmax. Thus, future work involves extending the method to other related tasks, such as machine

translation and text summarization, and investigating the potential gains from transfer learning.

References

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proc. of Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Janhavi R Chaudhary and Ankit C Patel. 2018. Machine translation using deep learning: a survey. *In-*

ternational Journal of Scientific Research in Science, Engineering and Technology, 4(2):145–150.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. Acm Sigkdd Explorations Newsletter, 19(2):25–35.

Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. 2018. Large scale fine-grained categorization and domain-specific transfer learning. In Proc. of IEEE conference on computer vision and pattern recognition (CVPR), pages 4109–4118.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In The NeurIPS'18 Competition, pages 187–208. Springer.

Qi Dong, Shaogang Gong, and Xiatian Zhu. 2017. Class rectification hard mining for imbalanced deep learning. In Proc. of IEEE International Conference on Computer Vision (CVPR), pages 1851–1860.

Edward J Dudewicz and Edward C Van Der Meulen. 1981. Entropy-based tests of uniformity. Journal of the American Statistical Association, 76(376):967–974.

Stephen Fagan and Ramazan Gençay. 2011. An introduction to textual econometrics. Handbook of empirical economics and finance, pages 133–154.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In Proc. of Annual Meeting of the Association for Computational Linguistics (ACL), pages 889–898.

Joshua Goodman. 2001. Classes for fast maximum entropy training. In Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 1, pages 561–564. IEEE.

Edouard Grave, Armand Joulin, Moustapha Cissé, Hervé Jégou, et al. 2017. Efficient softmax approximation for gpus. In Proc. of International Conference on Machine Learning (ICML), pages 1302–1310. JMLR.

Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008a. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In Proc. of IEEE international joint conference on neural networks (IJCNN), pages 1322–1328. IEEE.

Zhongjun He, Qun Liu, and Shouxun Lin. 2008b. Improving statistical machine translation using lexicalized rule selection. In Proc. of International Conference on Computational Linguistics (COLING), pages 321–328.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In Proc. of Annual Meeting of the Association for Computational Linguistics (ACL), pages 1638–1649.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In International Conference on Learning Representations (ICLR).

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In Proc. of IEEE conference on computer vision and pattern recognition (CVPR), pages 5375–5384.

Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. In Proc. of World Wide Web Conference (WWW), pages 2879–2885.

Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. 2019. Striking the right balance with uncertainty. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 103–112.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In Proc. of Empirical Methods in Natural Language Processing (EMNLP), pages 540–551.

Ilia Kulikov, Alexander Miller, Kyunghyun Cho, and Jason Weston. 2019. Importance of search and evaluation strategies in neural dialogue modeling. In Proc. of International Conference on Natural Language Generation (INLG), pages 76–87.

Solomon Kullback. 1997. Information theory and statistics. Courier Corporation.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011. Structured output layer neural network language model. In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5524–5527. IEEE.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In Proc. of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pages 110–119.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Proc. of Advances in neural information processing systems (NIPS)*, pages 1081–1088.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of Annual Meeting on Association for Computational Linguistics (ACL)*, pages 311–318. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations (ICLR)*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Claude E Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of Advances in neural information processing systems (NIPS)*, pages 5998–6008.

Jesse Vig. 2018. Deconstructing bert: Distilling 6 patterns from 100 million parameters. *Medium.*

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proc. of AAAI Conference on Artificial Intelligence (AAAI)*.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 7029–7039.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations (ICLR)*.

Hemantha W Wijesekera and Thomas M Dillon. 1997. Shannon entropy as an indicator of age for turbulent overturns in the oceanic thermocline. *Journal of Geophysical Research: Oceans*, 102(C2):3279–3291.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *Proc. of ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 1097–1100.

George Kingsley Zipf. 1949. Human behavior and the principle of least effort.

Geoffrey Zweig and Konstantin Makarychev. 2013. Speed regularization and optimality in word classing. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8237–8241. IEEE.

