Graphmax for Text Generation

Bin Liu   LIUBIN@SWUFE.EDU.CN
The Center of Statistical Research, School of Statistics,
Southwestern University of Finance and Economics, Chengdu, China

Guosheng Yin   GYIN@HKU.HK
Corresponding author
Department of Statistics and Actuarial Science,
The University of Hong Kong, Hong Kong, China

Abstract

In text generation, a large language model (LM) makes a choice of each new word based only on the former selection of its context using the `softmax` function. Nevertheless, the link statistics information of concurrent words based on a scene-specific corpus is valuable in choosing the next word, which can help to ensure the topic of the generated text to be aligned with the current task. To fully explore the co-occurrence information, we propose a `graphmax` function for task-specific text generation. Using the graph-based regularization, `graphmax` enables the final word choice to be determined by both the global knowledge from the LM and the local knowledge from the scene-specific corpus. The traditional `softmax` function is regularized with a graph total variation (GTV) term, which incorporates the local knowledge into the LM and encourages the model to consider the statistical relationships between words in a scene-specific corpus. The proposed `graphmax` is versatile and can be readily plugged into any large pre-trained LM for text generation and machine translation. Through extensive experiments, we demonstrate that the new GTV-based regularization can improve performances in various natural language processing (NLP) tasks in comparison with existing methods. Moreover, through human experiments, we observe that participants can easily distinguish the text generated by `graphmax` or `softmax`.

1. Introduction

The `softmax` operator is a fundamental component of text generation models, particularly in deep neural language models (Sutskever et al., 2011; Radford et al., 2019). In these models, a context embedding $\mathbf{z}$ is generated by the hidden layers of a deep neural network, such as a recurrent neural network (Sutskever et al., 2011) or a transformer (Radford et al., 2019). The output layer, which is typically a fully connected linear layer, is connected to the `softmax` function to compute a probability distribution for the next word choice.

The `softmax` is a continuous mapping function that transforms an n-dimensional input vector onto an $(n-1)$-simplex. The commonly adopted setting of `softmax` for text generation imposes a strong hypothesis that the probability of the next word is globally continuous over all the possible words in a dictionary. In linguistics, the arrangement of words and phrases needs to comply with syntax or popular language rules to create fluent and meaningful sentences, which should match the human habitual ways of expression. For example, when the context is a smartphone review in an e-commerce site, people prefer to using internet-style words rather than a formal style. In other words, the best selection of the next word could only be achieved by incorporating the human expression habits as

reflected by the current corpus. We refer to the human preference of expression of word choices as local fluency information. However, the traditional `softmax` function fails to exploit such fluency knowledge.

The total variation (TV) shows good performance in modeling the local spatial information, which has been successfully applied to image processing tasks (Lellmann et al., 2009; Chambolle et al., 2010; Jia et al., 2021). A naive way of computing TV over 2D signal images is given by

$$\text{TV}_{\mathbb{R}^2} = \sum_{i,j} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2},$$

where $x_{i,j}$ is a pixel value in a 2D image, and the subindices i, j are the coordinates of this pixel (Rudin, Osher, & Fatemi, 1992). In image processing, it is shown to produce piecewise smoothness regularization within a bounded $\mathbb{R}^2$ space.

We extend the total variation to natural language processing (NLP) tasks to explore the concurrent words (or fluency) information. In particular, we define a graphical total variation for the 1D sequential text data, which can quantify the local word choice variation based on a scene-specific corpus. As shown in Figure 1, the language rules and special expression preferences for word order choices in the corpus can be statistically quantified by a weighted directed word concurrent graph $G = (V, \mathbf{A}, \mathcal{C})$, where $V = \{v_i\}_{i=1}^N$ is a set of nodes (words), $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix, and $\mathcal{C}$ is the corpus from which the graph is constructed. Suppose that a mapping $\mathcal{F} : \mathcal{C} \longrightarrow G$ counts all the 2-gram tuples (w_t, w_{t+1}) of the sentences in $\mathcal{C}$ and then maps them to the directed edges of graph G as shown in Figure 1, where w_i is a word in the dictionary $\mathcal{D}$, corresponding to the vertex v_i of G. In practice, $\mathbf{A}_{i,j}$ is calculated by the mapping $\mathcal{F}$ based on the frequency of the corresponding 2-gram tuples (w_t, w_{t+1}) appeared in all the sentences of the corpus $\mathcal{C}$. Let $\mathbf{x}$ denote a graph signal over $G = (V, \mathbf{A}, \mathcal{C})$, and then we can calculate the graph-shifted text signal as $\mathbf{A}\mathbf{x}$ (Chen et al., 2014). We define the graph total variation (GTV) for the text data as $\|\mathbf{x} - \mathbf{A}\mathbf{x}\|_2$. In text generation, the graph signal $\mathbf{x}$ can be a probability distribution for the choice of the next word.

As discussed earlier, the traditional `softmax` function aims to link the dense output of a deep neural language model (LM) onto a probability simplex. It is globally continuous but fails to incorporate the local task-specific preferences into an LM explicitly. As a remedy, we propose a GTV-regularized `softmax` function, called `graphmax`, which can be incorporated into any global pre-trained LM to improve the local fluent satisfaction of the generated sentences. For example, the `graphmax` can be integrated into GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020) to obtain a scene-based text generation model. The contributions of our work are three-fold:

1. We utilize GTV to capture the local style of scene-specific text data.

2. We propose an n-gram mixture language model, `graphmax`, which can be plugged into a global pre-trained LM for a scene-specific task.

3. We evaluate the performance of `graphmax` on text generation and machine translation tasks. The experimental results demonstrate that `graphmax` outperforms the traditional `softmax` in scene-specific text generation and macine translation.
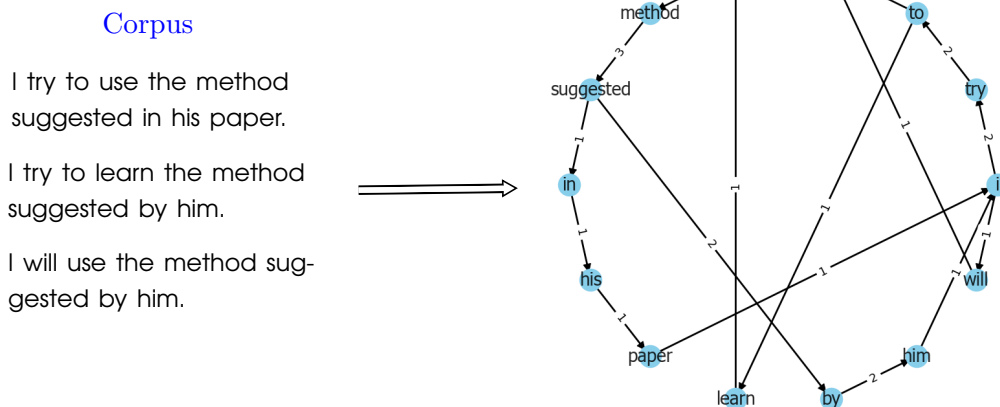
Figure 1: Construction of a weighted directed graph $G = (V, \mathbf{A}, \mathcal{C})$ (right panel) from a corpus (left panel), where $V = \{v_i\}_{i=1}^{N}$ is a set of nodes corresponding to the words in the dictionary of the corpus, $\mathbf{A}$ is an $N \times N$ weighted adjacency matrix, and $\mathcal{C}$ is the corpus that graph G is built upon. The directed edges denote the 2-grams of the corpus, where the weights are calculated based on the frequency of 2-grams in the corpus. A general n-gram model can be easily modeled by an n-order adjacency matrix $\mathbf{A}^n$.

The rest of the paper is organized as follows. In Section 2, we review the literature relevant to our method and present the architecture of `graphmax` for text generation. In Section 3, we propose to optimize the proposed model with the projected gradient descent, and Section 4 elaborates on the detailed algorithm and theoretical properties of `graphmax`. The experimental results are presented in Section 5, and Section 6 concludes with some remarks.

2. Methodology

In this section, we will present the fundamental framework of `graphmax`. To establish a solid foundation, we commence by conducting a literature review on the topic of `softmax`.

2.1 Literature Review on Softmax

The `softmax` function is fundamentally important for many applications in machine learning. It produces a probability distribution across multiple categories, enabling models' prediction capability, e.g., recognizing a digital handwritten image or generating the next word in a text sequence. However, the traditional `softmax` has limitations when applied to high-dimensional scenarios, such as text generation (Yang et al., 2018). To address this issue, Yang et al. (2018) proposed a mixture softmax function that improves the expres-

siveness of the `softmax` operator by replacing the single layer in the traditional `softmax` with an ensemble layer that has more parameters. In language modeling, however, a word can have multiple meanings depending on the context, which means that a single word may lead to multi-sense embeddings. Miao et al. (2019) proposed a kernelized Bayesian softmax function that enhances the expressiveness of the `softmax` operator by incorporating a multi-sense kernel function. This approach provides a more flexible and expressive framework for modeling word senses and contexts in language modeling tasks.

