Coupled generation*

Ben Dai[1], Xiaotong Shen[1], and Wing Wong[2]

Abstract

Instance generation creates representative examples to interpret a learning model, as in regression and classification. For example, representative sentences of a topic of interest describe the topic specifically for sentence categorization. In such a situation, a large number of unlabeled observations may be available in addition to labeled data, for example, many unclassified text corpora (unlabeled instances) are available with only a few classified sentences (labeled instances). In this article, we introduce a novel generative method, called a coupled generator, producing instances given a specific learning outcome, based on indirect and direct generators. The indirect generator uses the inverse principle to yield the corresponding inverse probability, enabling to generate instances by leveraging an unlabeled data. The direct generator learns the distribution of an instance given its learning outcome. Then, the coupled generator seeks the best one from the indirect and direct generators, which is designed to enjoy the benefits of both and deliver higher generation accuracy. For sentence generation given a topic, we develop an embedding-based regression/classification in conjuncture with an unconditional recurrent neural network for the indirect generator, whereas a conditional recurrent neural network is natural for the corresponding direct generator. Moreover, we derive finite-sample generation error bounds for the indirect and direct generators to reveal the generative aspects of both methods thus explaining the benefits of the coupled generator. Finally, we apply the proposed methods to a real benchmark of abstract classification and demonstrate that the coupled generator composes reasonably good sentences from a dictionary to describe a specific topic of interest.

Keywords: Classification, Natural language processing, Numerical embeddings, Semisupervised generation, Unstructured data.

1 Introduction

Generating an essay or a text for given structured information is an important Artificial Intelligence (AI) problem, which automatically imitates a certain style of writing. Whereas solving this AI problem is rather challenging, we tackle its simpler version in this article,

*[1] School of Statistics, University of Minnesota, Minneapolis, MN 55455; [2] Department of Statistics and Biomedical Data Science, Stanford University, CA 94305 helpful comments and suggestions. Research supported in part by NSF grants DMS-1712564, DMS-1721216, DMS-1952539, DMS-1952386, and NIH grants 1R01GM126002, R01HL105397, R01AG065636.

which we call instance (example) generation, that is, generation of representative instances given a specific outcome to describe and interpret the corresponding learning model, for instance, classification and regression.

The use of black-box predictive models such as deep neural networks has delivered a high empirical learning accuracy in many real-life applications [14, 15]. Yet, it is difficult to make a sense of such a learning model. From the generative perspective, instance generation can describe the relationship between an instance and an outcome retrospectively. Its applications include a topic description of sentence categorization, abstractive text summarization [12], and image captioning [25], where generated sentences render descriptive examples of topics, texts, and images. In such a situation, sentence generation allows us to compose a novel essay and image captioning when the structured information is supplied. For example, the UCI abstract categorization benchmark[1] consists of sentences from abstracts of articles, which are labeled with one of five topic categories. The goal here is learning a sentence generation mechanism to compose a novel abstract given a specific topic, in which the generation performance is measured by the cross-entropy error based on a test sample.

In the literature, instance generation, despite its vast important applications in AI, remains largely unexplored, although some approaches have been suggested for sentence generation. For example, a computational linguistics approach represents words/phrases as trees to model linguistic dependencies [20], a learning approach uses a large text corpus to learn a sentence's structure without any access to linguistic annotation [5]. In [26], a sentence generating model is proposed to produce a document by sampling the latent topic of a sentence and then words of the sentence using a recurrent neural network (RNN). In [37, 17], image captioning links the image content to a language model through an interplay between a convolution neural network (CNN) and an RNN. Yet, there is a paucity of works on instance generation given structured information, and incorporating both labeled and unlabeled data.

[1] https://archive.ics.uci.edu/ml/datasets/Sentence+Classification

One of the primary characteristics of topic-instance data is that the amount of unlabeled data may be significantly larger than that of labeled data. For example, in sentence generation, uncategorized sentences are about ten times more than categorized ones. This is in a parallel situation of semisupervised learning with a different focus on leveraging unlabeled data to enhance the predictive accuracy of supervised learning [42, 18], which is in contrast to our generation objective given a learning outcome.

Our main contribution lies in the development of a new semisupervised generation framework for producing instances given an outcome. On this ground, we propose three generative methods–indirect, direct, and coupled generators. The indirect generator uses the principle of inverse learning to estimate the conditional probability distribution of an outcome given an instance, enabling to leverage unlabeled data, if available. On the other hand, the direct generator estimates the corresponding conditional probability of an instance given an outcome in a supervised manner. Then, the coupled generator is designed to enjoy the benefits of both generations. The proposed generators are illustrated in sentence generation, where we generate a sentence through sequential next-word-prediction. Specifically, we develop regularized embedding-based regression/classification in conjuncture with an unconditional RNN for the indirect generator, whereas we use a conditional RNN for the direct generator.

To shed light on the generative performance of the three generators, we derive finite-sample generation error bounds for each method. Interestingly, the generation error of the indirect generator is governed by the complexity of the parameter space of the conditional densities of an outcome given an instance and that of marginal densities. Similarly, that of the direct generator is determined by the conditional densities of an instance given an outcome. As a result, the indirect and direct generators have their own advantages with respect to generation with the unlabeled data is large, and importantly the coupled generator enjoys the benefits of both in terms of generation accuracy. This, together with a real benchmark of sentence categorization, demonstrates the utility of the coupled generation for composing

reasonably good sentences to describe a specific topic. Numerically, the proposed method outperforms a separate RNN method and the indirect generator can leverage additional unlabeled data for further enhancing the performance.

This paper is organized as follows. Section 2 introduces the framework of coupled generation based on indirect and direct generations. Section 3 develops a theory of the generation performance of the proposed methods. Section 4 is devoted to the development of a novel sentence generative method given a topic of interest through sequential next-word prediction. Section 5 investigates the operating characteristics of the coupled generator and compares it with the direct and indirect generators as well as one competitor. The Appendix contains technical proofs.

2 Methods

Consider a generative model in which the goal is to generate an instance $\boldsymbol{X}$ given an outcome $\boldsymbol{Y}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent instance and response variables, which can be numerical or unstructured such as texts and documents that cannot be expressed in a predefined manner. In this article, we focus on instance generation under a generative model, based on the conditional distribution $p_{\boldsymbol{X}|\boldsymbol{Y}}$ of $\boldsymbol{X}$ given an outcome of $\boldsymbol{Y}$. As an example, in sentence generation [26], instance generation produces representative examples of $\boldsymbol{X}$ given a specific topic of $\boldsymbol{Y}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent a sentence and its associated topic.

For instance generation, a labeled training sample $(\boldsymbol{x}^i, \boldsymbol{y}^i)_{i=1}^n$ is available as well as an instance-only sample $(\boldsymbol{x}^j)_{j=1}^{\tilde{n}}$, whose sample size $\tilde{n}$ may greatly exceed or smaller than the sample size n. In our context, we leverage the unlabeled sample to enhance the generative accuracy of instance generation.

Indirect generator. An indirect generator produces instances using an estimate of $p_{\boldsymbol{X}|\boldsymbol{Y}}$ through the inverse relation (1): an estimate of $p_{\boldsymbol{Y}|\boldsymbol{X}}$ based on $(\boldsymbol{x}^i, \boldsymbol{y}^i)_{i=1}^n$ in (2) and the

marginal density $p_{\boldsymbol{X}}$ based on combined data $(\boldsymbol{x}^i)_{i=1}^n$ and $(\boldsymbol{x}^j)_{j=1}^{\tilde{n}}$ in (3). That is,

$$\text{Indirect:} \quad \widehat{p}^{\,b}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{\widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}_{\boldsymbol{X}}(\boldsymbol{x})}{\int_{\boldsymbol{x}\in\mathcal{X}} \widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}}, \tag{1}$$

$$\widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}} = \underset{p_{\boldsymbol{Y}|\boldsymbol{X}}\in\mathcal{F}_b}{\operatorname{argmin}} -n^{-1}\sum_{i=1}^{n} \log\left(p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}^i|\boldsymbol{x}^i)\right) + \lambda_b J_b(p_{\boldsymbol{Y}|\boldsymbol{X}}), \tag{2}$$

$$\widehat{p}_{\boldsymbol{X}} = \underset{p_{\boldsymbol{X}}\in\mathcal{F}_m}{\operatorname{argmin}} -(n+\tilde{n})^{-1}\Big(\sum_{i=1}^{n} \log\left(p_{\boldsymbol{X}}(\boldsymbol{x}^i)\right) + \sum_{j=1}^{\tilde{n}} \log\left(p_{\boldsymbol{X}}(\boldsymbol{x}^j)\right)\Big) + \lambda_m J_m(p_{\boldsymbol{X}}), \tag{3}$$

where $\widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}}$ and $\widehat{p}_{\boldsymbol{X}}$ in (1) are regularized maximum likelihood estimates of $p_{\boldsymbol{Y}|\boldsymbol{X}}$ and $p_{\boldsymbol{X}}$, J_b and J_m are regularizers, for example, L_1- or L_2-regularization in a neural network model, $\lambda_b \geq 0$ and $\lambda_m \geq 0$ are tuning parameters controlling the weights of regularization, and $\mathcal{F}_b$ in (2) and $\mathcal{F}_m$ in (3) are parameter spaces of $p_{\boldsymbol{Y}|\boldsymbol{X}}$ and $p_{\boldsymbol{X}}$, respectively. Note that $\int_{\boldsymbol{x}\in\mathcal{X}} \widehat{p}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})\widehat{p}_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}$ in (1) normalizes $\widehat{p}^{\,b}_{\boldsymbol{X}|\boldsymbol{Y}}$ to become a probability density, although normalization is unnecessary when only some aspects of the distribution such as the modes or percentiles are of concern, as opposed to the distribution itself. Importantly, the indirect generator leverages instance-only (unlabeled) data $(\boldsymbol{x}^j)_{j=1}^{\tilde{n}}$, but any potential bias in estimation of $p_{\boldsymbol{X}}$ based on $(\boldsymbol{x}^j)_{j=1}^{\tilde{n}}$ could translate into that of $p_{\boldsymbol{Y}|\boldsymbol{X}}$.

Direct generator. A direct generator uses $p_{\boldsymbol{X}|\boldsymbol{Y}}$ to generate instances, estimated by minimizing the negative regularized conditional likelihood of $\boldsymbol{X}$ given $\boldsymbol{Y}$ based on $(\boldsymbol{x}^i, \boldsymbol{y}^i)_{i=1}^n$:

$$\text{Direct:} \quad \widehat{p}^{\,\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = \widehat{p}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}),$$

$$\widehat{p}_{\boldsymbol{X}|\boldsymbol{Y}} = \underset{p_{\boldsymbol{X}|\boldsymbol{Y}}\in\mathcal{F}_{\mathrm{f}}}{\operatorname{argmin}} -n^{-1}\sum_{i=1}^{n} \log\left(p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}^i|\boldsymbol{y}^i)\right) + \lambda_{\mathrm{f}} J_{\mathrm{f}}(p_{\boldsymbol{X}|\boldsymbol{Y}}), \tag{4}$$

where $\mathcal{F}_{\mathrm{f}}$ is a parameter space of $p_{\boldsymbol{X}|\boldsymbol{Y}}$, J_{f} is a regularizer, and $\lambda_{\mathrm{f}} \geq 0$ is a tuning parameter controlling the weight of regularization.

It appears that (4) can be extended to leverage additional unlabeled data through the conditional likelihood of $p_{\boldsymbol{X}|\boldsymbol{Y}}$ and a mixture relation $\int_{\boldsymbol{y}} p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})p_{\boldsymbol{Y}}(\boldsymbol{y})d\boldsymbol{y} = p_{\boldsymbol{X}}(\boldsymbol{x})$. Unfor-

tunately, however, the mixture approach may suffer from an asymptotic bias when additional unlabeled data is included, thus degrading the estimation performance of $p_{\boldsymbol{X}|\boldsymbol{Y}}$ [8, 9, 39]. This is because the aforementioned mixture relation may not hold when $\mathcal{F}_{\mathrm{f}}$ is misspecified, and moreover its impact could be minimal even it holds, especially when the support of $\boldsymbol{Y}$ is large. As suggested by the theorem in Section 4 [39], the supervised and semisupervised maximum likelihood estimates may converge to different values, and thus more unlabeled data produces a larger estimation bias as measured by the Kullback-Leibler divergence, when the model is misspecified in that $p^0_{\boldsymbol{X}}$ does not belong to the parameter space $\mathcal{F}_{\mathrm{f}} = \{p_{\boldsymbol{X}}(\boldsymbol{x}) = \int_{\boldsymbol{y}} p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})p_{\boldsymbol{Y}}(\boldsymbol{y})d\boldsymbol{y}; p_{\boldsymbol{X}|\boldsymbol{Y}} \in \mathcal{F}_{\mathrm{f}}\}$ or the mixture relation is not satisfied. Furthermore, as demonstrated by Figures 1 and 2 of [8] and Figure 4.1 of [7], empirical studies indicate that an EM algorithm based on both labeled and unlabeled data tends to degrade performance solely based on the labeled data when the size of labeled data exceeds 30 in SecStr dataset. As a result, $p_{\boldsymbol{X}|\boldsymbol{Y}}$ estimated from labeled data renders a better performance than that on labeled and unlabeled data.