A Datasets

A.1 Melo-Lyrics Data Collection

Few datasets have been publicly available for Korean text generation, and none of them has gained public consensus as a benchmark dataset, partly due to their small sample sizes. We collect lyrics data for three rationales. First, we test our model on a language other than English. Second, a large number of songs and lyrics are available. Lastly, lyrics show distributional characteristics at odds with Wikipedia. The crawling session was held between 5th July 2019 to 29th July 2019. After crawling enough data, we discarded those containing more than ten foreign language words, except for English. English was an exception since using English in Korean lyrics is natural and common. We also manually refined the lyrics by deleting noises, including advertisements and unnecessary meta-information about the lyrics writer transcriber. The remaining data consist of roughly 478 thousand lyrics with 51 million words. Indexed data can be downloaded from the url below[6]. We plan to release the raw data for research purposes only.

A.2 Data Statistics

The number of articles (songs) and containing words for training, test and validation phases are reported in Table 5.

	train	test	valid
# of articles	28,475	60	60
# of words	113,655,420	269,551	236,966

(a) Wikitext-103 dataset

	train	test	valid
# of songs	430,837	23,935	23,935
# of words	46,343,239	2,566,598	2,501,304

(b) Melo-Lyrics dataset

Table 5: Statistics on the datasets

B Hyperparameter Configurations

The detailed hyperparameters used are illustrated.

B.1 Model Hyperparamters

Table 6 reports the detailed list of model hyperparameters. The dropout and drop attention ratios are chosen from a set {0, 0.1} based on validation

---

[6] https://drive.google.com/drive/folders/1bXborfoUiaHYU0X_1t-TmnBkClU4ts9O

Hyperparameter \ Dataset	Wikitext-103	Melo-Lyrics
# of layers	12	
Hidden dimension	512	
Projection dimension	2048	
# of heads	8	
Head dimmension	64	
Dropout	0.1	
Drop attention	0	
Sequence length	1024	512
Vocabulary size	30,000	40,000
Total # of parameters	69.0M	76.5M

Table 6: Model hyperparameter settings

losses. Sequence length is selected from a set {512, 1024}. We assigned 10,000 more vocabularies to training models on the Melo-Lyrics dataset, illuminating the characteristics of Korean language where words with varying forms may have similar meanings.

B.2 Training Hyperparamters

Hyperparameter \ Dataset	Wikitext-103	Melo-Lyrics
Batch size	8	16
Learning rate	0.0001	0.0002
Finetuning LR	0.00001	0.00002
Finetuning step	1500	1500
Gradient clipping	0.25	0.25
Weight decay	0.001	0
Optimizer	Adam	Adam
- β_1	0.9	0.9
- β_2	0.999	0.999
- ϵ	1e-8	1e-8

Table 7: Training hyperparameter settings

Table 7 shows the training configurations. The learning rate, gradient clipping norm, weight decay are selected from a set, {0.00005, 0.0001, 0.00015, 0.0002, 0.00025}, {0.25, 5.0}, {0, 0.001, 0.0001}, respectively. Batch sizes are chosen to accommodate the GPU memory constraint. We use default Adam configurations in PyTorch. Finetuning learning rate, selected from a set {0.00001, 0.00002}, is used to finetune UL-token-seq and FACE. Of the four variants of FACE, we use FACE-OPR, which reportedly performs best.

C Transfer Learning

Pre-trained language models are widely used for downstream applications by adapting them to domain-specific distributions. Significant gains can be realized if F-^{2} softmax is successfully applied to fine-tune pre-trained language models, as most

Model	PPL	MSJ	SB	Uniq
From scratch	13.2	52.4	76.1	25.2k
Transferred	9.8	56.1	74.7	23.9k

Table 8: Evaluation results on transfer learning.

pre-trained language models are trained with standard softmax function. To validate our approach on transfer learning settings, we pre-train a language model on news datasets consisting of 10GB of articles and fine-tune the model on the Melo-Lyrics dataset. The results are shown in Table 8. The transferred model both increased the quality and diversity of the generation. However, the transferred model exhibits narrower vocabulary usage. We conjecture it is arisen by the vocabulary mismatch between datasets used for pre-training and fine-tuning. We plan to further investigate the vocabulary mismatch problem.