Gao and Pavel (2017) provided a comprehensive summary and analysis of the properties of the `softmax` function using the convex analysis and monotone operator theory. They demonstrated that the `softmax` function is a monotone gradient map of the log-sum-exp function. In text generation, the `softmax` function computes a probability distribution over the words in a dictionary and produces a high-dimensional sparse vector. Martins and Astudillo (2016) proposed a sparse softmax function that returns a sparse posterior distribution, where the loss function of the sparse `softmax` is analogous to the logistic loss. However, the existing works fail to consider the relationships among all components of the sparse vector. Overall, Gao and Pavel (2017) provided a valuable theoretical foundation for understanding the properties of the `softmax` function, while Martins and Astudillo (2016) offered a practical solution for handling the sparsity issue associated with `softmax` in text generation tasks.

To model the concurrent relationships among words, we propose a GTV-regularized `softmax` function where GTV characterizes the corpus-specific knowledge and improves the local fluency in text generation. In contrast, the total variation (TV), proposed by Rudin et al. (1992), is a spatial regularity that penalizes signals with excessive and possibly spurious local detail, which has been widely applied in computer vision and signal processing (Lellmann et al., 2009; Chambolle et al., 2010; Jia et al., 2021; Lellmann & Schnörr, 2011). For example, Lellmann et al. (2009) and Lellmann and Schnörr (2011) applied TV to multi-class image labeling, and Jia et al. (2021) proposed a 2D TV-regularized U-net for image segmentation.

The use of graphical techniques for signal processing has become increasingly popular in dealing with signals that have irregular structures, such as biological mechanisms, social networks, and citation networks. Recent studies by Chen et al. (2014) and Chen et al. (2015) have extended graphical signal recovery techniques to matrix completion (Liu et al., 2016) and semi-supervised learning (Lv et al., 2022), where TV over a graph is imposed on the object of matrix completion. Through investigation of the relevance of TV and graph energy in graph signal classification, Ahmed, Dare, and Boudraa (2017) found that TV is a compact and informative attribute for efficient graph discrimination while graph energy aims to quantify the complexity of the graph structure. These findings have important implications for understanding the relationship between signal processing and the underlying graph structure. In a related study, Raguet and Landrieu (2018) extended the cut-pursuit algorithm to the GTV regularization of functions with a separable nondifferentiable part.

Many existing works on GTV are focused on undirected graphs, while there has been increasing interest in directed graphs (Shi et al., 2019). In the context of text generation, the relationships between concurrent words are directed and weighted, which has motivated researchers to derive GTV over a weighted directed graph.

2.2 Conditional Text Generation

Deep LMs have been shown to be promising for conditional natural language generation with users' additional input. In controllable text generation tasks with different targets, different types of conditional constraints are often required, such as an image in image captioning (Anderson et al., 2017) or a style embedding vector in task-specific text generation (Ficler & Goldberg, 2017).

Traditional natural language generation methods rely on a large corpus, which is typically expensive to train. To address this issue, researchers have explored a new paradigm of pre-trained LMs, such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and BART (Lewis et al., 2020; Koto et al., 2020; Chen et al., 2022; Vougiouklis et al., 2020). By incorporating additional domain-specific codes (Keskar et al., 2019), such as sentiment labels (Dathathri et al., 2020) or attribute vectors (Yu, Yu, & Sagae, 2021), the goal is to modify the pre-trained LM with little fine-tuning cost. For example, the plug-and-play language model (PPLM) (Dathathri et al., 2020) builds a user-specified bag-of-words classifier on top of GPT-2 to increase the likelihood of the target attribute. Koncel-Kedziorski et al. (2019) proposed text generation from graph-based constraints, while it only inputs the knowledge graph embedding into a transformer-based model for text generation, which is an implicit way of using the graph. In contrast, our proposal explicitly regularizes the process of text generation by incorporating GTV over a weighted directed graph.

2.3 The Softmax Function

Given a context word sequence $\mathbf{w}_{1:t-1} = (w_1, \dots, w_{t-1})$, at each decoder time step, the feature map of $(w_1, \dots, w_{t-1})$ is denoted as $\mathbf{z} = \phi(\mathbf{w}_{1:t-1})$, where $\phi(\cdot)$ is a well-trained large-scale LM, $\mathbf{z} = (z_1, \dots, z_N) \in \mathbb{R}^N$, and N is the cardinality of a dictionary. Typically, we use the `softmax` function to link $\mathbf{z}$ with the prediction of the next word w_t. From the perspective of optimization, the `softmax` function corresponds to the minimizer of the following problem[1],

$$\min_{\mathbf{x}} \quad -\langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{x}, \log \mathbf{x} \rangle \qquad \text{s.t.} \quad \mathbf{1}^\top \mathbf{x} = 1, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^N$ is a graph signal (Chen et al., 2014), and $\mathbf{1}$ is an N-vector of all ones. The global minimizer of (1) is denoted by $\mathbf{x}^*$, whose i-th component is

$$x_i^* = \frac{e^{z_i}}{\sum_{i=1}^N e^{z_i}}, \quad i = 1, \dots, N, \tag{2}$$

where z_i is the i-th component of $\mathbf{z}$.

2.4 Graphmax: Graph Regularized Softmax

In text generation, an LM predicts the next word based on the history word sequence with a `softmax` function. Suppose the total number of words in a dictionary is N. An N-dimensional context representation vector $\mathbf{z}$ can be taken as the input for the `softmax` mapping as shown in Equation (2). By constructing a directed graph $G = (V, \mathbf{A}, \mathcal{C})$ of the

---

1. The object of the optimization is the conjugate function of the negative entropy $\langle \mathbf{x}, \log \mathbf{x} \rangle$.

words in the dictionary based on a large human corpus, we can take this graph G as an elementary filtering operation that replaces a signal coefficient for each word with a weighted linear combination of coefficients at its neighboring nodes of G, where $V = \{v_i\}_{i=1}^N$ is the node set, $\mathbf{A}$ is an $N \times N$ weighted adjacency matrix, and $\mathcal{C}$ is the corpus that graph G is built upon.

The smoothness of signals over G can be quantified by the graph total variation (GTV),

$$\text{GTV} = \left\| \mathbf{x} - \frac{1}{|\mu_{\max}|}\mathbf{A}\mathbf{x} \right\|_2^2,$$

where $\mu_{\max}$ denotes the largest eigenvalue of the adjacency matrix $\mathbf{A}$. In practice, we can normalize the adjacency matrix $\mathbf{A}$ to make its maximum eigenvalue $\mu_{\max} = 1$ to simplify the computation. Specifically, for a node v_i, its outgoing edges have weights $1/\deg(v_i)$, where $\deg(v_i)$ is the out-degree of v_i (the number of outgoing edges). Via matrix normalization, we set $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$ (Schlichtkrull et al., 2018; Shi et al., 2019), where $\mathbf{D}$ is a diagonal matrix with the i-th diagonal element $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij} + \epsilon$ and ϵ is a small number to avoid division by zero. The normalization of the adjacency matrix ensures that the largest eigenvalue $\mu_{\max} = 1$.

By imposing GTV on the optimization problem in Equation (1), we obtain a graph-regularized `softmax`,

$$\begin{aligned} \min_{\mathbf{x}} \quad & -\langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{x}, \log \mathbf{x} \rangle + \lambda \|\mathbf{x} - \tilde{\mathbf{A}}\mathbf{x}\|_2^2 \\ \text{s.t.} \quad & \mathbf{1}^\top \mathbf{x} = 1, \quad \mathbf{x}, \mathbf{z} \in \mathbb{R}^N, \end{aligned} \tag{3}$$

where $\tilde{\mathbf{A}}$ is a normalized adjacency matrix of the directed graph and $\lambda > 0$ is a tuning parameter. We name the solution to (3) as `graphmax`. The GTV regularizer in (3) forces $\mathbf{x}$ to be close to $\tilde{\mathbf{A}}\mathbf{x}$ where $\tilde{\mathbf{A}}$ is a (normalized) adjacency matrix of the directed graph that contains local knowledge from scene-specific corpus. Note that $\tilde{\mathbf{A}}$) is like a projection matrix which projects $\mathbf{x}$ to the space of $\tilde{\mathbf{A}}$). Imagining $\tilde{\mathbf{A}} = \mathbf{I}$, an identity matrix, the regularizer would not exist (equal to zero), which indicates the components of $\mathbf{x}$ are all independent so that no local information on scene-specific text is utilized. By forcing $\mathbf{x}$ to be close to $\tilde{\mathbf{A}} = \mathbf{I}$, the local scene-specific knowledge would be incorporated in predicting the choice probabilities for the next word.

3. Optimization

It is difficult to solve Equation (3) with traditional methods, such as the gradient descent or Lagrangian method. Instead, we propose to optimize it with the projected gradient descent detailed as follows.