In summary, how to leverage unlabeled data to enhance the generation performance remains an open question, which depends on model assumptions that may not be verifiable in practice. It is worth mentioning that (4) is a general formulation without assuming any specific assumption on how $p_{\boldsymbol{X}}$ is related to $\mathcal{F}_{\mathrm{f}}$. However, if such an assumption becomes available in practice, (4) can be generalized based on it to incorporate unlabeled data for improvement. At present, we shall not pursue this aspect as the indirect method can benefit from additional unlabeled data, as suggested by Theorem 1 in Section 3.

Coupled generator. The level of difficulty of estimating $\widehat{p}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}$ and that of $\widehat{p}^{b}_{\boldsymbol{X}|\boldsymbol{Y}}$ may differ, particularly when $p_{\boldsymbol{X}}$ can be well-estimated from both instance-only and unlabeled data. Depending on situations, the former may be more difficult than the latter, and vice versa. Some theoretical results for this aspect are illustrated in Section 4.5. Then we propose coupled generation by choosing, between the two, the one maximizing a predictive

log-likelihood, or minimizing a negative log-likelihood, such as (23) in the sentence generation example. In particular, a coupled generator is defined as,

$$\widehat{p}^{c}_{\boldsymbol{X}|\boldsymbol{Y}} = \begin{cases} \widehat{p}^{f}_{\boldsymbol{X}|\boldsymbol{Y}} & \text{if } \widehat{p}^{f}_{\boldsymbol{X}|\boldsymbol{Y}} \text{ has a higher log-likelihood value on a validation set than } \widehat{p}^{b}_{\boldsymbol{X}|\boldsymbol{Y}}, \\ \widehat{p}^{b}_{\boldsymbol{X}|\boldsymbol{Y}} & \text{otherwise.} \end{cases} \tag{5}$$

The probability density $\widehat{p}^{c}_{\boldsymbol{X}|\boldsymbol{Y}}$ gives the whole spectrum of values of $\boldsymbol{X}$ given $\boldsymbol{Y}$. First, we may generate representative instances using the mode of $p_{\boldsymbol{X}|\boldsymbol{Y}}$ to give one representation or sampling-based on $p_{\boldsymbol{X}|\boldsymbol{Y}}$ for multiple representations. Second, discriminative features $\boldsymbol{X}$ with respect to $\boldsymbol{Y}$ can be extracted by comparing $\widehat{p}^{c}_{\boldsymbol{X}|\boldsymbol{Y}}$ at different $\boldsymbol{Y}$-values retrospectively. For example, in classification with $Y = \pm 1$, a comparison of $\widehat{p}^{c}_{\boldsymbol{X}|Y=1}$ and $\widehat{p}^{c}_{\boldsymbol{X}|Y=-1}$ leads to discriminative features. This aspect will be further investigated elsewhere.

Coupled learning has its distinct characteristics although it appears remotely related to semisupervised variational auto-encoders [18] and inverse autoregressive flows [19]. In particular, [19] uses a generative model $p_{\boldsymbol{X}|\boldsymbol{Y}}$ and $p_{\boldsymbol{X}}$ to enhance a discriminative model $p_{\boldsymbol{Y}|\boldsymbol{X}}$ regarding the marginal distribution as a mixture of conditional distributions, whereas the proposed indirect generator integrates the unlabeled data to separately estimate the marginal distribution. Furthermore, [18] estimates the marginal density of $\boldsymbol{X}$ $p_{\boldsymbol{X}}$ via a chain of latent factors and inevitable transformations of autoregressive neural networks and connects blocks by invertible relations. Yet, the proposed method links two conditional densities by Bayes' law. Finally, the theoretical justification of [19] and [18] remains unknown.

3 Theory

This section develops a learning theory to investigate the generation errors of direct, indirect, and coupled generators. In particular, we derive finite-sample generation error bounds for estimators $\widehat{p}^{b}_{\boldsymbol{X}|\boldsymbol{Y}}$, $\widehat{p}^{f}_{\boldsymbol{X}|\boldsymbol{Y}}$, and $\widehat{p}^{c}_{\boldsymbol{X}|\boldsymbol{Y}}$ of (1), (4) and (5).

The generation error for generating $\boldsymbol{X}$ given $\boldsymbol{Y}$ is defined as the expected Hellinger-distance between two conditional densities $p_{\boldsymbol{X}|\boldsymbol{Y}}$ and $q_{\boldsymbol{X}|\boldsymbol{Y}}$ with respect to $\boldsymbol{Y}$:

$$d\big(p_{\boldsymbol{X}|\boldsymbol{Y}}, q_{\boldsymbol{X}|\boldsymbol{Y}}\big) = \big(\mathbb{E}_{\boldsymbol{Y}} h^2(p_{\boldsymbol{X}|\boldsymbol{Y}}, q_{\boldsymbol{X}|\boldsymbol{Y}})\big)^{1/2} \equiv \big(\mathbb{E}_{\boldsymbol{Y}} \int (p_{\boldsymbol{X}|\boldsymbol{Y}}^{1/2} - q_{\boldsymbol{X}|\boldsymbol{Y}}^{1/2})^2 d\mu\big)^{1/2},$$

where μ is the Lebesgue measure on $\boldsymbol{x}$, and $\mathbb{E}_{\boldsymbol{Y}}$ is the expectation with respect to $\boldsymbol{Y}$.

Three parameter spaces $\mathcal{F}_b$, $\mathcal{F}_m$, and $\mathcal{F}_{\mathrm{f}}$ are defined for estimating $p_{\boldsymbol{Y}|\boldsymbol{X}}$ in (2), $p_{\boldsymbol{X}}$ in (3), and $p_{\boldsymbol{X}|\boldsymbol{Y}}$ in (4), each of which is allowed to depend on the corresponding sample size. Then their regularized parameter spaces are given as follows: $\mathcal{F}_{b,k} = \{p_{\boldsymbol{Y}|\boldsymbol{X}} \in \mathcal{F}_b : J(p_{\boldsymbol{Y}|\boldsymbol{X}}) \le k\}$ for (2), $\mathcal{F}_{m,k} = \{p_{\boldsymbol{X}} \in \mathcal{F}_m : J(p_{\boldsymbol{X}}) \le k\}$ for (3), and $\mathcal{F}_{\mathrm{f},k} = \big\{p_{\boldsymbol{X}|\boldsymbol{Y}} \in \mathcal{F}_{\mathrm{f}} : J(p_{\boldsymbol{X}|\boldsymbol{Y}}) \le k\big\}$ for (4). On this ground, we define the metric entropy to measure their complexities to be used for our theory.

The u-bracketing metric entropy $H(u, \mathcal{F})$ of space $\mathcal{F}$ with respect to a distance D is defined as the logarithm of the cardinality of the u-bracketing of $\mathcal{F}$ of the smallest size. A u-bracketing of $\mathcal{F}$ is a finite set (of pairs of functions) $\{(p_j^L, p_j^U), j = 1, \cdots, N\}$ such that for any $p \in \mathcal{F}$, there is a j such that $p_j^L \le p \le p_j^U$ with $d(p_j^L, p_j^U) \le u$; $j = 1, \cdots, N$. Note that $d^2(p_{\boldsymbol{Y}|\boldsymbol{X}}, q_{\boldsymbol{Y}|\boldsymbol{X}}) = \mathbb{E}_{\boldsymbol{X}}\big(h^2(p_{\boldsymbol{Y}|\boldsymbol{X}}, q_{\boldsymbol{Y}|\boldsymbol{X}})\big)$, $h^2(p_{\boldsymbol{X}}, q_{\boldsymbol{X}})$, and $\mathbb{E}_{\boldsymbol{Y}} h^2(p_{\boldsymbol{X}|\boldsymbol{Y}}, q_{\boldsymbol{X}|\boldsymbol{Y}})$, respectively for $\mathcal{F}_{b,k}$, $\mathcal{F}_{m,k}$, and $\mathcal{F}_{\mathrm{f},k}$.

To quantify the degree of approximation of the true density $p^0_{\boldsymbol{Y}|\boldsymbol{X}}$ by $\mathcal{F}_b$, we introduce a distance $\rho_b(p^0_{\boldsymbol{Y}|\boldsymbol{X}}, p_{\boldsymbol{Y}|\boldsymbol{X}}) = \mathbb{E}_{\boldsymbol{X}}\mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}} g_\alpha(p^0_{\boldsymbol{Y}|\boldsymbol{X}}/p_{\boldsymbol{Y}|\boldsymbol{X}})$, where $g_\alpha(x) = \alpha^{-1}(x^\alpha - 1)$ for $\alpha \in (0, 1)$. As suggested in Section 4 of [38], this distance is stronger than the corresponding Hellinger distance. Similarly, $\rho_m(p^0_{\boldsymbol{X}}, p_{\boldsymbol{X}}) = \mathbb{E}_{\boldsymbol{X}} g_\alpha(p^0_{\boldsymbol{X}}/p_{\boldsymbol{X}})$ and $\rho_{\mathrm{f}}(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, p_{\boldsymbol{X}|\boldsymbol{Y}}) = \mathbb{E}_{\boldsymbol{X}}\mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}} g_\alpha(p^0_{\boldsymbol{X}|\boldsymbol{Y}}/p_{\boldsymbol{X}|\boldsymbol{Y}})$ are defined for approximating the true densities $p^0_{\boldsymbol{X}}$ and $p^0_{\boldsymbol{X}|\boldsymbol{Y}}$ by $\mathcal{F}_m$ and $\mathcal{F}_{\mathrm{f}}$, respectively.

Let $p^*_{\boldsymbol{Y}|\boldsymbol{X}} \in \mathcal{F}_b$ and $p^*_{\boldsymbol{X}} \in \mathcal{F}_m$ be two approximating points of $p^0_{\boldsymbol{Y}|\boldsymbol{X}}$ and $p^0_{\boldsymbol{X}}$ in that $\rho_b(p^0_{\boldsymbol{Y}|\boldsymbol{X}}, p^*_{\boldsymbol{Y}|\boldsymbol{X}}) \le \gamma_b$ and $\rho_m(p^0_{\boldsymbol{X}}, p^*_{\boldsymbol{X}}) \le \gamma_m$ for some sequences $\gamma_b \ge 0$ and $\gamma_m \ge 0$. Of course, $\gamma_b = 0$ when $p^0_{\boldsymbol{Y}|\boldsymbol{X}} \in \mathcal{F}_b$ and $\gamma_m = 0$ when $p^0_{\boldsymbol{X}} \in \mathcal{F}_m$.

Theorem 1 (Indirect generator)**.** *Suppose there exist some positive constants* c_1-c_6*, such that, for any* $\epsilon_b > 0$ *and* $\lambda_b \geq 0$*,*

$$\sup_{k\geq 1} \int_{2^{-8}L_k}^{2^{1/2}L_k^{1/2}} H^{1/2}(u/c_3, \mathcal{F}_{b,k})du/L_k \leq c_2 n^{1/2}, \quad L_k = c_1\epsilon_b^2 + \lambda_b(k-1), \tag{6}$$

and, for any $\epsilon_m > 0$ *and* $\lambda_m \geq 0$*,*

$$\sup_{k\geq 1} \int_{2^{-8}L_k}^{2^{1/2}L_k^{1/2}} H^{1/2}(u/c_6, \mathcal{F}_{m,k})du/L_k \leq c_5 (n+\tilde{n})^{1/2}, \quad L_k = c_4\epsilon_m^2 + \lambda_m(k-1), \tag{7}$$

then

$$P\big(d(\widehat{p}^b_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \geq 2(\eta_b + \eta_m)\big) \leq 8\exp\big(-c_7 n\eta_b^2\big) + 8\exp\big(-c_8(n+\tilde{n})\eta_m^2\big), \tag{8}$$
$$\eta_b = \max(\epsilon_b, \gamma_b^{1/2}), \quad \eta_m = \max(\epsilon_m, \gamma_m^{1/2}),$$

provided that $\lambda_b \max(J_b(p^*_{\boldsymbol{Y}|\boldsymbol{X}}), J_b(p^0_{\boldsymbol{Y}|\boldsymbol{X}}), 1) \leq c_9\eta_b^2$ *and* $\lambda_m \max(J_m(p^*_{\boldsymbol{X}}), J_m(p^0_{\boldsymbol{X}}), 1) \leq c_9\eta_m^2$*, and* c_7-c_9 *are some positive constants. Consequently,* $d(\widehat{p}^b_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) = O_p(\eta_b + \eta_m)$ *as* $n, \tilde{n} \to \infty$ *under under* $p^0_{\boldsymbol{X},\boldsymbol{Y}}$*.*

Theorem 1 indicates that the generation error of the indirect generator is governed by the estimation errors ϵ_b and ϵ_m from (2) and (3), and the approximation errors γ_b and γ_m, where ϵ_b and ϵ_m can be obtained by solving the entropy integral equations in (6) and (7). Moreover, optimal rates of convergence can be obtained through tuning of λ_b and λ_m. Note that η_m could be tuned as a smaller order of η_b when the size of unlabeled data greatly exceeds the size of labeled data. Then the generalization error of the indirect method is governed primarily by η_b. In other words,the indirect generator's performance is mainly determined by the estimation of $p_{\boldsymbol{Y}|\boldsymbol{X}}$.