Winner		Loser	W-L	Dataset
F^2-Softmax		MLE	6-1	
F^2-Softmax		FACE	6-1	
F^2-Softmax		UL-token	6-1	
F^2-Softmax		UL-token-seq	5-2	
FACE	*beats*	MLE	4-3	Wikitext-103
UL-token		FACE	4-3	
UL-token-seq		FACE	4-3	
UL-token		MLE	5-2	
UL-token-seq		MLE	4-3	
UL-token		UL-token-seq	4-3	
F^2-Softmax		UL-token+seq	4-3	
F^2-Softmax		UL-token	4-3	
F^2-Softmax		MLE	4-3	
FACE		F^2-Softmax	4-3	
FACE	*beats*	MLE	4-3	Melo-Lyrics
UL-token		FACE	4-3	
FACE		UL-token+seq	4-3	
MLE		UL-token	4-3	
MLE		UL-token+seq	4-3	
UL-token		UL-token+seq	4-3	

(a) Results on pair-wise evaluations between models.

Rank	Model	Wins	Losses
1	F^2-Softmax	7	1
2	UL-token	5	3
3	FACE	4	4
4	MLE	2	6
	UL-token-seq	2	6

(b) Model ranking based on the number of wins in pair-wise evaluations.

Table 9: Analysis on the greedy sampling results.

(a) Wikitext-103

Metrics / Models	PPL	KLD	MS-Jaccard n=1	n=2	n=3	Self-BLEU n=1	n=2	n=3	Distinct n=1	n=2	n=3	Rep	Uniq
top-k=3													
MLE	**24.7**	1.51	52.1	35.6	24.3	93.4	83.2	69.7	45.1	71.9	83.0	0.67	8.48k
FACE	29.7	1.26	53.3	33.2	21.7	92.3	76.5	57.6	53.4	77.1	85.5	2.1	10.3k
UL-token	25.8	1.28	54.3	36.7	24.8	93.7	82.4	68.1	50.0	77.3	87.1	0.39	10.2k
UL-token+seq	27.5	1.33	50.6	35.4	23.5	**95.3**	83.4	66.9	57.6	86.6	94.2	**0.11**	10.6k
F^2-Softmax	25.6	**0.62**	**67.4**	**42.4**	**26.4**	93.3	**71.9**	**48.1**	**65.7**	**89.7**	**94.4**	0.33	**15.7k**
top-k=5													
MLE	**24.7**	1.34	55.9	37.9	25.6	94.1	82.8	67.7	50.0	78.1	87.8	0.73	9.2k
FACE	29.7	1.08	53.2	34.1	22.0	93.2	**77.0**	56.2	58.3	82.4	89.4	0.9	11.6k
UL-token	25.8	1.11	57.6	38.7	25.8	93.9	81.1	64.4	54.6	82.0	90.3	0.55	12.1k
UL-token+seq	27.5	1.07	55.8	36.5	23.7	**95.8**	82.1	62.5	61.4	90.4	**96.0**	**0.09**	12.2k
F^2-Softmax	25.6	**0.59**	**67.3**	**41.9**	**25.9**	93.4	70.6	45.9	**67.8**	**90.9**	95.0	0.22	**16.2k**
top-k=10													
MLE	**24.7**	1.12	60.2	40.6	**27.0**	94.0	80.9	63.1	54.7	83.0	91.0	0.89	10.3k
FACE	29.7	1.02	53.6	33.9	21.4	93.1	**74.4**	50.6	63.5	87.2	92.6	0.61	12.0k
UL-token	25.8	0.95	62.0	41.1	26.9	94.2	79.0	59.4	59.3	86.5	93.1	0.61	12.3k
UL-token+seq	27.5	0.93	60.9	39.3	25.0	**95.2**	78.0	55.0	65.9	92.9	96.8	**0.05**	12.9k
F^2-Softmax	25.6	**0.57**	**67.4**	**41.8**	25.6	93.5	69.5	44.2	**69.1**	**91.5**	**95.2**	0.22	**16.7k**
top-k=20													
MLE	**24.7**	0.95	63.5	**42.4**	**27.8**	94.2	78.9	58.8	58.9	86.5	93.0	0.39	11.5k
FACE	29.7	0.96	53.2	33.3	20.6	93.5	72.6	46.4	67.6	90.2	94.2	0.33	12.0k
UL-token	25.8	0.79	65.5	42.4	27.3	94.2	76.5	54.5	63.3	89.4	94.6	0.55	13.6k
UL-token+seq	27.5	0.80	64.8	41.0	25.4	**95.2**	**74.9**	49.4	69.9	94.4	97.2	0	**14.1k**
F^2-Softmax	25.6	**0.54**	**67.2**	41.2	25.2	93.4	68.2	42.3	**69.8**	91.9	95.3	0.16	17.1k
Human	-	-	-	-	-	95.2	74.1	50.9	69.1	92.1	95.8	0.11	15.2k