3.1 Projected Gradient Descent

In the optimization problem of Equation (3), the feasible set $\Delta = \{\mathbf{x} | \mathbf{1}^\top \mathbf{x} = 1, 0 \leq x_i \leq 1\}$ is a probability simplex, which is a convex set. The graph regularized objective function is

$$f(\mathbf{x}) = -\langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{x}, \log \mathbf{x} \rangle + \lambda \|\mathbf{x} - \tilde{\mathbf{A}}\mathbf{x}\|_2^2 \,,$$

which can be shown to be strictly convex as follows. We examine the second-order condition of $f(\mathbf{x})$,

$$\nabla^2 f(\mathbf{x}) = \text{diag}\left(\frac{1}{\mathbf{x}}\right) + 2\lambda(\mathbf{I} - \tilde{\mathbf{A}})^\top(\mathbf{I} - \tilde{\mathbf{A}}). \tag{4}$$

Because $\mathbf{I}$ is an identity matrix and $\tilde{\mathbf{A}}$ is a normalized adjacency matrix of the directed graph G, $(\mathbf{I} - \tilde{\mathbf{A}})$ is non-singular. That is, $(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{x} = \mathbf{0}$ holds only if $\mathbf{x} = \mathbf{0}$. Hence, the matrix $(\mathbf{I} - \tilde{\mathbf{A}})^\top(\mathbf{I} - \tilde{\mathbf{A}})$ is positive definite, which leads to the conclusion that the Hessian matrix in (4) is positive definite as $\lambda > 0$.

The optimization problem in Equation (3) can be solved by the projected gradient descent algorithm (Duchi et al., 2008; Jain & Kar, 2017), which involves two steps: the gradient descent step and a subsequent projection of the gradient onto a probability simplex. Specifically, the two steps are given by

$$\begin{cases} \mathbf{a}^{t+1} \leftarrow \mathbf{x}^t - \alpha\nabla f(\mathbf{x}^t), & \text{gradient descent}, \\ \mathbf{x}^{t+1} \leftarrow \Pi_{\mathcal{C}}(\mathbf{a}^{t+1}), & \text{gradient projection}, \end{cases} \tag{5}$$

where α is a learning rate, $\mathbf{a}^{t+1}$ is an intermediate updated gradient at iteration $t+1$ and $\Pi_{\mathcal{C}}(\cdot)$ is the projection operator.

3.2 Projection onto Probability Simplex Δ

Given any closed set $\Delta \subset \mathbb{R}^N$, for any point $\mathbf{a} \in \mathbb{R}^N$ the projection operator $\Pi_\Delta(\cdot)$ is defined as

$$\Pi_\Delta(\mathbf{a}) = \arg\min_{\mathbf{x}\in\Delta} \|\mathbf{x} - \mathbf{a}\|_2.$$

In the case of $\Delta = \{\mathbf{x}|\mathbf{1}^\top\mathbf{x} = 1, 0 \leq x_i \leq 1\}$, the definition of the projection operation $\Pi_\Delta(\cdot)$ can be reformulated as an optimization procedure,

$$\min_{\mathbf{x}} \quad \|\mathbf{x} - \mathbf{a}\|_2 \quad \text{s.t.} \quad \sum_{i=1}^N x_i = 1, \quad x_i \geq 0, \tag{6}$$

which can be solved by the Lagrangian method (Duchi et al., 2008). By applying the standard KKT conditions for the Lagrangian of Equation (6), the i-th component of the optimal $\mathbf{x}^*$ can be obtained as

$$x_i^* = \max\{a_i + \gamma, 0\}, \quad i = 1, \ldots, N. \tag{7}$$

where γ can be computed from a vector of temporal variables $\mathbf{b} = (b_1, \ldots, b_N) \in \mathbb{R}^N$ and an integer k,

$$\gamma = \frac{1}{k}\left(1 - \sum_{j=1}^k b_j\right).$$

The vector $\mathbf{b} = (b_1, \ldots, b_N)$ can be easily obtained by applying a sort function $\text{sort}(\cdot)$ to the vector $\mathbf{a} \in \mathbb{R}^N$,

$$(b_1, \ldots, b_N) = \text{sort}(\mathbf{a}),$$

such that $b_1 \leq \cdots \leq b_N$. The integer k, $1 \leq k \leq N$, can be calculated from the sorted vector $\mathbf{b}$,

$$k = \arg\max_{1 \leq j \leq N} \left\{ b_j + \frac{1}{j} \left(1 - \sum_{i=1}^{j} b_i \right) > 0 \right\}.$$

4. Computation and Theoretical Properties

In this section, we will outline the procedure of the `graphmax` algorithm and subsequently delve into a comprehensive analysis of its theoretical properties.

4.1 Computational Complexicity

Algorithm 1 outlines the procedure for the `graphmax` algorithm. In particular, step 3 computes the current gradient using gradient descent, and steps 4–7 correspond to the gradient projection $\Pi_\Delta(\cdot)$. The time complexity of the gradient projection onto the probability simplex is dominated by the cost of sorting the components of vector $\mathbf{a}$, which is $\mathcal{O}(N \log N)$. The maximum number of iterations is T, and thus the complexity of Algorithm 1 is $\mathcal{O}(TN \log N)$.

In practice, the value of T is typically small. In our experiments, we set $T = 20$ and both the learning rate and the convergence threshold as 10^{-4}, which are reasonable choices for most applications.

Algorithm 1: Projected Gradient Descent for `graphmax`.

    Input: Learning rate α, maximum number of iteration steps T

1   $\mathbf{x}^1 \leftarrow \mathbf{0}$;
2   **while** $t < T$ **do**
3      $\mathbf{a}^{t+1} \leftarrow \mathbf{x}^t - \alpha \nabla f(\mathbf{x}^t)$;
4   Sort $\mathbf{a}^{t+1} = (a_1^{t+1}, \ldots, a_N^{t+1})$ as $(b_1^{t+1}, \ldots, b_N^{t+1})$;
5   Find $k = \max\{1 \leq j \leq N : b_j^{t+1} + (1 - \sum_{i=1}^{j} b_j^{t+1})/j > 0\}$;
6   Calculate $\gamma = (1 - \sum_{i=1}^{k} b_j^{t+1})/k$;
7   Calculate $x_i^{t+1} = \max\{a_i^{t+1} + \gamma, 0\}$;
8   If $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2 < 10^{-4}$ then break;
9   **end**
    Output: A near-optimal minimum $\mathbf{x}^*$ of the `graphmax`.

4.2 How `graphmax` Works

Step 3 of Algorithm 1 serves as an interface between the pre-trained language model and the word concurrent graph. In this step, we calculate the gradient $\nabla f(\mathbf{x}^t)$ of the objective function, as shown in Equation (3). The gradient consists of three distinct parts,

$$\nabla f(\mathbf{x}^t) = \underbrace{-\mathbf{z}}_{\text{global knowledge from LM}} + \log \mathbf{x}^t + 1 + \underbrace{2\lambda(\mathbf{I} - \tilde{\mathbf{A}})^\top (\mathbf{I} - \tilde{\mathbf{A}})\mathbf{x}^t}_{\text{local knowledge from scene-specific corpus}}, \qquad (8)$$

where the first part $-\mathbf{z}$ is the decoder output of a pre-trained LM, such as GPT-2. As explained in Section 3.1, we have $\mathbf{z} = \phi(\mathbf{w}_{1:t-1})$, where $\phi(\cdot)$ is the well-trained large-scale LM, so that $\mathbf{z}$ carries the global linguistic knowledge from the pre-trained LM. It is important to note that the traditional `softmax` only receives $\mathbf{z}$ as the input and produces a probabilistic distribution as indicated by Equation (2). The second part in (8), $\log \mathbf{x}^t + 1$, depends solely on $\mathbf{x}^t$. The third part in (8), $2\lambda(\mathbf{I} - \tilde{\mathbf{A}})^\top(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{x}^t$, is derived from the word concurrent graph, which is constructed from a scene-specific corpus (as depicted in Figure 1). This part represents the local knowledge relevant to the target task. The local knowledge is modeled as a second-order polynomial of the normalized adjacency matrix, denoted by $\tilde{\mathbf{A}}$. It is worth noting that $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{A}}^2$ correspond to 2-gram and 3-gram models, respectively. Therefore, the `graphmax` with $(\mathbf{I} - \tilde{\mathbf{A}})^\top(\mathbf{I} - \tilde{\mathbf{A}})$ can be viewed as a (2,3)-gram mixture model with a penalty of 2λ.

Equation (8) demonstrates that the local knowledge and the global pre-trained LM operate in a plug-in mode. This property allows the local knowledge to be easily integrated into any pre-trained LM using the proposed `graphmax`. Unlike traditional conditional natural language generation models that modify $\mathbf{z}$ with additional inputs, `graphmax` does not alter $\mathbf{z}$. Therefore, there is no need to fine-tune the pre-trained model. To perform task-specific text generation, we can simply attach the `graphmax` module with local knowledge to a global pre-trained model.

4.3 Theoretical Properties

For the `softmax` defined in Equation (2), we include the following proposition for completeness and comparison.