For direct generator, let $p^*_{\boldsymbol{X}|\boldsymbol{Y}} \in \mathcal{F}_{\mathrm{f}}$ be an approximation of $p^0_{\boldsymbol{X}|\boldsymbol{Y}}$ in that $\rho_{\mathrm{f}}(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, p^*_{\boldsymbol{X}|\boldsymbol{Y}}) \leq \gamma_{\mathrm{f}}$

for some $\gamma_{\mathrm{f}} \geq 0$.

Theorem 2 (Direct generator). *Suppose there exist some positive constants c_{10}-c_{12}, such that, for any $\epsilon_{\mathrm{f}} > 0$, and $\lambda_{\mathrm{f}} \geq 0$,*

$$\sup_{k \geq 1} \int_{2^{-8} L_k}^{2^{1/2} L_k^{1/2}} H^{1/2}\big(u/c_{12}, \mathcal{F}_{\mathrm{f},k}\big) du / L_k \leq c_{11} n^{1/2}, \quad L_k = c_{10}\epsilon_{\mathrm{f}}^2 + \lambda_{\mathrm{f}}(k-1), \tag{9}$$

then

$$P\big(d(\widehat{p}^{f}_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \geq \eta_{\mathrm{f}}\big) \leq 8\exp(-c_{13} n \eta_{\mathrm{f}}^2), \quad \eta_{\mathrm{f}} = \max\left(\epsilon_{\mathrm{f}}, \gamma_{\mathrm{f}}^{1/2}\right), \tag{10}$$

provided that $\lambda_{\mathrm{f}} \max(J_{\mathrm{f}}(p^*_{\boldsymbol{X}|\boldsymbol{Y}}), J_{\mathrm{f}}(p^0_{\boldsymbol{X}|\boldsymbol{Y}}), 1) \leq c_9 \eta_{\mathrm{f}}^2$, and $c_{13} > 0$ is a constant. Consequently, $d(\widehat{p}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) = O_p(\eta_{\mathrm{f}})$ as $n \to \infty$ under $p^0_{\boldsymbol{X},\boldsymbol{Y}}$.

In contrast to the indirect generation, the generation error η_{f} of the direct generation could be much larger or smaller than that of the indirect generation η_b depending on the complexities of $\mathcal{F}_{\mathrm{f}}$, $\mathcal{F}_b$, and the corresponding approximation errors γ_{f} and γ_b, when $p_{\boldsymbol{X}}$ can be sufficiently well estimated. This suggests that either may outperform the other depending on the model assumptions.

Note that $\gamma_{\mathrm{f}}, \gamma_b$, and γ_m are the approximation errors of the approximation capabilities of function spaces $\mathcal{F}_{\mathrm{f}}, \mathcal{F}_b$ and $\mathcal{F}_m$ [35, 40]. In particular, when the function space is defined by a ReLU deep neural network to approximate a function in a Sobolev space, the approximation error is available and related to the scale of the neural network [40].

Theorem 3 says that the coupled generator performs no worse than the indirect and direct generators when (5) is used to select based on an independent cross-validation sample of size N.

Theorem 3 (Coupled generation). *Under $p^0_{\boldsymbol{X},\boldsymbol{Y}}$, as $N \to \infty$, the coupled generator defined in (5) satisfies $K(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, \widehat{p}^{c}_{\boldsymbol{X}|\boldsymbol{Y}}) \leq \min\big(K(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, \widehat{p}^{b}_{\boldsymbol{X}|\boldsymbol{Y}}), K(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, \widehat{p}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}})\big)$, where $K(p_{\boldsymbol{X}|\boldsymbol{Y}}, q_{\boldsymbol{X}|\boldsymbol{Y}})$ is the Kullback-Leibler divergence between $p_{\boldsymbol{X}|\boldsymbol{Y}}$ and $q_{\boldsymbol{X}|\boldsymbol{Y}}$.*

Remarks: In Theorem 3, if $K(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, p_{\boldsymbol{X}|\boldsymbol{Y}}) \leq c_{14}^2 d^2(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, p_{\boldsymbol{X}|\boldsymbol{Y}})$ for some constant $c_{14} > 0$, then $d(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, \widehat{p}^c_{\boldsymbol{X}|\boldsymbol{Y}}) \leq c_{15} \min\big(d(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, \widehat{p}^b_{\boldsymbol{X}|\boldsymbol{Y}}), d(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, \widehat{p}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}})\big)$, which occurs when the likelihood ratio is bounded.

4 Sentence generation given a topic

This section derives generative methods of sentence generation, which integrates the likelihood methods developed previously with language models to compose a sentence. As a result, a new sentence can be generated, which may not appear in training data; see Table 3 for an example.

A complete sentence is represented by a word vector $\boldsymbol{X}_{1:T} = (X_1, \cdots, X_T)'$, where X_t is the t-th word, T is a sentence-specific length, and $'$ denotes the transpose of a vector. For convenience, we write X_1 = "START" and X_{T+1} = "END" as the null words of the first and last words of a sentence, respectively. For example, X_1 ="START", X_2 = "Football", X_3 = "is", X_4 = "a", X_5 = "popular", X_6 = "sport", and X_7 = "END". Together with $\boldsymbol{X}_{1:T}$, its associated topic category $\boldsymbol{Y} = (Y_1, \cdots, Y_K)'$ is available, where $Y_j \in \{0, 1\}$ or $Y_j \in \mathbb{R}$. Finally, we construct a dictionary $\mathcal{D} = (w_1, \cdots, w_{|\mathcal{D}|})'$ to contain all composing words, that is $X_t \in \mathcal{D}$; $t = 1, \cdots, T$, with $|\mathcal{D}|$ denoting $\mathcal{D}$'s size.

For simplicity, we consider the case of a fixed T, where sentences of different lengths can be processed with a fixed length, as illustrated in Table 1. Sentence generation given a topic $\boldsymbol{Y}$ generates a sentence $\boldsymbol{X}_{1:T+1}$ using the conditional probability $P\big(\boldsymbol{X}_{1:T+1} = \boldsymbol{x}_{1:T+1} | \boldsymbol{Y} = \boldsymbol{y}\big)$. However, estimation of this probability at the sentence level is infeasible. Therefore, we decompose it at the word level by the probability chain rule:

$$\log\big(p\big(\boldsymbol{X}_{1:T+1} = \boldsymbol{x}_{1:T+1} | \boldsymbol{Y} = \boldsymbol{y}\big)\big) = \sum_{t=1}^{T} \log\big(p(X_{t+1} = x_{t+1} | \boldsymbol{X}_{1:t} = \boldsymbol{x}_{1:t}, \boldsymbol{Y} = \boldsymbol{y})\big). \tag{11}$$

This decomposition (11) permits sequential generation of a sentence through next-word-prediction given existing words by learning $p(X_{t+1} = x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y})$ from data; $t = 1, \cdots, T$.

Yet, estimation of $p(X_{t+1} = x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y})$ remains challenging for unstructured $\boldsymbol{X}_{1:t}$ because of a lack of observations in any conditioning event of $\boldsymbol{X}_{1:t}$ given $\boldsymbol{Y}$ even with large training data. Furthermore, it is difficult to utilize unlabeled data to estimate $p(X_{t+1} = x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y})$.

4.1 Indirect generator

In this context, we derive a version of (2) and that of (3) through (11) to estimate the inverse probabilities. Specifically, $p(x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y})$ can be written as

$$p(x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_{1:t+1})p(x_{t+1}|\boldsymbol{x}_{1:t})}{\sum_{x_{t+1}\in\mathcal{D}} p(\boldsymbol{y}|\boldsymbol{x}_{1:t}, x_{t+1})p(x_{t+1}|\boldsymbol{x}_{1:t})}; \tag{12}$$

for $t = 1, \cdots, T$. Then, we estimate the inverse probability $p(\boldsymbol{y}|\boldsymbol{x}_{1:t+1})$ based on labeled data $(\boldsymbol{x}^i_{1:t}, \boldsymbol{y}^i)_{i=1}^n$ and estimate $p(x_{t+1}|\boldsymbol{x}_{1:t})$ based on $(\boldsymbol{x}^i_{1:t})_{i=1}^n$ for $t = 1, \cdots, T^i$, and unlabeled data $(\boldsymbol{x}^j_{1:t})_{j=1}^{\tilde{n}}$ for $t = 1, \cdots, T^j$.

Estimation of $p(\boldsymbol{y}|\boldsymbol{x}_{1:t})$ may proceed with unstructured predictors $\boldsymbol{x}_{1:t}$. To proceed, we map a sentence $\boldsymbol{x}_{1:t}$ to a numerical vector $\mathcal{E}(\boldsymbol{x}_{1:t}) \in \mathbb{R}^p$, known as a numerical embedding of size p via a pre-trained embedding model such as Doc2Vec [23, 24] and BERT [11]. If a pre-trained embedding model is sufficient in that $p(\boldsymbol{y}|\boldsymbol{X}_{1:t} = \boldsymbol{x}_{1:t}) = p(\boldsymbol{y}|\mathcal{E}(\boldsymbol{X}_{1:t}) = \mathcal{E}(\boldsymbol{x}_{1:t}))$ [10], the numerical embedding $\mathcal{E}(\boldsymbol{x}_{1:t})$ captures word-to-word relations expressed in terms of co-occurrence of words, which may raise the level of predictability of unstructured predictors $\boldsymbol{X}_{1:t}$. Next, we model $p(\boldsymbol{y}|\boldsymbol{x}_{1:t})$ through $p\big(\boldsymbol{y}|\mathcal{E}(\boldsymbol{x}_{1:t})\big)$ when $\boldsymbol{Y} \in \{0, 1\}^K$ is categorical or

$\boldsymbol{Y} \in \mathbb{R}^K$ is continuous with an embedded label $\boldsymbol{Y}$:

$$p(\boldsymbol{y}|\boldsymbol{x}_{1:t}) = \begin{cases} \boldsymbol{y}'\boldsymbol{\sigma}\big(f\big(\mathcal{E}(\boldsymbol{x}_{1:t})\big)\big), & \text{if } \boldsymbol{y} \in \{0,1\}^K, \\ (2\pi)^{-K/2} \exp\big(-\frac{1}{2}\big\|\boldsymbol{y} - f\big(\mathcal{E}(\boldsymbol{x}_{1:t})\big)\big\|_2^2\big), & \text{if } \boldsymbol{y} \in \mathbb{R}^K, \end{cases} \tag{13}$$

where K is the dimension of $\boldsymbol{Y}$, $\boldsymbol{\sigma}(\cdot)$ is the softmax function [1], and f is a nonparametric classification or regression function forest [3] or linear function $f(\mathcal{E}(\boldsymbol{x}_{1:t})) = \boldsymbol{\theta}_b \mathcal{E}(\boldsymbol{x}_{1:t})$ with $\boldsymbol{\theta}_b \in \mathbb{R}^{K \times p}$. For illustration, we use a linear representation $f\big(\mathcal{E}(\boldsymbol{x}_{1:t})\big) = \boldsymbol{\theta}_b \mathcal{E}(\boldsymbol{x}_{1:t})$ in (13) sequentially. Now the cost function $L_b(\boldsymbol{\theta}_b)$ in (2) becomes

$$L_b(\boldsymbol{\theta}_b) = -\frac{1}{n}\sum_{i=1}^{n}(T^i)^{-1}\sum_{t=1}^{T^i}\log\big(p(\boldsymbol{y}^i|\mathcal{E}(\boldsymbol{x}^i_{1:t})\big) + \lambda_b J_b(f), \tag{14}$$

where $\lambda_b \geq 0$ is a tuning parameter and $J_b(f) \geq 0$ is a regularizer, for example, $J_b(f) = \|\boldsymbol{\theta}_b\|_F^2$ if $f\big(\mathcal{E}(\boldsymbol{x}_{1:t})\big) = \boldsymbol{\theta}_b \mathcal{E}(\boldsymbol{x}_{1:t})$, where $\|\cdot\|_F$ is the Frobenius-norm of a matrix.