(a) Wikitext-103

(b) Melo-Lyrics

Metrics / Models	PPL	KLD	MS-Jaccard n=1	n=2	n=3	Self-BLEU n=1	n=2	n=3	Distinct n=1	n=2	n=3	Rep	Uniq
top-k=3													
MLE	**13.1**	0.39	64.6	44.9	31.8	**94.7**	77.7	59.5	31.8	43.3	51.3	11.7	22.0k
FACE	13.9	0.45	59.4	40.6	28.9	94.9	77.5	60.1	34.5	47.5	56.7	8.0	22.5k
UL-token	13.8	0.57	56.2	40.1	28.7	96.1	82.3	67.3	35.5	48.2	55.9	6.2	21.2k
UL-token+seq	16.6	0.66	50.7	36.1	25.4	97.5	86.1	69.9	41.4	**68.1**	**82.4**	1.1	20.6k
F^2-Softmax	13.2	**0.20**	**75.1**	**51.1**	**35.0**	95.7	**75.4**	**52.5**	**44.5**	57.2	63.8	11.3	**24.0k**
top-k=5													
MLE	**13.1**	0.39	64.2	45.1	31.8	95.7	80.0	61.9	37.2	51.8	61.3	0.3	22.8k
FACE	13.9	0.44	59.8	40.7	28.8	95.8	79.5	61.7	39.8	55.4	65.7	4.5	22.6k
UL-token	13.8	0.53	57.8	41.3	29.2	97.0	83.8	68.2	42.1	57.2	65.9	**2.6**	21.6k
UL-token+seq	16.6	0.59	53.3	37.4	25.8	**97.9**	86.1	68.0	48.6	**76.8**	**88.5**	0.5	20.7k
F^2-Softmax	13.2	**0.19**	**76.1**	**51.7**	**35.1**	96.3	**76.2**	**52.7**	**50.1**	63.9	70.4	7.5	**24.3k**
top-k=10													
MLE	**13.1**	0.37	65.3	46.0	32.1	96.7	82.1	63.3	46.5	64.3	74.2	2.4	22.1k
FACE	13.9	0.42	59.5	41.1	28.6	96.7	81.3	62.0	48.9	67.4	77.5	**2.1**	22.6k
UL-token	13.8	0.46	60.3	42.8	29.7	97.7	84.6	67.2	52.4	69.6	77.7	1.1	22.2k
UL-token+seq	16.6	0.51	55.6	38.5	25.9	98.4	85.2	63.4	**58.4**	85.6	93.3	0.1	21.3k
F^2-Softmax	13.2	**0.16**	**77.5**	**52.2**	**34.9**	97.0	**76.5**	**51.9**	57.4	**72.1**	**78.0**	4.3	**24.7k**
top-k=20													
MLE	**13.1**	0.34	67.6	47.3	32.4	**97.5**	83.1	62.2	56.3	75.5	84.0	1.1	22.4k
FACE	13.9	0.39	60.3	41.5	28.3	**97.5**	82.0	60.2	58.6	**78.1**	86.5	0.9	22.7k
UL-token	13.8	0.39	62.4	43.6	29.6	98.2	84.0	64.2	61.0	78.3	84.7	0.3	22.8k
UL-token+seq	16.6	0.45	58.3	39.4	25.7	98.7	79.1	57.1	67.2	90.7	95.0	0.07	22.3k
F^2-Softmax	13.2	**0.13**	**78.8**	**52.4**	**34.4**	**97.5**	76.1	50.3	64.0	78.7	**83.9**	2.5	**25.2k**
Human	-	-	-	-	-	97.5	76.6	53.2	62.5	77.5	82.5	2.1	27.9k

(b) Melo-Lyrics

Table 10: Evaluation results on different top-k.