Proposition 1. *(Martins & Astudillo, 2016) Let $\mathbf{z} = (z_1, \ldots, z_N) \in \mathbb{R}^N$ be a feature map vector. If $z_i \le z_j$, then the following inequalities hold,*

$$\mathit{softmax}_j(\mathbf{z}) - \mathit{softmax}_i(\mathbf{z}) \ge 0,$$

$$\mathit{softmax}_j(\mathbf{z}) - \mathit{softmax}_i(\mathbf{z}) \le \frac{1}{2}(z_j - z_i),$$

where $\mathit{softmax}_i(\mathbf{z})$ denotes the i-th component of $\mathit{softmax}(\mathbf{z})$.

The proof can be referred to Martins and Astudillo (2016). The proposed `graphmax` possesses a similar property.

Proposition 2. *For a feature map vector $\mathbf{z} = (z_1, \ldots, z_N) \in \mathbb{R}^N$, if $z_i \le z_j$, then the following inequalities hold,*

$$\mathit{graphmax}_j(\mathbf{z}) - \mathit{graphmax}_i(\mathbf{z}) \ge 0,$$

$$\mathit{graphmax}_j(\mathbf{z}) - \mathit{graphmax}_i(\mathbf{z}) \le z_j - z_i,$$

where $\mathit{graphmax}_i(\mathbf{z})$ denotes the i-th component of $\mathit{graphmax}(\mathbf{z})$.

The proof of Proposition 2 is provided in Appendix A.2. This proposition shows that the `graphmax` function shares similar properties with the `softmax` function. As is well-known, the `softmax` function maps a vector of real numbers to a probability distribution over those

numbers in a consistent manner, such that larger numbers correspond to larger probabilities. By comparing the two propositions, we observe that not only does the `graphmax` function maintain this consistency but it also has an upper bound.

5. Experiments

Experimental Settings All models are implemented using Pytorch 1.7 on an Intel(R) Xeon(R) CPU E5-2680 v4 2.40GHz, Tesla K80 GPU, and 128G memory, based on the Ubuntu 16.04 platform. The parameters in Algorithm 1 are specified as follows: The maximum number of iteration steps for the `graphmax` algorithm is set as $T = 20$, the learning rate α is 10^{-4}, and the convergence threshold is 10^{-4}. Specifically, the iteration process stops when the l_2 norm of the difference between consecutive iterations, $\|\mathbf{x}^{t+1}-\mathbf{x}^t\|_2$, is less than 10^{-4}. The reported results are averaged over 100 independent runs.

Datasets and Baseline Methods We choose the Amazon (Zhang, Zhao, & LeCun, 2015) and Yelp (McAuley & Leskovec, 2013; Zhang et al., 2015) datasets for general text generation, WMT'16 (Bojar et al., 2016) and WMT'21 corpora (Tran et al., 2021) for machine translation. The baseline methods are chosen as follows,

- For text generation: GPT-2 (Radford et al., 2019), CTRL (Conditional Transformer Language model) (Keskar et al., 2019), PPLM (Plug-and-Play LM) (Dathathri et al., 2020), and OPT (Open Pre-trained Transformer) (Zhang et al., 2022)[2]; LLAMA2 (Touvron et al., 2023).

- For machine translation: BART (Lewis et al., 2020) and mBART (Liu et al., 2020).

Generally, we set the baselines at their best configurations as reported in the respective papers. For examples, we set PPLM with the sentiment label, KL-scale 0.1, GM-scale 0.95, and step size 0.05. We assign CTRL with a review domain control code in the corresponding experiments.

Evaluation Metrics We employ a hybrid approach for performance evaluation, combining both human and automatic metrics. Specifically, we utilize two well-known automatic metrics: BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002) and perplexity (PPL), for assessing the quality of text generation and machine translation.

BLEU counts the matches of n-grams in the generated text with n-grams in the reference text. For instance, a 1-gram comparison would consider each individual token, whereas a 2-gram comparison would assess every pair of words. We employ BLEU-2, BLEU-3, BLEU-4, BLEU-5 scores to measure the $\{2, 3, 4, 5\}$-gram matching between the generated text and reference text respectively.

Another important metric is the sentence fluency, which depicts the naturalness of human expressions. However, assessing the fluency of a natural language generation system is challenging (Kann, Rothe, & Filippova, 2018). Therefore, we use human testers to evaluate the fluency of the generated text as a supplement.
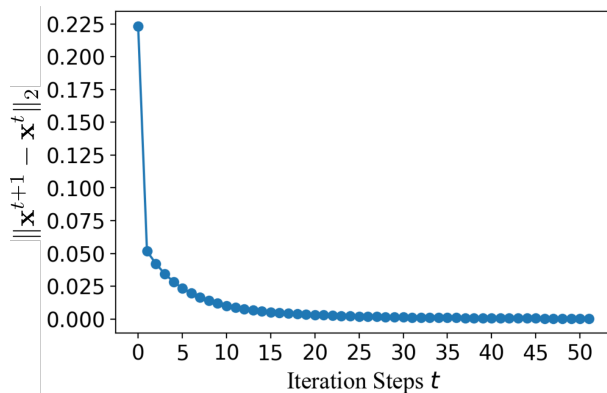
---

2. https://huggingface.co/facebook/opt-125m

Figure 2: The visualization process of Algorithm 1, i.e., $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2$ versus iteration steps t.

Graph Construction   We conceptualize a corpus as a directed graph, wherein the vertices represent individual words, and the edges denote the sequential relationship between two adjacent words in a sentence. The direction of each edge is determined by the word link of a 2-gram. We then tally the frequency of the 2-grams appeared in our corpus as the weight of the corresponding edge. This graphical representation intuitively captures the natural flow of expression.

To construct these graphs, we utilize datasets employed in our experimental analysis. For instance, we construct the word concurrent graph by leveraging datasets from food/restaurant reviews or e-commerce reviews sourced from Yelp. These datasets exhibit distinct characteristics that diverge from the typical structure of general text. Notably, the online review language tends to incorporate more colloquial phrases, shorter sentences, and internet slangs. These unique features of review text present a convenient opportunity to juxtapose the local fluency of our proposed model with that of other approaches.

The concurrent graph is constructed based on the statistics of 2-grams, which corresponds to a 2-gram model. However, this construction can be easily extended to an n-gram model by utilizing a higher-order adjacency matrix, denoted as $\tilde{\mathbf{A}}^{n-1}$. As discussed in Section 4, the proposed `graphmax` acts as a $\{2,3\}$-gram mixture model when the input is a first-order normalized adjacency matrix $\tilde{\mathbf{A}}$. For an n-order input matrix $\tilde{\mathbf{A}}^n$, `graphmax` models an $\{n+1, n+2, \ldots, 2n+1\}$-grams mixture. In practice, we can even augment the input matrix by replacing $\tilde{\mathbf{A}}$ with $\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2 + \cdots + \tilde{\mathbf{A}}^n$, which yields a more complex n-grams mixture model. This approach further captures higher-order word dependencies and enhances the ability to model complex language structures.

Convergence of `graphmax`   The core part of our model is the projection gradient descent, which is used to calculate the GTV regularized `softmax`. Unlike the traditional `softmax`, which has a closed form solution, the proposed `graphmax` needs to be computed iteratively. Fortunately, the computational cost of this iterative procedure is still affordable. Taking the largest dataset Amazon as an example, the corresponding directed graph contains a total of $N = 50527$ words, and the adjacency matrix $\hat{\mathbf{A}}$ is of size 50527×50527. Figure 2

GPT-2	***You do not want to read this book if you don't want to change.*** *Shepherd Tebbit Reading is business. It's business in the eyes of God. Hertha Wise Sorry to end this, but this has been a hunch for a couple of months now. See more posts on wikipedia.com. Also read Martin Walker's The Last Dance. Updated 5:18 PM*
CTRL	***You do not want to read this book if you don't want to change.*** *It is not a guide to your own life, it is not an answer to your own questions. It is a warning to others to take the time to read this book. It is not an easy book to follow, and you will be left with a lot of questions that are going to be difficult to answer. This book is not about being afraid of change. It is about being a true believer. I think you are probably going to find a lot of mistakes, and that's okay. In this book, we are going to learn a lot and learn how to change our life. That means that you are going to find a lot of happiness.*
PPLM	***You do not want to read this book if you don't want to change.*** *The first part is an explanation of what it is like to play with a computer and then to get your head around it. You might just want to play with a computer, or you might want to use a calculator or a phone to get a picture of what to do. I really think it's important that you understand the basics of your game.*
OPT	***You do not want to read this book if you don't want to change.*** *This is a very inspiring book, which encourages you to change your life and become a much better person. If you're unhappy with the way you are, you should read this book! The author describes in the book several ways to improve yourself and your personality.*
New	***You do not want to read this book if you don't want to change.*** *Blossom flows, thickly. Whatever you say it has become pretty clear. You are learning from the book and finding things that you must learn. You are learning from the criticism, the griping, and the washing up. What you learn and remember about it is that it is what we learned from it, and so what we learn will help us get back into this new energy of determination that helped us overcome the most difficult elements of battle. Years ago during our fighting back before we even got a chance to fight back, we will never see a great American.*

Table 1: Comparisons of book reviews generated by GPT-2, CTRL, PPLM, OPT and GPT-2+`graphmax` (denoted as 'New') on generating product reviews, with the local graph of the `graphmax` constructed from the Amazon review corpus.

shows that the `graphmax` converges in 20 iteration steps, where the X-axis represents the iterative steps t and the Y-axis is the l_2 gap between two adjacent iterations $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2$. Table 3 presents a comparison of the average time required for inferring the next word in five experimental scenarios, between LLAMA2 and LLAMA2+`graphmax`. It is worth noting that the time cost associated with `graphmax` is approximately 15–17 times higher than that of the original LLAMA2 implementation, which is consistent with the findings derived from the analysis of time complexity.