Figure 1 about here

On the other hand, the next word probability is estimated by a RNN in a form of

$$p(x_{t+1}|\boldsymbol{x}_{1:t}) = \boldsymbol{o}_{[x_{t+1}]}(x_t, \boldsymbol{h}_t; \boldsymbol{\theta}_m), \text{ with } \boldsymbol{h}_t = \boldsymbol{h}(x_t, \boldsymbol{h}_{t-1}; \boldsymbol{\theta}_m), \boldsymbol{h}_0 = \boldsymbol{0}, \tag{15}$$

where $[x_{t+1}] = \{j : w_j = x_{t+1}\}$, $\boldsymbol{o}_j(x_t, \boldsymbol{h}_t, \boldsymbol{\theta}_m)$ is the probability of occurrence of the j-th word in $\mathcal{D}$, and $\boldsymbol{h}(x_t, \boldsymbol{h}_{t-1}, \boldsymbol{\theta}_m)$ is a hidden state function, such as a long short-term memory unit (LSTM) [16], a bidirectional unit [32], a gated recurrent unit (GRU) [6], and GPT2 [30], $\boldsymbol{\theta}_m$ is the parameter of a specific RNN model, for example, $\boldsymbol{\theta}_m = (\boldsymbol{W}_m^o, \boldsymbol{W}_m^x, \boldsymbol{W}_m^h)$ in a basic RNN,

$$\boldsymbol{o}(x_t, \boldsymbol{h}_t, \boldsymbol{\theta}_m) = \boldsymbol{\sigma}(\boldsymbol{W}_m^o \boldsymbol{h}_t), \ \boldsymbol{h}_t = \phi(\boldsymbol{W}_m^x \boldsymbol{1}_{[x_t]} + \boldsymbol{W}_m^h \boldsymbol{h}_{t-1}), \text{ and } \boldsymbol{h}_0 = \boldsymbol{0}, \tag{16}$$

where $\boldsymbol{\sigma}(\cdot)$, as defined before, is the softmax and ϕ is an activation function such as the ReLU function [1], $\boldsymbol{W}_m^o \in \mathbb{R}^{r_m \times |\mathcal{D}|}$, $\boldsymbol{W}_m^x \in \mathbb{R}^{|\mathcal{D}| \times r_m}$, and $\boldsymbol{W}_m^h \in \mathbb{R}^{r_m \times r_m}$, and r_m is the number of latent factors of the RNN. See Figure 1 for a display of the architecture of a basic RNN.

On the ground of (15), the cost function $L_m(\boldsymbol{\theta}_m)$ in (3) becomes

$$\begin{aligned} L_m(\boldsymbol{\theta}_m) = &-(n+\tilde{n})^{-1} \sum_{i=1}^{n} (T^i)^{-1} \sum_{t=1}^{T^i} \log\left(\boldsymbol{o}_{[x_{t+1}^i]}(x_t^i, \boldsymbol{h}_t^i, \boldsymbol{\theta}_m)\right) \\ &- (n+\tilde{n})^{-1} \sum_{j=1}^{\tilde{n}} (T^j)^{-1} \sum_{t=1}^{T^j} \log\left(\boldsymbol{o}_{[x_{t+1}^j]}(x_t^j, \boldsymbol{h}_t^j, \boldsymbol{\theta}_m)\right) + \lambda_m J_m(\boldsymbol{\theta}_m), \end{aligned} \tag{17}$$

where $\lambda_m \geq 0$ is a tuning parameter and $J_m(\boldsymbol{\theta}_m)$ is a regularizer regularizing the weights matrix and the activation layer [22].

Minimizing (14) and (17) yields estimators $\widehat{\boldsymbol{\theta}}_b$ and $\widehat{\boldsymbol{\theta}}_m$, respectively. Then the conditional probabilities are estimated as $\widehat{p}(\boldsymbol{y}|\boldsymbol{x}_{1:t+1}; \widehat{\boldsymbol{\theta}}_b) = \sigma\big(\widehat{\boldsymbol{\theta}}_b \mathcal{E}(\boldsymbol{x}_{1:t+1})\big)$ and $\widehat{p}(x_{t+1}|\boldsymbol{x}_{1:t}; \widehat{\boldsymbol{\theta}}_m) = \boldsymbol{o}_{[x_{t+1}]}(x_t, \boldsymbol{h}_{t-1}, \widehat{\boldsymbol{\theta}}_m)$. Plugging these estimates into (12), we obtain the estimated probability, and the process is summarized as,

$$\begin{aligned} &\widehat{p}^b(X_{t+1} = x|\boldsymbol{x}_{1:t}, \boldsymbol{y}) = \frac{\widehat{p}(\boldsymbol{y}|\boldsymbol{x}_{1:t}, x; \widehat{\boldsymbol{\theta}}_b)\widehat{p}(X_{t+1} = x|\boldsymbol{x}_{1:t}; \widehat{\boldsymbol{\theta}}_m)}{\sum_{x \in \mathcal{D}} \widehat{p}(\boldsymbol{y}|\boldsymbol{x}_{1:t}, x; \widehat{\boldsymbol{\theta}}_b)\widehat{p}(X_{t+1} = x|\boldsymbol{x}_{1:t}; \widehat{\boldsymbol{\theta}}_m)} \\ &\widehat{\boldsymbol{\theta}}_b = \underset{\boldsymbol{\theta}_b}{\operatorname{argmin}}\, L_b(\boldsymbol{\theta}_b), \qquad \widehat{\boldsymbol{\theta}}_m = \underset{\boldsymbol{\theta}_m}{\operatorname{argmin}}\, L_m(\boldsymbol{\theta}_m). \end{aligned} \tag{18}$$

Then, a sentence is sequentially generated as follows:

$$\widehat{x}_{t+1} = \underset{x \in \mathcal{D}}{\operatorname{argmax}}\, \widehat{p}^b(X_{t+1} = x|\boldsymbol{X}_{1:t} = \boldsymbol{x}_{1:t}, \boldsymbol{Y} = \boldsymbol{y}); \quad t = 1, \cdots, \widehat{T}. \tag{19}$$

This generation process begins with $\widehat{\boldsymbol{x}}_1$ = "START" or pre-specified t_0-words $\widehat{\boldsymbol{x}}_{1:t_0}$ and proceeds until $\widehat{\boldsymbol{x}}_{\widehat{T}}$ = "END" is reached, where $\widehat{T}$ is an index at termination. It is worth mentioning that the denominator in (18) normalizes the probability but may not need to be

computed when a maximizer of (18) is desired in (19).

4.2 Direct generator

The direct generation is inspired by a conditional RNN (C-RNN; [37, 17]) by estimating

$$p(x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y}) = \boldsymbol{o}_{[x_{t+1}]}(x_t, \boldsymbol{h}_{t-1}, \boldsymbol{y}, \boldsymbol{\theta}_{\mathrm{f}}), \text{ with } \boldsymbol{h}_t = \boldsymbol{h}(x_t, \boldsymbol{h}_{t-1}, \boldsymbol{\theta}_{\mathrm{f}}), \ \boldsymbol{h}_0 = \boldsymbol{h}_0(\boldsymbol{y}, \boldsymbol{\theta}_{\mathrm{f}}), \tag{20}$$

where $\boldsymbol{\theta}_{\mathrm{f}}$ represents the parameters of a RNN, and $\boldsymbol{h}_0$ is built on the label information as opposed to $\boldsymbol{h}_0 = \boldsymbol{0}$ in (16). As in (16), the direct generator requires additional parameters $\boldsymbol{W}_{\mathrm{f}}^y \in \mathbb{R}^{r_{\mathrm{f}} \times K}$ for $\boldsymbol{\theta}_{\mathrm{f}} = (\boldsymbol{W}_{\mathrm{f}}^o, \boldsymbol{W}_{\mathrm{f}}^x, \boldsymbol{W}_{\mathrm{f}}^h, \boldsymbol{W}_{\mathrm{f}}^y)$ to model the effect from $\boldsymbol{y}$ as follows:

$$\boldsymbol{o}(x_t, \boldsymbol{h}_t, \boldsymbol{y}, \boldsymbol{\theta}_{\mathrm{f}}) = \boldsymbol{\sigma}(\boldsymbol{W}_{\mathrm{f}}^o \boldsymbol{h}_t), \ \boldsymbol{h}_t = \phi(\boldsymbol{W}_{\mathrm{f}}^x \boldsymbol{1}_{[x_t]} + \boldsymbol{W}_{\mathrm{f}}^h \boldsymbol{h}_{t-1}), \text{ and } \boldsymbol{h}_0 = \phi(\boldsymbol{W}_{\mathrm{f}}^y \boldsymbol{y}), \tag{21}$$

where $\boldsymbol{W}_{\mathrm{f}}^o \in \mathbb{R}^{r_{\mathrm{f}} \times |\mathcal{D}|}$, $\boldsymbol{W}_{\mathrm{f}}^x \in \mathbb{R}^{|\mathcal{D}| \times r_{\mathrm{f}}}$, $\boldsymbol{W}_{\mathrm{f}}^h \in \mathbb{R}^{r_m \times r_{\mathrm{f}}}$, and r_{f} is the number of latent factors of the RNN. On this ground, the cost function in (4) becomes

$$L_{\mathrm{f}}(\boldsymbol{\theta}_{\mathrm{f}}) = -n^{-1} \sum_{i=1}^{n} (T^i)^{-1} \sum_{t=1}^{T_i} \log \left(\boldsymbol{o}_{[x_{t+1}^i]}(x_t^i, \boldsymbol{h}_{t-1}^i, \boldsymbol{y}^i, \boldsymbol{\theta}_{\mathrm{f}}) \right) + \lambda_{\mathrm{f}} J_{\mathrm{f}}(\boldsymbol{\theta}_{\mathrm{f}}), \tag{22}$$

where $\lambda_{\mathrm{f}} \geq 0$ is a tuning parameter and $J_{\mathrm{f}}(\boldsymbol{\theta}_{\mathrm{f}})$ is a nonnegative regularizer. Minimizing (22) in $\boldsymbol{\theta}_{\mathrm{f}}$ yields an estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{f}}$, thus the estimated probability $\widehat{p}^{\mathrm{f}}(x|\boldsymbol{x}_{1:t}, \boldsymbol{y}) = \boldsymbol{o}_{[x]}(x_t, \boldsymbol{h}_{t-1}, \boldsymbol{y}, \widehat{\boldsymbol{\theta}}_{\mathrm{f}})$, from (20). Then, sentence generation proceeds as in (19).

Worth of note is that the direct and indirect generators can be respectively implemented using different RNN models, for example, GPT2 for the direct RNN (20) while LSTM for the indirect RNN in (15). Moreover, different model architectures of RNNs may yield different empirical results. This aspect is illustrated in Section 5.

4.3 Coupled generator

Given the estimated probabilities $\widehat{p}^{\mathrm{f}}(x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y})$ and $\widehat{p}^{b}(x_{t+1}|\boldsymbol{x}_{1:t}, \boldsymbol{y})$. The coupled generator chooses one between $\widehat{p}^{\mathrm{f}}$ and $\widehat{p}^{b}$ on a validation set by minimizing an empirical version of the log-likelihood loss,

$$\mathrm{Ent}(\widehat{p}) = -T^{-1}\sum_{t=1}^{T}\log\widehat{p}(X_{t+1} = x_{t+1}|\boldsymbol{X}_{1:t} = \boldsymbol{x}_{1:t}, \boldsymbol{Y} = \boldsymbol{y}). \tag{23}$$

4.4 Large-scale computation

This section develops a computational scheme for the indirect generator in (14)-(17), and the direct generator in (22) can be treated by a standard RNN implementation as in [36, 29]. In particular, when stochastic backpropagation is used through the time gradient method, the computation complexity is of order of the number of parameters per time step [27].

In what follows, we apply gradient descent [41] or stochastic gradient descent [31] to solve (14). For (17), we apply a classical back-propagation algorithm. In each case, we use analytic a gradient expression for updates.