Model	Texts	VR (%)
Prefix	= Clayton Kershaw = Clayton Edward Kershaw (born March 19 , 1988) is an American professional baseball pitcher for the Los Angeles Dodgers of Major League Baseball (MLB). A left @-@ handed starting pitcher, Kershaw	
MLE	began his career at **shortstop** for the Los Angeles **Dodgers** in 2006. He spent his entire career with the **Dodgers** before moving to the minor leagues in 2006 , where he spent two seasons and played for the New York Yankees before **retiring** in 2008. He was traded to the San Francisco Giants in 2011 , where he spent two seasons with the St. Louis Cardinals. Ker**shaw** was a two @-@ time MLB player in 2006.	6.0
F^2-Softmax	set a team record with eight National League runs scored in 110 innings pitched in his first three seasons. A right @-@ handed pitch attempts to make the **pitching swing** start , but his starting place as an extra @-@ major leagues team goes so far as to **boost** his chances of victory. He is best known for his efforts in both **pitching** and baseruster fastball (known as the " Wild Wings ") and his willingness to throw **pitches** into **bats**.	8.9

Model	Texts	VR (%)
Prefix	= Hed PE = Hed PE, also known as (hed) Planet Earth and stylized as ( ), is an American rock band from Huntington Beach, California. Formed in 1994, the band performs a style of	
MLE	music called the and has released two studio albums , The of which were released as the group's second studio album in 1995. The band has released four studio albums, two albums, two albums and two albums, as well as a live album of their own. = = History = = **Hed Pe** was born in **Huntington** Beach, California, to a family of American immigrants, in the town of **Huntington** Beach.	4.4
F^2-Softmax	jazz opera , jazz and **avant** @-@ **garde** jazz for the 1990s. It started under **Epic** Records in 2003 and received a $ 100 @,@ 000 grant from Columbia Records. The New York Knicks released their second album The Way **Beyond** the **Sunset**, in March 2003. **Led** by vocalist Eddie Murphy, **Hedwig** grew up in Portland and worked as a **consulting** keyboardist for Mercury Records, then joined Mercury Records in 1999.	12.3

Model	Texts	VR (%)
Prefix	= Little Gidding (poem) = Little Gidding is the fourth and final poem of T. S. Eliot's Four Quartets, a series of poems that discuss time, perspective, humanity, and salvation. It was first published in	
MLE	the spring of **1838**. The poem is the first of **Eliot**'s poems to be published by the public. The first of these is " The Great ", the first poem to be translated into the English language. **Eliot** wrote in his **diary** of the poem, " The Great ", that the poem "was written in the morning and afternoon of the night ".	4.6
F^2-Softmax	August 1969 with poetry about progress in **salvation**, and was praised by critics for its lyrical content. The poem was positively received by those who criticized its emphasis on **contemp**ation, lyrical content, and political themes, while **Eliot**'s critical **essays** about the poetry and short story were criticized. The poem is set in light of the faith, which focuses on the desire for freedom, change in nature, and self @-@ **esteem**.	6.0

Model	Texts	VR (%)
Prefix	= Olmec colossal heads = The Olmec colossal heads are at least seventeen monumental stone representations of human heads sculpted from large basalt boulders. The heads date from at least before 900 BC and are a	
MLE	type of stone carved from the walls of the Olmec colossal head of a small stone temple. It has a wide and narrow head and has a long and wide head that is a wide, broad face. The heads are long and long and **triangular** with rounded **corners**. The head is long and broad with long, curved **corners**. The legs are short and narrow, with short, straight sides, which are not wide.	3.4
F^2-Softmax	broad, highly curved structure that includes the uppermost **skeleton** of the king @-@ made objects. The body of Olmec colossal is **elongated** and consists of approximately eight layers on the **obverse**, followed by two large fragments. The **jaws** were composed of small bony **projections** formed by various iron **skeletons** and long thin **joints**. The skull is **unusually** similar to contemporary humans , with only one **skeleton** estimated at 0 @.@ 25 m (1 @.@ 5 ft) across.	11.1

Table 11: Generated samples.