5.1 General Text Generation

E-Commerce Product Review Generation In the e-commerce product review generation experiment, we construct a graph with 4 million product reviews in the Amazon

Methods	BLEU-2	BLEU-3	BLEU-4	BLEU-5
GPT-2	$0.983_{\pm0.017}$	$0.919_{\pm0.024}$	$0.788_{\pm0.041}$	$0.615_{\pm0.021}$
GPT-2+graphmax	$\mathbf{0.989}_{\pm0.021}$	$\mathbf{0.945}_{\pm0.062}$	$\mathbf{0.832}_{\pm0.026}$	$\mathbf{0.660}_{\pm0.015}$
CTRL	$0.965_{\pm0.047}$	$0.912_{\pm0.031}$	$0.832_{\pm0.017}$	$0.584_{\pm0.014}$
CTRL+graphmax	$\mathbf{0.971}_{\pm0.064}$	$\mathbf{0.947}_{\pm0.049}$	$\mathbf{0.854}_{\pm0.022}$	$\mathbf{0.589}_{\pm0.031}$
PPLM	$0.988_{\pm0.028}$	$0.935_{\pm0.032}$	$0.761_{\pm0.049}$	$0.631_{\pm0.051}$
PPLM+graphmax	$\mathbf{0.991}_{\pm0.034}$	$\mathbf{0.946}_{\pm0.084}$	$\mathbf{0.812}_{\pm0.049}$	$\mathbf{0.673}_{\pm0.021}$
OPT	$0.995_{\pm0.045}$	$0.967_{\pm0.049}$	$0.876_{\pm0.038}$	$0.788_{\pm0.011}$
OPT+graphmax	$\mathbf{0.998}_{\pm0.019}$	$\mathbf{0.978}_{\pm0.022}$	$\mathbf{0.912}_{\pm0.011}$	$\mathbf{0.817}_{\pm0.032}$
LLAMA2	$\mathbf{0.988}_{\pm0.034}$	$0.987_{\pm0.067}$	$0.910_{\pm0.038}$	$0.845_{\pm0.011}$
LLAMA2+graphmax	$\mathbf{0.998}_{\pm0.026}$	$\mathbf{0.988}_{\pm0.045}$	$\mathbf{0.922}_{\pm0.011}$	$\mathbf{0.867}_{\pm0.032}$

Table 2: The BLEU scores of GPT-2, CTRL, PPLM, OPT, LLAMA2, and the proposed methods on generating product reviews ($\pm$ standard deviation), averaged over five cross-validation folds. The local graph is constructed with the Amazon corpus, and the best results are highlighted in boldface.

LLAMA2	Prompt 1	Prompt 2	Prompt 3	Prompt 4	Prompt 5
without graphmax	0.092	0.085	0.097	0.073	0.103
with graphmax	1.21	1.41	1.43	1.14	1.24

Table 3: Comparison of the average inference time (in seconds), with the base model LLAMA2 (Touvron et al., 2023) and the local corpus Amazon.

corpus (Zhang et al., 2015). Table 1 shows representative book reviews generated by the four baseline methods (GPT-2, CTRL, PPLM, OPT) as well as the proposed graphmax incorporated into GPT-2 (GPT-2+graphmax). From the perspective of narrative logic, PPLM, GPT-2, OPT, and our method perform better than CTRL. The sentences generated by CTRL often appear as paraphrases of a start sentence. On the other hand, GPT-2, OPT, and our method elaborate more on the topic and engage in further discussions. The book review generated by GPT-2 contains extensive information, such as another book's title, people related to this book, etc., while such abundance of multi-source information can be distracting. Similarly, although OPT may highly recommend a book, the basis for such a recommendation is not attributed to the book's content. In contrast, the proposed GPT-2 with graphmax attempts to gradually persuade the reader by focusing on the book's content. It utilizes sentences of varying lengths, which resembles genuine reviews.

Table 2 shows the BLEU scores of the text generated by the baseline methods and our model. The relative improvement of GPT-2 with `graphmax` over the baselines is more significant in the BLEU-5 metric than the BLEU-2 metric.

Food Review Generation This experiment constructs a graph based on the Yelp food review (McAuley & Leskovec, 2013; Zhang et al., 2015) dataset, which contains more than 700,000 food reviews.

GPT-2	***Now I know why people review this place so highly***, *it's even ridiculous that a local show will automatically get chosen as the greatest movie in the annals of writing. I mean even though it's written by the ones I love in all these years! Just take a glance. How many times have you seen that absolutely 100 dollar DVD and read the comments? That only proves that you are as mature as I am and as invested in The Cove as I am in Denzel Washington. Ridiculous. But I couldn't get a dentist just looking at them and thinking "well, the worst thing that could happen in a movie is the less you have to do it.*
CTRL	***Now I know why people review this place so highly***. *This is a place for good food, good music and amazing people. I was here for an informal "crawl" on a Thursday night, and I saw a young woman and some young guys in their late 20s, all sitting on the bench, drinking beer, with their feet dangling down in a pile of trash. We had been to a lot of great places here, but nothing beats a little party on an old-fashioned Saturday night and getting in to the groove and making some good friends.*
PPLM	***Now I know why people review this place so highly***. *There's always a lot of reviews here for anything, but it's not a place where we're talking about an interesting place. But I'm not sure. This article was originally published at the University of Massachusetts Press.*
OPT	***Now I know why people review this place so highly***, *Now I know why people review this place so highly, it's because it's a great place to stay. I've been staying here for years and I've never been disappointed. The rooms are clean and comfortable, the staff is friendly and helpful, and the location is great. I've stayed here for business and pleasure.*
New	***Now I know why people review this place so highly***, *for any reason—shear interest, get bored, try new things. However, I've been coming here for a few weeks now, and when I first opened the door and saw this bar, my Roast was completely filled with treats, salads, and soul food. I really enjoyed everything I've had in here for the past several days. The food came out in such a timely fashion that I stopped and have been keeping an eye on the Trolley Room Lounge bar (with its lights on) a little since I finished my morning. The food is very good, as I've been ordering fresh baked foods since my college days, I still just couldn't stop with homemade sweet potatoes. I thought the bakery was pretty good. Overall, the atmosphere was clean, laid back, and friendly, and it was friendly.*

Table 4: Comparisons of restaurant reviews generated by GPT-2, CTRL, PPLM, OPT and GPT-2+`graphmax` (denoted as 'New'), with the local graph constructed by the Yelp corpus.

Methods	BLEU-2	BLEU-3	BLEU-4	BLEU-5
GPT-2	$0.992_{\pm 0.044}$	$0.931_{\pm 0.034}$	$0.782_{\pm 0.032}$	$0.590_{\pm 0.065}$
GPT-2+graphmax	$\mathbf{0.994}_{\pm 0.038}$	$\mathbf{0.953}_{\pm 0.022}$	$\mathbf{0.834}_{\pm 0.024}$	$0.658_{\pm 0.008}$
CTRL	$\mathbf{0.989}_{\pm 0.047}$	$0.912_{\pm 0.031}$	$0.743_{\pm 0.017}$	$\mathbf{0.607}_{\pm 0.014}$
CTRL+graphmax	$0.971_{\pm 0.054}$	$\mathbf{0.947}_{\pm 0.039}$	$\mathbf{0.854}_{\pm 0.023}$	$0.589_{\pm 0.041}$
PPLM	$\mathbf{0.992}_{\pm 0.32}$	$0.941_{\pm 0.013}$	$0.792_{\pm 0.049}$	$0.621_{\pm 0.026}$
PPLM+graphmax	$0.991_{\pm 0.014}$	$\mathbf{0.946}_{\pm 0.033}$	$\mathbf{0.812}_{\pm 0.019}$	$\mathbf{0.673}_{\pm 0.034}$
OPT	$0.997_{\pm 0.013}$	$0.978_{\pm 0.031}$	$0.845_{\pm 0.019}$	$0.713_{\pm 0.021}$
OPT+graphmax	$\mathbf{0.998}_{\pm 0.029}$	$\mathbf{0.985}_{\pm 0.042}$	$\mathbf{0.876}_{\pm 0.009}$	$\mathbf{0.723}_{\pm 0.012}$
LLAMA2	$\mathbf{0.998}_{\pm 0.007}$	$0.995_{\pm 0.078}$	$\mathbf{0.889}_{\pm 0.039}$	$0.843_{\pm 0.062}$
LLAMA2+graphmax	$\mathbf{0.998}_{\pm 0.029}$	$\mathbf{0.996}_{\pm 0.012}$	$0.887_{\pm 0.034}$	$\mathbf{0.876}_{\pm 0.016}$

Table 5: The BLEU scores of GPT-2, CTRL, PPLM, LLAMA2, and OPT with or without graphmax on generating restaurant reviews (± standard deviation), averaged over five cross-validation folds.