Gradient for indirect generation. The gradient expression for $\boldsymbol{\theta}_m$ in (17) is given in [29], while that for $\boldsymbol{\theta}_b$ in (14) is computed as

$$\frac{\partial L_b}{\partial \boldsymbol{\theta}_{b,k}} = \begin{cases} \lambda_b\boldsymbol{\theta}_{b,k} - n^{-1}\sum_{i=1}^{n}(T^i)^{-1}\sum_{t=1}^{T^i}\big(y_k^i - \sigma_k(\boldsymbol{\theta}_b\mathcal{E}(\boldsymbol{x}_{1:t}^i)\big)\mathcal{E}(\boldsymbol{x}_{1:t}^i), & \boldsymbol{y}^i \in \{0,1\}^K, \\ \lambda_b\boldsymbol{\theta}_{b,k} - n^{-1}\sum_{i=1}^{n}(T^i)^{-1}\sum_{t=1}^{T^i}\big(y_k^i - \boldsymbol{\theta}'_{b,k}\mathcal{E}(\boldsymbol{x}_{1:t}^i)\big)\mathcal{E}(\boldsymbol{x}_{1:t}^i), & \boldsymbol{y}^i \in \mathbb{R}^K, \end{cases} \tag{24}$$

where $\boldsymbol{\theta}_{b,k}$ denotes the kth column of $\boldsymbol{\theta}_b$.

The detail of gradient descent for the indirect generator is summarized as follows.

Algorithm 1 (Gradient descent for indirect generator):

Step 1 (Initialization): Specify a RNN architecture, randomly initialize $\boldsymbol{\theta}_b$ and $\boldsymbol{\theta}_m$, a step size for gradient descent, tuning parameters λ_b and λ_m, and the number of maximal

training iterations.

Step 2 (Regression): Updating $\boldsymbol{\theta}_b$ in (14) based on the gradient in (24).

Step 3 (RNN Gradient update): Solving (17) by updating the indirect RNN parameters $\boldsymbol{\theta}_m$ via gradient descent based on back-propagation in [29].

Step 4 (Termination): Iterate **Step 3** until a stopping criterion is met.

Algorithm 1 can be updated by a stochastic gradient scheme [2]. Lemma 1 describes computational properties of Algorithm 1.

Lemma 1. *If the cost functions L_b in (14) and L_m in (17) are continuously twice differential, and the probability measure of random initialization is absolutely continuous with respect to the Lebesgue measure. Then, $\widehat{\boldsymbol{\theta}}_b$ is a global minimizer of (14), while $\widehat{\boldsymbol{\theta}}_m$ is a local minimizer of (17) almost surely, provided that the step size in Algorithm 1 is sufficiently small.*

4.5 Theory for sentence generation

This section generalizes the theoretical result of Section 3 to the problem of next-word-prediction.

Now we use $\boldsymbol{p}_{\boldsymbol{X}|\boldsymbol{Y}}$, $\boldsymbol{p}_{\boldsymbol{Y}|\boldsymbol{X}}$, and $\boldsymbol{p}_{\boldsymbol{X}}$ to respectively represent $\{p_{X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y}}\}_{t=1}^{T}$, $\{p_{\boldsymbol{Y}|\boldsymbol{X}_{1:t}}\}_{t=1}^{T}$, and $\{p_{X_{t+1}|\boldsymbol{X}_{1:t}}\}_{t=1}^{T}$. The expected square Hellinger-distance for next-word-prediction is

$$d\big(\boldsymbol{p}_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{q}_{\boldsymbol{X}|\boldsymbol{Y}}\big) = \Big(\bar{\mathbb{E}} h^2(p_{X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y}}, q_{X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y}})\Big)^{\frac{1}{2}}, \tag{25}$$

where $\bar{\mathbb{E}}(\cdot) = T^{-1}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{X}_{1:t},\boldsymbol{Y}}(\cdot)$.

The metric entropy of $\mathcal{F}_{b,k}$ is defined by a distance $\kappa^2(\boldsymbol{p}_{\boldsymbol{Y}|\boldsymbol{X}}, \boldsymbol{q}_{\boldsymbol{Y}|\boldsymbol{X}}) = \bar{\mathbb{E}} h^2(p_{\boldsymbol{Y}|\boldsymbol{X}_{1:t}}, q_{\boldsymbol{Y}|\boldsymbol{X}_{1:t}})$. Similarly, $\kappa^2(\boldsymbol{p}_{\boldsymbol{X}}, \boldsymbol{q}_{\boldsymbol{X}}) = \bar{\mathbb{E}} h^2(p_{X_{t+1}|\boldsymbol{X}_{1:t}}, q_{X_{t+1}|\boldsymbol{X}_{1:t}})$, and $d^2(\boldsymbol{p}_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{q}_{\boldsymbol{X}|\boldsymbol{Y}})$ are used for $\mathcal{F}_{m,k}$ and $\mathcal{F}_{\mathrm{f},k}$, respectively.

The approximation error for $\boldsymbol{p}^0_{\boldsymbol{Y}|\boldsymbol{X}}$ is $\rho_b(\boldsymbol{p}^0_{\boldsymbol{Y}|\boldsymbol{X}}, \boldsymbol{p}_{\boldsymbol{Y}|\boldsymbol{X}}) = \bar{\mathbb{E}} g_\alpha\big(\frac{p^0(\boldsymbol{Y}|\boldsymbol{X}_{1:t+1})}{p(\boldsymbol{Y}|\boldsymbol{X}_{1:t+1})}\big)$. Similarly, the approximation errors $\rho_m(\boldsymbol{p}^0_{\boldsymbol{X}}, \boldsymbol{p}_{\boldsymbol{X}}) = \bar{\mathbb{E}} g_\alpha\big(\frac{p^0(X_{t+1}|\boldsymbol{X}_{1:t})}{p(X_{t+1}|\boldsymbol{X}_{1:t})}\big)$ and $\rho_{\mathrm{f}}(\boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}_{\boldsymbol{X}|\boldsymbol{Y}}) = \bar{\mathbb{E}} g_\alpha\big(\frac{p^0(X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y})}{p(X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y})}\big)$

are used for $\boldsymbol{p}^0_{\boldsymbol{X}}$ and $\boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}$.

Corollary 1 (Sequential generation)**.** *All the results in Theorems 1 and 2 continue to hold with the distance $d(\cdot,\cdot)$ defined in (25).*

Next we provide a theoretical example to illustrate Corollary 1.

Theoretical example. Suppose that the RNN in (15) is a basic recurrent network with $\boldsymbol{\theta}_m = (\boldsymbol{W}^o_m, \boldsymbol{W}^x_m, \boldsymbol{W}^h_m)$, that is, $\boldsymbol{o}(x_t, \boldsymbol{h}_{t-1}, \boldsymbol{\theta}_m) = \boldsymbol{\sigma}(\boldsymbol{W}^o_m \boldsymbol{h}_{t-1})$, $\boldsymbol{h}_t = \phi(\boldsymbol{W}^x_m \mathbf{1}_{[x_t]} + \boldsymbol{W}^h_m \boldsymbol{h}_{t-1})$, and $\boldsymbol{h}_0 = \mathbf{0}_{r_m}$, where $\boldsymbol{W}^o_m \in \mathbb{R}^{r_m \times |\mathcal{D}|}$, $\boldsymbol{W}^x_m \in \mathbb{R}^{|\mathcal{D}| \times r_m}$, and $\boldsymbol{W}^h_m \in \mathbb{R}^{r_m \times r_m}$, r_m is the number of latent factors of the RNN, and $\phi(\boldsymbol{z})$ is an activation function, such as the sigmoid function $\phi(\boldsymbol{z}) = 1/(1+\exp(-\boldsymbol{z}))$, the tanh function $\phi(\boldsymbol{z}) = \tanh(\boldsymbol{z})$, and the Rectified linear unit (ReLU) $\phi(\boldsymbol{z}) = \boldsymbol{z}_+$. For illustration, we focus on the sigmoid function.

The RNN in (20) is that $\boldsymbol{o}(x_t, \boldsymbol{h}_{t-1}, \boldsymbol{\theta}_{\mathrm{f}}) = \boldsymbol{\sigma}(\boldsymbol{W}^o_{\mathrm{f}} \boldsymbol{h}_{t-1})$, $\boldsymbol{h}_t = \phi(\boldsymbol{W}^x_{\mathrm{f}} \mathbf{1}_{[x_t]} + \boldsymbol{W}^h_{\mathrm{f}} \boldsymbol{h}_{t-1})$, and $\boldsymbol{h}_0 = \phi(\boldsymbol{W}^y_{\mathrm{f}} \boldsymbol{y})$. The network parameters are $\boldsymbol{\theta}_{\mathrm{f}} = (\boldsymbol{W}^o_{\mathrm{f}}, \boldsymbol{W}^x_{\mathrm{f}}, \boldsymbol{W}^h_{\mathrm{f}}, \boldsymbol{W}^y_{\mathrm{f}})$, where $\boldsymbol{W}^o_{\mathrm{f}} \in \mathbb{R}^{r_{\mathrm{f}} \times |\mathcal{D}|}$, $\boldsymbol{W}^x_{\mathrm{f}} \in \mathbb{R}^{|\mathcal{D}| \times r_{\mathrm{f}}}$, $\boldsymbol{W}^h_{\mathrm{f}} \in \mathbb{R}^{r_{\mathrm{f}} \times r_{\mathrm{f}}}$, and $\boldsymbol{W}^y_{\mathrm{f}} \in \mathbb{R}^{r_{\mathrm{f}} \times K}$, and r_{f} is the number of latent factors of the RNN in the direct generation.

Corollary 2 gives the generation errors of the direct and indirect generators.

Corollary 2 (Theoretical example)**.** *For the estimated next-word probabilities $\widehat{\boldsymbol{p}}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}$ by the direct generator in (22), we have that $d(\widehat{\boldsymbol{p}}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}) = O_p(\eta_{\mathrm{f}})$, where*

$$\eta_{\mathrm{f}} = \max\Big\{\Big(\frac{\Lambda_{\mathrm{f}}}{n}\log\big(\frac{n\max(r_{\mathrm{f}}, 2c_{15})2^T T^{-1/2}}{\Lambda_{\mathrm{f}}}\big)\Big)^{\frac{1}{2}}, \gamma_{\mathrm{f}}^{\frac{1}{2}}\Big\},$$

$\Lambda_{\mathrm{f}} = r_{\mathrm{f}}(2|\mathcal{D}| + r_{\mathrm{f}} + K)$, $\lambda_{\mathrm{f}} = c_{17}\eta_{\mathrm{f}}^2$, and $c_{15} > 0$ and $c_{16} > 0$ are constants with $\mathbb{E}_{\boldsymbol{Y}}\|\boldsymbol{Y}\|_2^2 \le c_{15}$. Similarly, the estimated next-word probabilities $\widehat{\boldsymbol{p}}^{b}_{\boldsymbol{X}|\boldsymbol{Y}}$ by the indirect generator in (14) and

(17) *satisfies:* $d(\widehat{\boldsymbol{p}}^b_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}) = O_p(\eta_b + \eta_m)$, *where*

$$\eta_b = \max\Big\{\Big(\frac{\Lambda_b}{n}\log\Big(\frac{c_{16}n}{\Lambda_b}\Big)\Big)^{\frac{1}{2}}, \gamma_b^{\frac{1}{2}}\Big\},$$
$$\eta_m = \max\Big\{\Big(\frac{\Lambda_m}{n+\tilde{n}}\log\Big(\frac{r_m(n+\tilde{n})2^T T^{-1/2}}{\Lambda_m}\Big)\Big)^{\frac{1}{2}}, \gamma_m^{\frac{1}{2}}\Big\},$$

$\Lambda_b = Kp$, $\Lambda_m = r_m(2|\mathcal{D}| + r_m)$, $\lambda_b = c_{18}\eta_b^2$, $\lambda_m = c_{18}\eta_m^2$, *and* $c_{16} > 0$ *and* $c_{18} > 0$ *are constants with* $\bar{\mathbb{E}}\big\|\mathcal{E}(\boldsymbol{X}_{1:t})\big\|_2^2 \le c_{16}$.

Corollary 2 says that the generation error of the indirect generator in (1) becomes $d(\widehat{\boldsymbol{p}}^b_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}) = O_p\big(\frac{\Lambda_b}{n}\log\big(\frac{n}{\Lambda_b}\big)\big)^{\frac{1}{2}}$ when $(\tilde{n}+n)/n = O(\frac{\Lambda_m \log(r_m n/\Lambda_m)}{\Lambda_b \log(c_{16}n/\Lambda_b)})$, when $\gamma_b = \gamma_m = 0$. In fact, the generation error is primarily dominated by its estimation error of $p(\boldsymbol{Y}|\boldsymbol{X}_{1:t})$, because $p(X_{t+1}|\boldsymbol{X}_{1:t})$ can be well estimated with the help of large unlabeled data with $\tilde{n} \gg n$. In this situation, the indirect method outperforms the direct method, particularly when $\Lambda_b < \Lambda_{\mathrm{f}}$, suggesting that the estimation complexity of the indirect method is less than that of the direct method. Interestingly, the generation error of the direct generator agrees with that of the maximum likelihood estimates under the Hellinger-distance [34, 38]. With respect to tuning, a large value of Λ_b, Λ_m, and Λ_{f} increases the complexity of the corresponding functional space for probability estimation, thus reduces the generation errors. Consequently, the generation errors of the direct and indirect generators indeed depend on the model complexity of parameter spaces $\mathcal{F}_b$ and $\mathcal{F}_{\mathrm{f}}$.

To illustrate the synergy of indirect and direct generators' respective strengths, we consider two situations. First, $d\big(\widehat{\boldsymbol{p}}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}\big) = o_p(1)$ but $d\big(\widehat{\boldsymbol{p}}^b_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}\big)$ is bounded away from zero if an unlabeled sample $(\boldsymbol{X}^j_{1:T^j})_{j=1}^{\tilde{n}}$ follows a different marginal distribution of the labeled sample $(\boldsymbol{X}^i_{1:T^i})_{i=1}^n$. Second, $d\big(\widehat{\boldsymbol{p}}^b_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}\big) = o_p(1)$ but $d\big(\widehat{\boldsymbol{p}}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}, \widehat{\boldsymbol{p}}^0_{\boldsymbol{X}|\boldsymbol{Y}}\big)$ is bounded away from zero in the presence of a new word in labeled but unlabeled samples. However, in both situations, $d\big(\widehat{\boldsymbol{p}}^c_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}\big) = o_p(1)$, when Kullback-Leibler divergence is equivalent to the Hellinger-distance. In other words, only the coupled generator has a generation error

tending to zero in both situations.

5 Benchmark

This section examines the performance of the coupled, indirect, and direct generators in one benchmark example, and compares with a baseline method "Separate RNN", which fits RNNs for each topic as in [36]. The benchmark concerns sentence categorization based on a text corpus in the UCI machine learning repository[2]. This corpus contains 1,039 labeled sentences collected from abstracts and introductions of 30 articles, in which five topic categories are `AIM` (a specific aim of the present paper), `OWN` (description of own work presented in the present paper), `CONTRAST` (comparison statements with other works, including strengths and weaknesses), `BASIS` (statements of agreement with other works or continuation of other works), and `MISC` (generally accepted scientific background or description of other works). These labels originate from three scientific domains: computational biology (PLOS), the machine learning repository on arXiv (ARXIV), and the psychology journal judgment and decision making (JDM). For example, a typical sentence such as "The instantaneous loss bound of SYMBOL implies only convergence in probability." is labeled as "`MISC`" according to scientific topic classification. In addition to the aforementioned labeled sentences, this corpus contains 34,481 unlabeled sentences from 300 articles in PLOS, ARXIV, and JDM.

Before proceeding, we pre-process the text corpus to filter out redundant each sentence's component so that numerical embeddings are applied for the indirect generator. First, we replace all numerical values, symbolic values, and citations by "`NUMBER`", "`SYMBOL`", and "`CITATION`", respectively, and remove all standalone punctuation marks except commas, periods, and semicolons. For unlabeled sentences, we remove words appearing less than 20 times in the corpus, which leads to an unlabeled corpus of 8,286 sentences. On this ground,

[2] https://archive.ics.uci.edu/ml/datasets/Sentence+Classification

a dictionary is constructed, consisting of 5,369 words extracted from labeled and unlabeled sentences.

For training, we generate word strings for next-word-prediction based on the maximum length of all sentences in the dataset. Thus, all the previous tokens in the sentence are contributing to predict the next word. Specifically, we create next-word-prediction sequences consisting of consecutive previous words and fill with the null word "`NULL`" to pad all word strings as the same length. An example of such a next-word-prediction sequence is displayed in Table 1. In this fashion, we gather enough training sentences as the null words do not impact our learning process. Now, 28,180 labeled next-word-prediction sequences are generated from the original 1,039 labeled sentences, together with 174,355 unlabeled sequences from the original 8,286 unlabeled sentences.

The generation performance is measured by two commonly used metrics, namely, the next-word entropy loss and the Bi-Lingual Evaluation Understudy (BLEU) loss [28] over a test sample, approximating the predicted Kullback-Leibler divergence and Jaccard distance [13], respectively. Given sentences $(\widehat{\boldsymbol{x}}^i_{1:\widehat{T}^i})_{i=1}^{n_{\text{test}}}$ generated from $\widehat{\boldsymbol{p}}$ and its reference sentence $(\boldsymbol{x}^i_{1:T^i})_{i=1}^{n_{\text{test}}}$ given a topic $\boldsymbol{y}$, the entropy loss is defined in as the empirical version of (23), while the BLEU$_l$-loss ($l = 1, \cdots, 4$) can be written as

$$\text{BLEU}_l\text{-loss}(\widehat{\boldsymbol{p}}) = 1 - n_{\text{test}}^{-1} \sum_{i=1}^{n_{\text{test}}} \exp\big(\min(1 - \frac{\widehat{T}^i}{T^i}, 0)\big) \frac{|\text{gram}_l(\boldsymbol{x}^i_{1:T^i}) \cap \text{gram}_l(\widehat{\boldsymbol{x}}^i_{1:\widehat{T}^i})|}{|\text{gram}_l(\widehat{\boldsymbol{x}}^i_{1:\widehat{T}^i})|},$$

where n_{test} is the number of sentences in the testing set, $|\cdot|$ denotes a set's size and $\text{gram}_l(\cdot)$ is the l-gram set for a sentence. For a sentence "the cat in the hat", its 1-gram set is { "the", "cat", "in", "the", "hat"}, the 2-gram set is { "the cat", "cat in", "in the", "the hat"}, and the 3-gram set is { "the cat in", "cat in the", "in the hat" }. The BLEU$_l$-loss can be computed using the NLTK library in Python. Whereas the entropy loss measures the occurrence probability of the reference sentences, the BLEU$_l$-loss focuses on exact matching between l consecutive words of two sentence. Moreover, we also consider the SF-BLEU$_l$-loss

to evaluate the diversity of a generated sentences [43], defined as

$$\text{SF-BLEU}_l\text{-loss}(\widehat{\boldsymbol{p}}) = 1 - n_{\text{test}}^{-1} \sum_{i=1}^{n_{\text{test}}} \exp\big(\min(1 - \frac{\widehat{T}^i}{\widehat{T}_{\text{min}}^{-i}}, 0)\big) \frac{\max_{j \neq i} |\text{gram}_l(\widehat{\boldsymbol{x}}^i_{1:\widehat{T}^i}) \cap \text{gram}_l(\widehat{\boldsymbol{x}}^j_{1:\widehat{T}^j})|}{|\text{gram}_l(\widehat{\boldsymbol{x}}^i_{1:\widehat{T}^i})|},$$

where $\widehat{T}_{\text{min}}^{-i} = \text{argmin}_{\widehat{T}^j; j \neq i} |\widehat{T}^j - \widehat{T}^i|$, and a high SF-BLEU$_l$-loss score means more diverse.

For training, validation, and testing, we randomly split all the labeled articles into three sets with a partition ratio of 60%, 20%, and 20%, respectively. Moreover, for a sentence $\boldsymbol{x}_{1:T}$ and its associated topic $\boldsymbol{y}$ in a testing set, five starting words $\boldsymbol{x}_{1:5}$ as opposed to the null word are given to predict the rest of a sentence.

Consider two situations of the semantic label: (1) $\boldsymbol{Y} \in \{0,1\}^K$ is categorical and is coded as a 0-1 vector using the one-hot encoding from the topic category; (2) $\boldsymbol{Y} \in \mathbb{R}^K$ is continuous with each topic as a $p = 128$-dimensional vector based on Doc2Vec. In (2), each topic is represented by the averaged embedding of all the sentences in this topic category in training data.

In the case of $\boldsymbol{Y} \in \{0,1\}^K$, the indirect generator involves (14) and (17). For (14), we perform regularized multinomial logistic regression using the Python library `sklearn`[3] on the embedded next-word prediction sequence training samples $(\mathcal{E}(\boldsymbol{x}^i_{1:t}), \boldsymbol{y}^i)$, where $\mathcal{E}(\boldsymbol{x}^i_{1:t})$ is the numerical embedding of Doc2Vec[4] of size $p = 128$ and the optimal λ_b is obtained by minimizing the entropy loss based on validation data over a set of grids $\{.0001, .001, .01, .1, 1, 10, 100\}$. For (17), the indirect RNN is trained based on both labeled and unlabeled next-word prediction sequences in training data. The indirect RNN model in (17) is structured in four layers, including an embedding layer consisting of $5,369$ nodes with each node corresponding one word in the dictionary $\mathcal{D}$, an LSTM layer of 128 latent factors, a dense layer with output dimension $5,369$. Note that the tuning parameter in (17) is fixed as $\lambda_m = .0001$ through in the embedding layer to regularize words in the absence in a training set. Similarly, the direct

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
[4] https://radimrehurek.com/gensim/models/doc2vec.html

generator trains the RNN model in (22), which has the same configuration as the indirect RNN expect that the input dimension is $|\mathcal{D}| + K = 5,374$ in its embedding layer. Moreover, Separate RNN has the same structure as the indirect RNN given each topic.

As discussed in Section 4.2, different RNN model architectures may yield different empirical performance. Toward this end, we compare the LSTM architecture with GPT2 architecture for the direct RNNs. In particular, we consider the base GPT2 with 12 layers and 117M parameters [30] for the direct method, denoted as direct-GPT2. One key difference between LSTM and GPT2 lies in its masked self-attention layer, which masks future tokens and passes the attention information through tokens that are positioned at the left of the current position.

In the case of continuous $\boldsymbol{Y} \in \mathbb{R}^p$ after numerical embedding, the indirect generation proceeds as in the categorical case except that linear regression as opposed to multinomial logistic regression in (14) is performed using `sklearn` on the labeled embedded next-word prediction sequences in training data $(\mathcal{E}(\boldsymbol{x}^i_{1:t}), \boldsymbol{y}^i)$, where each $\boldsymbol{y}^i$ is a 128-dimensional embedding vector.

All RNN models are trained using `Keras`[5] with the batch and epoch sizes 200 and 100, and optimizer as Adam, and the over-fitting is prevented by early termination [4] of patience as 20. Moreover, the coupled generator is tuned as in (5).

Table 2 about here

As indicated in Table 2, when only labeled data is available, the coupled generator delivery higher accuracy than direct and indirect generators, which suggests the advantage of the proposed method. When combining with unlabeled data, the coupled generator outperforms the direct generator and separate RNN for both categorical and continuous labels, which selects the indirect generator in this case. With respect to the entropy loss, the amounts of

[5] `https://keras.io/`

improvement of the indirect generator over the separate RNN method and direct generator are 20.3% and 14.5% for the categorical case and 29.1% and 16.1% for the continuous case. With respect to BLEU_1–BLEU_4 losses, a similar situation occurs, with the amounts of improvement vary with the best improvement around 15.6%. Concerning unlabeled data, a comparison between the indirect generator with and without unlabeled data suggests that unlabeled data indeed help to improve the performance of the indirect generation over 14.5%. Interestingly, in terms of the entropy loss, the direct generator based on fine-tuned GPT2 outperforms the direct generator and indirect generator based on LSTM without unlabeled data, while the coupled generator achieves the best performance between them. However, they perform similarly in terms of BLEU_l scores. In view of the SF-BLEU_l scores, sentences generated by the direct and indirect generators have a high degree of diversity. Moreover, the semantic label $\boldsymbol{Y}$ after sentence embeddings Doc2Vec performs slightly worse than its categorical counterpart for the indirect and direct generations, indicating that semantic relations or linguistics dependencies, as captured by the sentence embeddings, may not have an impact given that there are only five categories. Finally, as suggested in Table 3, an abstract generated based on the five categories is reasonable except for three grammatical errors that are correctable by a grammar checker[6].

Supplementary Materials

The supplementary materials provide Python codes used in real data application.

Acknowledgments

The authors thank the editors, the associate editor, and two anonymous referees for helpful comments and suggestions.