Table 4 presents examples of the restaurant reviews generated by the baselines and the proposed model. From the view of the narrative logic, we observe that GPT-2 explores too much information, making it difficult to capture the main story-line "a place". In contrast, CTRL, PPLM, OPT, and our GPT-2+graphmax method can always make the text focused on the core story. We also see that OPT appears to resemble a blog post authored by a seasoned food enthusiast, utilizing a formal tone of expression rather than an internet review. In contrast, the proposed method (GPT-2+graphmax) prefers a freestyle of text that is much closer to a real e-commerce review written by human users. In contrast, our model tends to use shorter phrases, such as the "*shear interest*", "*get bored*", "*try new things*", "*laid back*", and even a repetitive style of a statement, such as "*and friendly, and it was friendly*", which are more similar to real food/restaurant reviews on e-commerce sites. In other words, the fluency of the generated text approaches that of real reviews. The comparison between GPT-2 and our method (GPT-2+graphmax) suggests that the local knowledge obtained in Equation (8) is indeed important in text generation.

We report the BLEU-2 to BLEU-5 scores (Papineni et al., 2002) in Table 5 to evaluate the overall performance for the food review generation. We set $\lambda = 1.0$. The BLEU-2 to BLEU-5 scores on generating long food reviews using the models with graphmax (+graphmax) are consistently better than PPLM, CTRL, OPT, and GPT-2 without graphmax. For baseline LLAMA2, graphmax achieves better BLEU-3 and BLEU-5 but worse BLEU-4 than LLAMA2 without graphmax.

The left panel of Figure 3 shows the perplexity (PPL) of the generated text by the competing methods and our model. The local graph is constructed with the Amazon corpus.

Different from BLEU, perplexity is relatively low when the contribution of the regularized local knowledge (with `graphmax`) is small. The right panel of Figure 3 summarizes the perplexity of generated food reviews. The results indicate that local knowledge benefits the improvement of perplexity for all the baseline models.
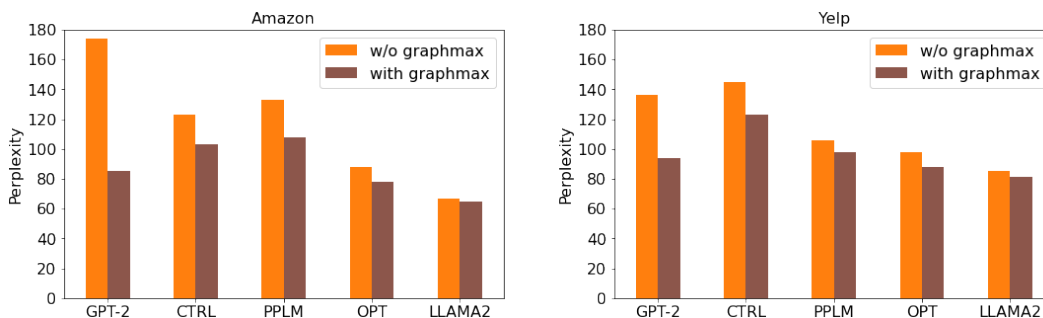

Figure 3: Comparisons of perplexity using GPT-2, CTRL, PPLM, OPT, and LLAMA2 with or without (w/o) `graphmax` on generating product reviews, with the local graph constructed from the Amazon and Yelp corpuses.

Choice of λ   The penalty parameter λ for the GTV in Equation (3) plays an important role in our model. A larger value of λ indicates that `graphmax` (local scene-specific knowledge) contributes more than the LM (global knowledge) on predicting the next word. We choose λ by taking an overall consideration of the BLEU and perplexity, because these two metrics evaluate our model from different perspectives. The experimental results indicate that the outcomes of BLEU and perplexity concerning λ are not consistently aligned. Specifically, as depicted in the top panels of Figure 4, BLEU attains the optimal value when $\lambda = 1.0$, whereas perplexity generally favors a smaller λ, as demonstrated in the bottom panels of Figure 4. Given our goal of training a model for generating scene-specific text, BLEU is more closely aligned with our objective than perplexity. Consequently, we select $\lambda = 1.0$ as the default value.

Comparisons with Other `softmax` Functions   In this experiment, we compare the proposed `graphmax` with the original `softmax`, `sparsemax` (Martins & Astudillo, 2016), and kernel Bayesian `softmax` (kerBayesian `softmax`) (Miao et al., 2019) functions on food and e-commerce review generation. As shown in Figure 5, the `graphmax` achieves the best performances on both the BLEU and perplexity metrics.

Visualization of x   We visualize the vector calculated by the proposed `graphmax` and compare it with the corresponding results of the traditional `softmax`. In the context "*I was so disappointed by this, and at this stage I was going to* $\cdots$", we visualize the top 10 next words respectively recommended by GPT2+`softmax` and GPT2+`graphmax`. As shown in Figure 6, the recommended lists of words by the two methods are slightly different. The next word of the aforementioned context predicted by `softmax` is "*give*", while that predicted by `graphmax` is "*be*".
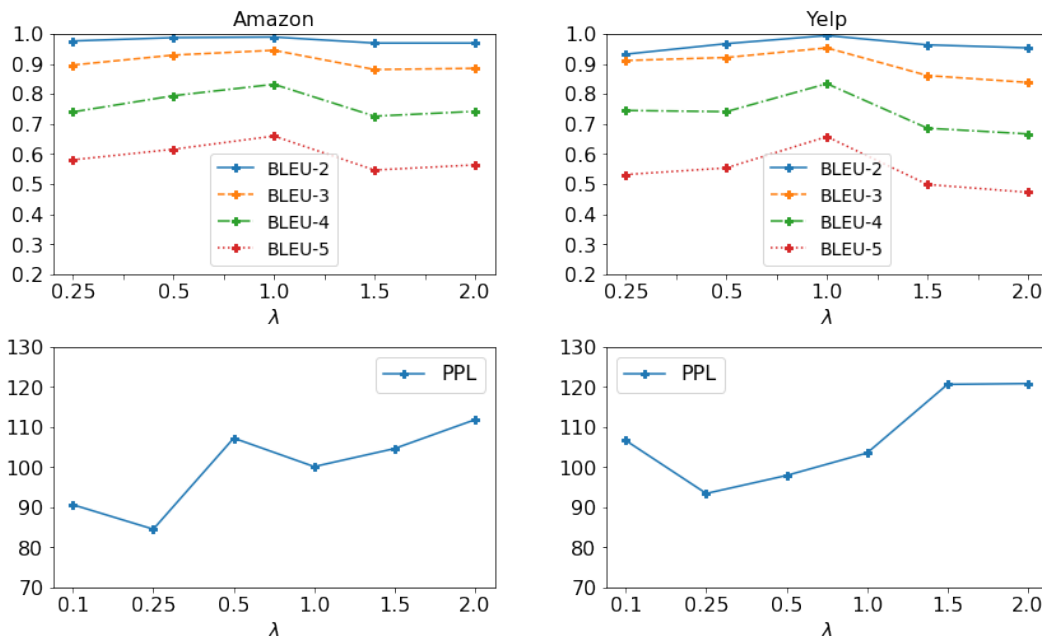
Figure 4: Searching λ using GPT-2 with the `graphmax` approach, considering both BLEU and perplexity as evaluation metrics. The results are presented in two panels: the upper panel shows the evaluation from a BLEU perspective, while the bottom panel presents the evaluation based on perplexity. The local graph used for parameter searching is constructed from the Amazon corpus in the left panels and the Yelp corpus in the right panels.

5.2 Machine Translation

To evaluate the performance of `graphmax` in machine translation, we conduct both monolingual and multilingual machine translations. We set the parameter $\lambda = 1.0$ in the following experiments.

Monolingual Machine Translation   We use BART (Lewis et al., 2020) as a base model and link it to the proposed `graphmax`. We construct a local style graph with the target language of the WMT'16 corpus, and then plug the local word preference graph into BART to improve machine translation. Table 6 shows translation results on both Romanian → English and English → Romanian directions.

Multilingual Machine Translation   As a baseline, we utilize mBART (Liu et al., 2020), a multilingual denoising pre-trained model for multilingual translation tasks. We incorporate `graphmax` into mBART and evaluate the performance on the WMT'21 corpus (Tran et al., 2021), which comprises translations from English to Czech (Cz), German (Ge), Hausa (Ha), Icelandic (Ic), Japanese (Ja), Russian (Ru), and Chinese (Ch), as shown in Table 7. The upper panel of the table presents the BLEU scores for translating English into seven languages, while the lower panel displays the inverse translations. Our results demonstrate
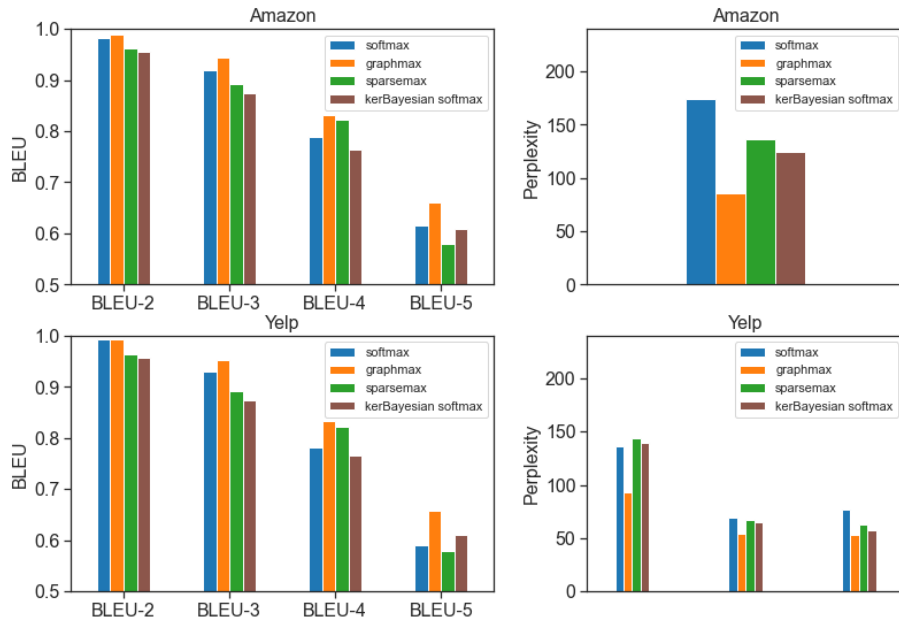
Figure 5: Comparisons of `graphmax` with `softmax`, `sparsemax`, and kernel Bayesian (kerBayesian) `softmax` on review generation.
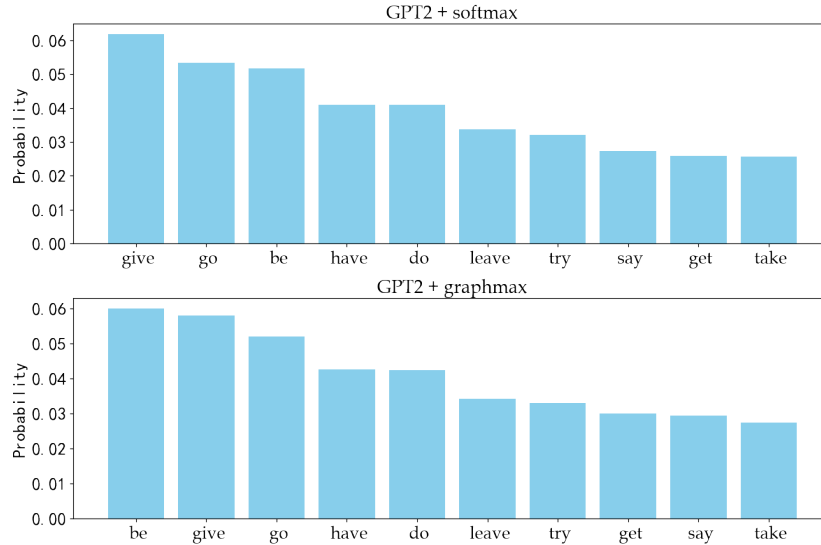

Figure 6: Visualization of the top 10 words respectively recommended by GPT2+`softmax` and GPT2+`graphmax` in the context "*I was so disappointed by this, and at this stage I was going to* ··· ".

Methods	Romanian→English	English→Romanian
BART	0.363	0.325
BART+`graphmax`	**0.372**	**0.335**

Table 6: The BLEU scores of BART and BART+`graphmax` on monolingual machine translation.

Methods	En→Cz	En→Ge	En→Ha	En→Ic	En→Ja	En→Ru	En→Ch
mBART	0.312	0.387	0.212	0.274	0.234	0.245	0.409
mBART+`graphmax`	**0.334**	**0.396**	**0.219**	**0.290**	**0.257**	**0.258**	**0.421**
	Cz→En	Ge→En	Ha→En	Ic→En	Ja→En	Ru→En	Ch→En
mBART	0.276	0.361	0.252	0.334	**0.222**	0.371	0.276
mBART+`graphmax`	**0.283**	**0.372**	**0.273**	**0.345**	0.220	**0.378**	**0.299**

Table 7: The BLEU scores of mBART and mBART+`graphmax` on multilingual machine translation.

that mBART+`graphmax` surpasses the performance of the baseline model in most translation tasks.

5.3 The n-gram Mixture Model

Section 2.4 (e.g., Equation (3)) and Section 5 elaborate on that `graphmax` with a first-order concurrent graph $\tilde{\mathbf{A}}$ acts as a 2-gram model. We can extend it to a general n-gram model by replacing $\tilde{\mathbf{A}}$ with a high-order adjacent matrix $\tilde{\mathbf{A}}^n$. Moreover, instead of using only $\tilde{\mathbf{A}}$, we can employ a mixture model consisting of $\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2 + \cdots + \tilde{\mathbf{A}}^n$. This mixed adjacency matrix integrates more useful knowledge into `graphmax`. The results in Table 8 indicate that an n-gram mixture model achieves better performance in most of the experiments involving text generation and machine translation. This further corroborates the effectiveness of incorporating higher-order dependencies in the form of $\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2$ with `graphmax`.

5.4 Human Evaluation

To evaluate the fluency of the generated text, we further conduct a human evaluation based on the work of Ahn, Morbini, and Gordon (2016). Participants are presented with a set of food and restaurant reviews generated by GPT-2 (baseline) and our proposed model (GPT-2+`graphmax`). They are asked to rate the quality of the text based on four perspectives: (1) text style, e.g., a human participant is asked whether the food review generated by the machine is in the social media slang style or in a formal style; (2) word choice; (3) word order variation; (4) story-line, which is a complete narrative logic of the generated text. In particular, participants read the reviews randomly selected from the baseline and the

Tasks	Metrics	{2,3}-gram	{2,3,4,5}-gram
	BLEU-2	0.989	0.992
	BLEU-3	0.945	0.965
Text generation	BLEU-4	0.832	0.875
	BLEU-5	0.660	0.652
	Perplexity	84.5	85.3
Machine translation	BLEU	0.372	0.379

Table 8: Comparisons of the $\{2,3\}$-gram model (with $\tilde{\mathbf{A}}$) and the $\{2,3,4,5\}$-gram mixture model (with $\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2$) on review generation with Amazon and machine translation on WMT'16.

	Methods	
Evaluation Category	GPT-2	GPT-2+`graphmax`
Text style	3.78 (2.56)	3.94 (1.39)
Word choice	4.43 (1.89)	4.31 (1.43)
Word order variation	4.12 (2.71)	4.67 (1.67)
Story-line	2.30 (2.45)	3.23 (1.41)
Average	3.66	4.04

Table 9: Results of human evaluation on text generation (standard deviations in brackets).

proposed model, and then rate the text from the four perspectives on a 5-point Likert scale: from 1 (worst) to 5 (best).

We randomly select 50 food and restaurant reviews generated by GPT-2 and another 50 reviews by the proposed model. The 100 reviews are shuffled and five participants (students) are invited to read and evaluate them independently. Moreover, we distribute the evaluation task to a wider range of participants through a questionnaire website[3]. A total of 27 participants complete the evaluation, including teachers, older adults, and secondary school students. The 100 mixed reviews are shuffled and presented for rating to all 32 participants, resulting in 426 valid ratings. The overall rating results for the four evaluation categories, as well as their averages, are shown in Table 9. Our proposed model (GPT-2+`graphmax`) performs better in controlling the text style and story-line coherence compared with the baseline. Overall, our method generates reviews that are more in line with the social media style and have a more coherent narrative logic.

---

3. https://www.wjx.cn/

6. Conclusion

Motivated by the word concurrent relationships which describe the human preferences of expression, we construct a weighted directed word concurrent graph for scene-specific text generation. The nodes of the graph are the words that appear in the corpus, the direction of edges is determined by the 2-grams of the corpus, and the corresponding weights can be calculated based on the frequency of the 2-grams. To incorporate the concurrent information into the process of text generation, we propose a concurrent graph regularized `softmax` function, called `graphmax`, for the next word prediction. The proposed `graphmax` is an n-gram mixture model that can be readily plugged into a large-scale pre-trained LM. We apply `graphmax` to scene-specific review generation and machine translation, and the results from both tasks demonstrate that our method can improve the performances than existing methods using `softmax`. This suggests that due to its simplicity and versatility `graphmax` should be used broadly in LMs.