[6] http://www.grammarcheckforsentence.com/

Appendix

Proof of Lemma 1. Note that $L_b(\boldsymbol{\theta}_b)$ in (14) is convex in $\boldsymbol{\theta}_b$ and $L_b(\boldsymbol{\theta}_b)$ and $L_m(\boldsymbol{\theta}_m)$ in (17) are continuously twice-differential. Then the result follows from Theorem 4 of [21]. This completes the proof. □

Proof of Theorem 1. Note that $\widehat{p}^b_{\boldsymbol{Y}}(\boldsymbol{y}) = \int \widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}^b_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}$ and

$$d^2(\widehat{p}^b_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) = \int \Big(\big(\frac{p^0_{\boldsymbol{Y}}(\boldsymbol{y})\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}^b_{\boldsymbol{X}}(\boldsymbol{x})}{\widehat{p}^b_{\boldsymbol{Y}}(\boldsymbol{y})}\big)^{1/2} - \big(p^0_{\boldsymbol{Y}}(\boldsymbol{y})p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})\big)^{1/2} \Big)^2 d\boldsymbol{x}d\boldsymbol{y}.$$

Furthermore, $\int \widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})dy = 1$. It follows from the triangular inequality that

$$\begin{aligned} d(\widehat{p}^b_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \le & \Big(\int \Big(\big(\frac{p^0_{\boldsymbol{Y}}(\boldsymbol{y})\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}^b_{\boldsymbol{X}}(\boldsymbol{x})}{\widehat{p}^b_{\boldsymbol{Y}}(\boldsymbol{y})}\big)^{1/2} - \big(\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}^b_{\boldsymbol{X}}(\boldsymbol{x})\big)^{1/2} \Big)^2 d\boldsymbol{x}d\boldsymbol{y} \Big)^{1/2} \\ & + \Big(\int \Big(\big(\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}^b_{\boldsymbol{X}}(\boldsymbol{x})\big)^{1/2} - \big(\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})p^0_{\boldsymbol{X}}(\boldsymbol{x})\big)^{1/2} \Big)^2 d\boldsymbol{x}d\boldsymbol{y} \Big)^{1/2} \\ & + \Big(\int \Big(\big(\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})p^0_{\boldsymbol{X}}(\boldsymbol{x})\big)^{1/2} - \big(p^0_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})p^0_{\boldsymbol{X}}(\boldsymbol{x})\big)^{1/2} \Big)^2 d\boldsymbol{x}d\boldsymbol{y} \Big)^{1/2} \\ = & \, h(p^0_{\boldsymbol{Y}}, \widehat{p}^b_{\boldsymbol{Y}}) + h(p^0_{\boldsymbol{X}}, \widehat{p}^b_{\boldsymbol{X}}) + \big(\mathbb{E}\big(h^2(\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}, p^0_{\boldsymbol{Y}|\boldsymbol{X}})\big)\big)^{1/2}. \end{aligned} \tag{26}$$

Note that $\widehat{p}^b_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y}) = \widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\widehat{p}^b_{\boldsymbol{X}}(\boldsymbol{x})$. By the triangle inequality,

$$\begin{aligned} & h(p^0_{\boldsymbol{Y}}, \widehat{p}^b_{\boldsymbol{Y}}) = \Big(\int \Big(\big(\int p^0_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}\big)^{\frac{1}{2}} - \big(\int \widehat{p}^b_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}\big)^{\frac{1}{2}} \Big)^2 d\boldsymbol{y} \Big)^{\frac{1}{2}} \\ & \le \big(\int \big(\big(p^0_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})\big)^{\frac{1}{2}} - \big(\widehat{p}^b_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})\big)^{\frac{1}{2}} \big)^2 d\boldsymbol{x}d\boldsymbol{y}\big)^{\frac{1}{2}} \\ & \le \big(\int \big(\big(p^0_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})\big)^{\frac{1}{2}} - \big(p^0_{\boldsymbol{X}}(\boldsymbol{x})\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\big)^{\frac{1}{2}} \big)^2 d\boldsymbol{x}d\boldsymbol{y}\big)^{\frac{1}{2}} \\ & + \big(\int \big(\big(\widehat{p}^b_{\boldsymbol{X}}(\boldsymbol{x})\big)^{\frac{1}{2}} - \big(p^0_{\boldsymbol{X}}(\boldsymbol{x})\big)^{\frac{1}{2}} \big)^2 d\boldsymbol{x}\big)^{\frac{1}{2}} \le h(p^0_{\boldsymbol{X}}, \widehat{p}^b_{\boldsymbol{X}}) + \big(\mathbb{E}\big(h^2\big(p^0_{\boldsymbol{Y}|\boldsymbol{X}}, \widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}\big)\big)\big)^{\frac{1}{2}}. \end{aligned}$$

Hence, $d(\widehat{p}^b_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \le 2\big(h(\widehat{p}^b_{\boldsymbol{X}}, p^0_{\boldsymbol{X}}) + \big(\mathbb{E}\big(h^2(\widehat{p}^b_{\boldsymbol{Y}|\boldsymbol{X}}, p^0_{\boldsymbol{Y}|\boldsymbol{X}})\big)\big)^{1/2}\big)$. Consequently,

$$
\begin{aligned}
&P\big(d(\widehat{p}^b_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \ge 2(\eta_b + \eta_m)\big) \\
&\qquad \le P\big(h(p^0_{\boldsymbol{X}}, \widehat{p}_{\boldsymbol{X}}) \ge \eta_m\big) + P\Big(\big(\mathbb{E}_{\boldsymbol{X}}\big(h^2(p^0_{\boldsymbol{Y}|\boldsymbol{X}}, \widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}})\big)\big)^{\frac{1}{2}} \ge \eta_b\Big) \equiv I_1 + I_2.
\end{aligned}
$$

To bound I_1, let $I_3 = P\Big((n+\tilde{n})^{-1}\sum_{i=1}^{n+\tilde{n}}\big(\log(p^0_{\boldsymbol{X}}(\boldsymbol{X}^i)) - \log(p^*_{\boldsymbol{X}}(\boldsymbol{X}^i))\big) - \lambda_m J(p^0_{\boldsymbol{X}}) + \lambda_m J(p^*_{\boldsymbol{X}})\big) \ge c_9\eta_m^2/4\Big)$; $I_4 = P\Big(\sup_{d_m(p,p^0)\ge\eta_m}(n+\tilde{n})^{-1}\sum_{i=1}^{n+\tilde{n}}\big(\log(\frac{p_{\boldsymbol{X}}(\boldsymbol{X}^i)}{p^0_{\boldsymbol{X}}(\boldsymbol{X}^i)} - \lambda_m J(p_{\boldsymbol{X}}) + \lambda_m J(p^0_{\boldsymbol{X}})\big) \ge -c_9\eta_m^2/4\Big)$, where $c_9 = 1 - 2\exp(-\tau/2)/(1-\exp(-\tau/2))^2 > 0$ is a constant defined by the truncation constant $\tau > 0$. Then I_1 is upper bounded by

$$
P\Big(\sup_{d(p,p^0)\ge\eta_m}(n+\tilde{n})^{-1}\sum_{i=1}^{n+\tilde{n}}\big(\log(p_{\boldsymbol{X}}(\boldsymbol{X}^i)/p^*_{\boldsymbol{X}}(\boldsymbol{X}^i))\big) - \lambda_m J(p_{\boldsymbol{X}}) + \lambda_m J(p^*_{\boldsymbol{X}})\big) \ge 0\Big) \le I_3 + I_4.
$$

By the Markov inequality,

$$
\begin{aligned}
I_3 &\le P\Big((n+\tilde{n})^{-1}\sum_{i=1}^{n+\tilde{n}}\big(\log(p^0_{\boldsymbol{X}}(\boldsymbol{X}^i)) - \log(p^*_{\boldsymbol{X}}(\boldsymbol{X}^i))\big) \ge c_9\eta_m^2/4 - \lambda_m J(p^*_{\boldsymbol{X}})\Big) \\
&\le \prod_{i=1}^{n+\tilde{n}}\mathbb{E}_{\boldsymbol{X}}\big(\frac{p^0_{\boldsymbol{X}}(\boldsymbol{X}^i)}{p^*_{\boldsymbol{X}}(\boldsymbol{X}^i)}\big)^{\alpha}\exp\big(-\frac{c_9\alpha}{8}(n+\tilde{n})\eta_m^2\big) \\
&\le \big(1+\alpha\gamma_m\big)^{n+\tilde{n}}\exp\big(-\frac{c_9\alpha}{8}(n+\tilde{n})\eta_m^2\big) \le \exp\big(-\frac{c_9\alpha}{8}(n+\tilde{n})\eta_m^2 + (n+\tilde{n})\alpha\gamma_m\big).
\end{aligned}
$$

By Corollary 1 of [33], $I_4 \le 7\exp(-c_8(n+\tilde{n})\eta_m^2/2)$, implying $I_1 \le 7\exp(-c_7(n+\tilde{n})\eta_m^2) + \exp\big(-\frac{c_9\alpha}{8}(n+\tilde{n})\eta_m^2 + (n+\tilde{n})\alpha\gamma_m\big)$ for some constant $c_7 > 0$. For I_2, a similar probabilistic bound can be established by applying the same argument of Theorem 2 and switching the role of $\boldsymbol{X}$ and $\boldsymbol{Y}$. This leads to $I_2 \le 7\exp\big(-c_8 n\eta_b^2\big) + \exp\big(-\frac{c_9\alpha}{8}n\eta_b^2 + n\alpha\gamma_b\big)$ for some constant $c_8 > 0$. The desired result then follows. □

Proof of Theorem 2. Denote

$$I_5 = P\big(\sup_{d(p,p^0)\geq \eta_{\rm f}} \big(n^{-1}\sum_{i=1}^n \big(\log(\frac{p^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)}{p^*_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)})\big) - \lambda J(p_{\boldsymbol{X}|\boldsymbol{Y}}) + \lambda J(p^*_{\boldsymbol{X}|\boldsymbol{Y}})\big) \geq 0\big),$$
$$I_6 = P\big(n^{-1}\sum_{i=1}^n \big(\log(\frac{p^{0}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)}{p^*_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)})\big) - \lambda J(p^{0}_{\boldsymbol{X}|\boldsymbol{Y}}) + \lambda J(p^*_{\boldsymbol{X}|\boldsymbol{Y}}) \geq c_9\eta_{\rm f}^2/4\big).$$

By the definition of a minimizer, for any $\eta_{\rm f} > 0$,

$$P\big(d(\widehat{p}^{\rm f}_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \geq \eta_{\rm f}\big) \leq I_5 + I_6,$$

where $\log(\mathbb{F}_i^{(\tau)}) = \log(p^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)) - \log(p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i))$ and

$$p^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = \begin{cases} \exp(-\tau)p^*_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}), & \text{if } p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) < \exp(-\tau)p^*_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}), \\ p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}), & \text{otherwise,} \end{cases}$$

is the left truncation of $p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})$.

Next, we bound I_5 and I_6 separately. An application of the same argument as in [38] yields that

$$\begin{aligned} I_5 &\leq P\Big(n^{-1}\sum_{i=1}^n \big(\log(p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)) - \log(p^*_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i))\big) \geq c_9\eta_{\rm f}^2/4 - \lambda_{\rm f} J(p^*_{\boldsymbol{X}|\boldsymbol{Y}})\Big) \\ &\leq P\Big(\prod_{i=1}^n (\frac{p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)}{p^*_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)})^\alpha \geq \exp\big(\frac{c_9\alpha}{8}n\eta_{\rm f}^2\big)\Big) \\ &\leq \prod_{i=1}^n \mathbb{E}_{\boldsymbol{Y}}\mathbb{E}_{\boldsymbol{X}|\boldsymbol{Y}}(\frac{p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)}{p^*_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)})^\alpha \exp\big(-\frac{c_9\alpha}{8}n\eta_{\rm f}^2\big) \leq \big(1+\alpha\gamma_{\rm f}\big)^n \exp\big(-\frac{c_9\alpha}{8}n\eta_{\rm f}^2\big) \\ &\leq \exp\big(-\frac{\alpha}{8}c_9 n\eta_{\rm f}^2 + n\log(1+\alpha\gamma_{\rm f})\big) \leq \exp\big(-\frac{\alpha}{8}c_9 n\eta_{\rm f}^2 + n\alpha\gamma_{\rm f}\big), \end{aligned} \tag{27}$$

where the second inequality follows from $\lambda_{\rm f} J(p^*_{\boldsymbol{X}|\boldsymbol{Y}}) \leq c_9\eta_{\rm f}^2/8$ and the third inequality follows from Markov's inequality.