Acknowledgements

Appendix A. Proofs

A.1 Equation (2) in `Softmax`

Proof. The augmented Lagrangian function of the optimization problem in Equation (1) is

$$L(\mathbf{x}, \lambda) = -\langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{x}, \log \mathbf{x} \rangle + \lambda(\mathbf{1}^\top \mathbf{x} - 1).$$

The first-order conditions yield

$$\frac{\partial L}{\partial \mathbf{x}} = -\mathbf{z} + \log \mathbf{x} + (1 + \lambda)\mathbf{1} = 0, \tag{9}$$

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{N} x_i - 1 = 0. \tag{10}$$

According to Equation (9), we have

$$\mathbf{x} = e^{\mathbf{z} - (1+\lambda)\mathbf{1}},$$

and its i-th component is

$$x_i = e^{z_i} \cdot e^{-(1+\lambda)} \tag{11}$$

Plugging Equation (11) into Equation (10), we have

$$\sum_{i=1}^{N} x_i = e^{-(1+\lambda)} \sum_{i=1}^{N} e^{z_i} = 1,$$

and thus $\sum_{i=1}^{N} e^{z_i} = e^{(1+\lambda)}$. By substituting it to Equation (11), we have $x_i = \frac{e^{z_i}}{\sum_{i=1}^{N} e^{z_i}}$, which completes the proof. $\square$

A.2 Proof of Proposition 2

Proof. For $a_i < a_j$, the first inequality is $\mathtt{graphmax}_j(\mathbf{a}) - \mathtt{graphmax}_i(\mathbf{a}) \geq 0$. A proof of contradiction can be easily constructed. Suppose that for $a_i < a_j$ we have $\mathtt{graphmax}_j(\mathbf{a}) - \mathtt{graphmax}_i(\mathbf{a}) < 0$. From the definition of the projection operator $\Pi_\Delta(\mathbf{a}) = \arg\min_{\mathbf{x}\in\Delta} \|\mathbf{x} - \mathbf{a}\|_2$ in Section 3.2, we have

$$\|\mathbf{x} - \mathbf{a}\|_2 \geq \|\mathtt{graphmax}(\mathbf{a}) - \mathbf{a}\|_2,$$

for $\mathbf{x} \in \Delta^{N-1}$. This leads to a contradiction,

$$\mathbf{a} = (a_1, a_2, \ldots, a_i, \ldots, a_j, \ldots)$$
$$\mathtt{graphmax}(\mathbf{a}) = (a_1, a_2, \ldots, a_j, \ldots, a_i, \ldots).$$

Concretely,

$$\|\mathbf{x} - \mathbf{a}\|_2 - \|\mathtt{graphmax}(\mathbf{a}) - \mathbf{a}\|_2$$
$$= \sum_{i=1}^N (x_i - a_i)^2 - \sum_{i=1}^N (\mathtt{graphmax}_i(\mathbf{a}) - a_i)^2$$
$$= (\mathtt{graphmax}_j(\mathbf{a}) - a_i)^2 + (\mathtt{graphmax}_i(\mathbf{a}) - a_j)^2$$
$$- (\mathtt{graphmax}_i(\mathbf{a}) - a_i)^2 - (\mathtt{graphmax}_j(\mathbf{a}) - a_j)^2$$
$$= 2(\mathtt{graphmax}_i(\mathbf{a}) - \mathtt{graphmax}_j(\mathbf{a}))(a_i - a_j) < 0$$

where $\mathtt{graphmax}_i(\mathbf{a})$ denotes the i-th component of the vector $\mathtt{graphmax}(\mathbf{a})$.

The second inequality $\mathtt{graphmax}_j(\mathbf{a}) - \mathtt{graphmax}_i(\mathbf{a}) \leq a_j - a_i$ contains three cases that can be proved one by one as follows.

For case 1,

$$\begin{cases} a_i + \gamma \geq 0, \\ a_j + \gamma \geq 0, \end{cases}$$

according to the definition of $\mathtt{graphmax}$ in Equation (7), we have

$$\mathtt{graphmax}_j(\mathbf{a}) - \mathtt{graphmax}_i(\mathbf{a}) = a_j - a_i < 0.$$

For case 2,

$$\begin{cases} a_i + \gamma \leq 0, \\ a_j + \gamma \geq 0, \end{cases}$$

we have

$$\mathtt{graphmax}_j(\mathbf{a}) - \mathtt{graphmax}_i(\mathbf{a}) = a_j + \gamma \leq a_j - a_i < 0.$$

Finally, for case 3, we have

$$\begin{cases} a_i + \gamma < 0, \\ a_j + \gamma < 0, \end{cases}$$

and this leads to

$$\mathtt{graphmax}_j(\mathbf{a}) - \mathtt{graphmax}_i(\mathbf{a}) = 0 < a_j - a_i.$$

This completes the proof. □

References

Ahmed, H. B., Dare, D., & Boudraa, A.-O. (2017). Graph signals classification using total variation and graph energy informations. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 667–671. IEEE.

Ahn, E., Morbini, F., & Gordon, A. (2016). Improving fluency in narrative text generation with grammatical transformations and probabilistic parsing. In *Proceedings of the 9th International Natural Language Generation conference*, pp. 70–73.

Anderson, P., Fernando, B., Johnson, M., & Gould, S. (2017). Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Névéol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., & Zampieri, M. (2016). Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 131–198, Berlin, Germany. Association for Computational Linguistics.

Chambolle, A., Caselles, V., Cremers, D., Novaga, M., & Pock, T. (2010). An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, *9*(263-340), 227.

Chen, S., Sandryhaila, A., Lederman, G., Wang, Z., Moura, J. M., Rizzo, P., Bielak, J., Garrett, J. H., & Kovačevic, J. (2014). Signal inpainting on graphs via total variation minimization. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8267–8271. IEEE.

Chen, S., Sandryhaila, A., Moura, J. M., & Kovačević, J. (2015). Signal recovery on graphs: Variation minimization. *IEEE Transactions on Signal Processing*, *63*(17), 4609–4624.

Chen, Z., Xu, J., Liao, M., Xue, T., & He, K. (2022). Two-phase multi-document event summarization on core event graphs. *J. Artif. Intell. Res.*, *74*, 1037–1057.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., & Liu, R. (2020). Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Duchi, J., Shalev-Shwartz, S., Singer, Y., & Chandra, T. (2008). Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279.

Ficler, J., & Goldberg, Y. (2017). Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pp. 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Gao, B., & Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*.

Jain, P., & Kar, P. (2017). Non-convex optimization for machine learning. *Foundations and Trends in Machine Learning*, *10*(3-4), 142–363.

Jia, F., Liu, J., & Tai, X.-C. (2021). A regularized convolutional neural network for semantic image segmentation. *Analysis and Applications*, *19*(01), 147–165.

Kann, K., Rothe, S., & Filippova, K. (2018). Sentence-level fluency evaluation: References help, but can be spared!. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 313–323.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation..

Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., & Hajishirzi, H. (2019). Text generation from knowledge graphs with graph transformers.. In *NAACL-HLT (1)*, pp. 2284–2293.

Koto, F., Lau, J. H., & Baldwin, T. (2020). Ffci: A framework for interpretable automatic evaluation of summarization. *J. Artif. Intell. Res.*, *73*.

Lellmann, J., Kappes, J., Yuan, J., Becker, F., & Schnörr, C. (2009). Convex multi-class image labeling by simplex-constrained total variation. In *International conference on scale space and variational methods in computer vision*, pp. 150–162. Springer.

Lellmann, J., & Schnörr, C. (2011). Continuous multiclass labeling approaches and algorithms. *SIAM Journal on Imaging Sciences*, *4*(4), 1049–1096.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online. Association for Computational Linguistics.

Liu, B., Xu, Z., Wu, S., & Wang, F. (2016). Manifold regularized matrix completion for multilabel classification. *Pattern Recognition Letters*, *80*, 58–63.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, *8*, 726–742.

Lv, S., Wei, L., Zhang, Q., Liu, B., & Xu, Z. (2022). Improved inference for imputation-based semisupervised learning under misspecified setting. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(11), 6346–6359.

Martins, A., & Astudillo, R. (2016). From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pp. 1614–1623.

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172. ACM.

Miao, N., Zhou, H., Zhao, C., Shi, W., & Li, L. (2019). Kernelized bayesian softmax for text generation. In *Advances in Neural Information Processing Systems*, pp. 12508–12518.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Raguet, H., & Landrieu, L. (2018). Cut-pursuit algorithm for regularizing nonsmooth functionals with graph total variation. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 4247–4256. PMLR.

Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, *60*(1-4), 259–268.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer.

Shi, L., Zhang, Y., Cheng, J., & Lu, H. (2019). Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1017–1024.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tran, C., Bhosale, S., Cross, J., Koehn, P., Edunov, S., & Fan, A. (2021). Facebook ai's wmt21 news translation task submission. In *Proc. of WMT*.

Vougiouklis, P., Maddalena, E., Hare, J. S., & Simperl, E. P. B. (2020). Point at the triple: Generation of text summaries from knowledge base triples (extended abstract). *J. Artif. Intell. Res.*, *69*, 1–31.

Yang, Z., Dai, Z., Salakhutdinov, R., & Cohen, W. W. (2018). Breaking the softmax bottleneck: A high-rank rnn language model. In *ICLR 2018 : International Conference on Learning Representations 2018*.

Yu, D., Yu, Z., & Sagae, K. (2021). Attribute alignment: Controlling text generation from pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2251–2268. Association for Computational Linguistics.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S.,

Sridhar, A., Wang, T., & Zettlemoyer, L. (2022). Opt: Open pre-trained transformer language models..

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pp. 649–657.