Our treatment of bounding I_6 relies on a chaining argument over a suitable partition of $\mathcal{F}_{\rm f}$ and the left-truncation of likelihood ratios as in [38, 33]. Now, consider a partition of $\mathcal{F}_{\rm f} = \cup_{k=1}^{\infty} \cup_{j=0}^{\infty} \mathcal{F}_{kj}$:

$$\mathcal{F}_{kj} = \big\{p \in \mathcal{F}_{\rm f} : 2^{i-1}\eta_n^2 \le d^2(p^0, p) \le 2^i \eta_n^2, 2^{j-1} J^0 \le J(p) \le 2^j J(p^0)\big\},$$
$$\mathcal{F}_{k0} = \big\{p \in \mathcal{F}_{\rm f} : 2^{i-1}\eta_n^2 \le d^2(p^0, p) \le 2^i \eta_n^2, J(p) \le J(p^0)\big\}; \quad k = 1, \cdots, j = 0, \cdots,$$

where $\log(\mathbb{F}_i^{(\tau)}) = \log(p^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i)) - \log(p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}^i|\boldsymbol{Y}^i))$. Then for any $\eta_{\rm f} > 0$,

$$\begin{aligned} I_6 &\le P\Big(\sup_{d(p,p^0)\ge \eta_{\rm f}} \Big(n^{-1} \sum_{i=1}^n \log\big(\mathbb{F}_i^{(\tau)}\big) - \lambda_{\rm f} J(p_{\boldsymbol{X}|\boldsymbol{Y}}) + \lambda_{\rm f} J(p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \Big) \ge -c_9 \eta_{\rm f}^2/4 \Big) \\ &\le \sum_{k=1}^{\infty} \sum_{j=0}^{\infty} P\Big(\sup_{p \in \mathcal{F}_{kj}} \Big(n^{-1} \sum_{i=1}^n \log\big(\mathbb{F}_i^{(\tau)}\big) - \lambda_{\rm f} J(p_{\boldsymbol{X}|\boldsymbol{Y}}) + \lambda_{\rm f} J(p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \Big) \ge -c_9 \eta_{\rm f}^2/4 \Big) \\ &\equiv \sum_{k=1}^{\infty} \sum_{j=0}^{\infty} I_{kj}, \end{aligned} \tag{28}$$

where $I_{kj} = P\Big(\sup_{f \in \mathcal{F}_{kj}} \big(n^{-1} \sum_{i=1}^n \log\big(\mathbb{F}_i^{(\tau)}\big) - \lambda_{\rm f} J(p_{\boldsymbol{X}|\boldsymbol{Y}}) + \lambda_{\rm f} J(p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \big) \ge -c_9 \eta_{\rm f}^2/4 \Big)$. To treat I_{kj}, we control the mean and variance of $\log\big(\mathbb{F}^{(\tau)}\big)$. By Lemma 4 of [38],

$$- \sup_{p \in \mathcal{F}_{kj}} \mathbb{E}\big(\log(\mathbb{F}^{(\tau)}) \big) = - \sup_{p \in \mathcal{F}_{kj}} \mathbb{E}_{\boldsymbol{Y}} \Big(\mathbb{E}_{\boldsymbol{X}|\boldsymbol{Y}} \big(\log(\mathbb{F}^{(\tau)}) \big) \Big) \ge c_9 \inf_{p \in \mathcal{F}_{kj}} d^2(p, p^*) \ge c_9 (2^{k-1} \eta_{\rm f})^2, \tag{29}$$

and the variance is bounded by

$$\begin{aligned} &\sup_{p \in \mathcal{F}_{kj}} \mathrm{Var}\big(\log(\mathbb{F}^{(\tau)}) \big) \le \sup_{p \in \mathcal{F}_{kj}} \mathbb{E}_{\boldsymbol{Y}} \Big(\mathbb{E}_{\boldsymbol{X}|\boldsymbol{Y}} \big(\log(\mathbb{F}^{(\tau)})^2 \big) \Big) \\ &\le 4 \exp(\tau) \sup_{p \in \mathcal{F}_{kj}} \mathbb{E}_{\boldsymbol{Y}} h^2\big(p^0_{\boldsymbol{X}|\boldsymbol{Y}}, p_{\boldsymbol{X}|\boldsymbol{Y}} \big) \le 4 \exp(\tau) (2^k \eta_{\rm f})^2 \le 8 \exp(\tau) \delta_{kj}/c_9, \end{aligned} \tag{30}$$

where the second inequality follows from Lemma 3 of [38]. Then, I_{kj} is upper-bounded by

$$\begin{aligned} I_{kj} &\le P\Big(\sup_{f\in\mathcal{F}_{kj}} \Big(n^{-1}\sum_{i=1}^{n} \log\big(\mathbb{F}_i^{(\tau)}\big) - \mathbb{E}\log\big(\mathbb{F}^{(\tau)}\big)\Big) \\ &\qquad\qquad \ge - \sup_{f\in\mathcal{F}_{kj}} \Big(\mathbb{E}\log\big(\mathbb{F}^{(\tau)}\big) + \lambda\big(J(p^0_{\boldsymbol{X}|\boldsymbol{Y}}) - J(p_{\boldsymbol{X}|\boldsymbol{Y}})\big)\Big) - c_9\eta_{\rm f}^2/4\Big) \\ &\le P\Big(\sup_{f\in\mathcal{F}_{kj}} \Big(n^{-1}\sum_{i=1}^{n} \log\big(\mathbb{F}_i^{(\tau)}\big) - \mathbb{E}\log\big(\mathbb{F}^{(\tau)}\big)\Big) \ge \delta_{kj}\Big) \le 3\exp\big(-a_3 n\delta_{kj}\big), \end{aligned} \tag{31}$$

where $a_3 > 0$ is a constant, $\delta_{kj} = c_9 2^{k-1}\eta_n^2/2 + \lambda(2^{j-1}-1)J(p^0_{\boldsymbol{X}|\boldsymbol{Y}})$, $\delta_{k0} = c_9 2^{k-2}\eta_n^2/2$, the second inequality follows from the assumption that $\lambda J(p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \le c_9\eta_{\rm f}^2/4$ and (29), and the last inequality follows from Lemma 2 and the fact that the j-th ($j \ge 2$) moment $\mathbb{E}\big(\big|\log(\frac{p^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}|\boldsymbol{Y})}{p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}|\boldsymbol{Y})})\big|^j\big)$ is bounded by

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{Y}}\mathbb{E}_{\boldsymbol{X}|\boldsymbol{Y}}\Big(\exp\Big(\Big|\log(\frac{p^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}|\boldsymbol{Y})}{p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}|\boldsymbol{Y})})\Big|\Big) - 1 - \frac{1}{2}\Big|\log(\frac{p^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}|\boldsymbol{Y})}{p^0_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{X}|\boldsymbol{Y})})\Big|\Big) \\ &\le j!2^j a_1 \mathbb{E}_{\boldsymbol{Y}}\|(p_{\boldsymbol{X}|\boldsymbol{Y}})^{1/2} - (p^0_{\boldsymbol{X}|\boldsymbol{Y}})^{1/2}\|_2^2, \end{aligned}$$

where $a_1 = \big(\exp(\tau/2) - 1 - \tau/2\big)/\big(1-\exp(-\tau/2)\big)^2 > 0$ is a constant and the last inequality follows from Lemma 5 in [34]. It suffices to verify the condition (2.4) of [38]. A combination of (28) and (31) yield that $I_6 \le \sum_{k=1}^{\infty}\sum_{j=0}^{\infty} 3\exp(-c_{13}n\delta_{kj}^2) \le 7\exp(-c_{13}n\eta_{\rm f}^2)$, which, together with (27) yields that $P\big(d(\widehat{p}^{\rm f}_{\boldsymbol{X}|\boldsymbol{Y}}, p^0_{\boldsymbol{X}|\boldsymbol{Y}}) \ge \eta_{\rm f}\big) \le I_5 + I_6 \le 7\exp(-c_{13}n\eta_{\rm f}^2) + \exp\big(-\frac{\alpha}{8}c_9 n\eta_{\rm f}^2 + n\alpha\gamma_{\rm f}\big)$. The desired result then follows. □

Proof of Theorem 3. Let $(\tilde{\boldsymbol{X}}^i, \tilde{\boldsymbol{Y}}^i)_{i=1}^N$ be a cross-validation sample. By (5),

$$-\frac{1}{N}\sum_{i=1}^{N}\log\widehat{p}^{\,c}_{\boldsymbol{X}|\boldsymbol{Y}}(\tilde{\boldsymbol{X}}^i|\tilde{\boldsymbol{Y}}^i) \le \min\Big(-\frac{1}{N}\sum_{i=1}^{N}\log\widehat{p}^{\rm f}_{\boldsymbol{X}|\boldsymbol{Y}}(\tilde{\boldsymbol{X}}^i|\tilde{\boldsymbol{Y}}^i), -\frac{1}{N}\sum_{i=1}^{N}\log\widehat{p}^{\,b}_{\boldsymbol{X}|\boldsymbol{Y}}(\tilde{\boldsymbol{X}}^i|\tilde{\boldsymbol{Y}}^i)\Big),$$

then the desired result follows from the law of large number, by taking the limit for both the sides as $N\to\infty$. This completes the proof. □

Proof of Corollary 1. For the direct sequential generation, we apply the same argument in the proof of Theorem 2. Denote

$$I_7 = P\Big(\sup_{d(p,p^0)\geq \eta_{\mathrm{f}}} \sum_{i=1}^{n} \frac{1}{nT} \sum_{t=1}^{T} \log \big(\frac{p^0(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)}{p^*(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)}\big) - \lambda_{\mathrm{f}} J_{\mathrm{f}}(\boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}) + \lambda_{\mathrm{f}} J_{\mathrm{f}}(\boldsymbol{p}^*_{\boldsymbol{X}|\boldsymbol{Y}}) \geq \frac{c_9 \eta_{\mathrm{f}}^2}{4} \Big),$$

$$I_8 = P\Big(\sup_{d(p,p^0)\geq \eta_{\mathrm{f}}} \sum_{i=1}^{n} \frac{1}{nT} \sum_{t=1}^{T} \log \big(\frac{p^{(\tau)}(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)}{p^0(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)}\big) - \lambda_{\mathrm{f}} J_{\mathrm{f}}(\boldsymbol{p}_{\boldsymbol{X}|\boldsymbol{Y}}) + \lambda_{\mathrm{f}} J_{\mathrm{f}}(\boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}) \geq -\frac{c_9 \eta_{\mathrm{f}}^2}{4} \Big).$$

Then $P\big(d(\widehat{\boldsymbol{p}}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}) \geq \eta_{\mathrm{f}}\big) \leq I_7 + I_8$, where $p^{(\tau)}(X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y})$ is the left truncation of $p(X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y})$ as defined in the proof of Theorem 2.

For I_7,

$$\begin{aligned}
I_7 &\leq P\Big(\prod_{i=1}^{n} \Big(\prod_{t=1}^{T} \frac{p^0(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)}{p^*(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)} \Big)^{\frac{\alpha}{T-1}} \geq \exp(\frac{c_9 \alpha}{8} n \eta_{\mathrm{f}}^2) \Big) \\
&\leq \prod_{i=1}^{n} \mathbb{E}\big(\prod_{t=1}^{T} \frac{p^0(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)}{p^*(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)} \big)^{\frac{\alpha}{T-1}} \exp(-\frac{c_9 \alpha}{8} n \eta_{\mathrm{f}}^2) \\
&\leq \prod_{i=1}^{n} \bar{\mathbb{E}}\Big(\big(\frac{p^0(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)}{p^*(X^i_{t+1}|\boldsymbol{X}^i_{1:t}, \boldsymbol{Y}^i)} \big)^{\alpha} \Big) \exp(-\frac{c_9 \alpha}{8} n \eta_{\mathrm{f}}^2) \\
&\leq (1 + \alpha \gamma_{\mathrm{f}})^n \exp(-\frac{c_9 \alpha}{8} n \eta_{\mathrm{f}}^2) \leq \bar{r} \exp(-\frac{\alpha}{8} c_9 n \eta_{\mathrm{f}}^2 + n \alpha \gamma_{\mathrm{f}}).
\end{aligned}$$

For I_8, let $\mathbb{F}_t^{(\tau)} = \log\big(p^{(\tau)}(X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y}) / p^0(X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y})\big)$. For the first moment,

$$\begin{aligned}
&\mathbb{E}\big(T^{-1} \sum_{t=1}^{T} \mathbb{F}_t^{(\tau)}\big) = T^{-1} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{X}_{1:t}, \boldsymbol{Y}} \big(\mathbb{E}_{X_{t+1}} \big(\mathbb{F}_t^{(\tau)} | \boldsymbol{X}_{1:t}, \boldsymbol{Y} \big) \big) \\
&\leq -c_9 T^{-1} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{X}_{1:t}, \boldsymbol{Y}} \big(\big\| (p^{(\tau)}_{X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y}})^{\frac{1}{2}} - (p^0_{X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y}})^{\frac{1}{2}} \big\|_2^2 \big) = -c_9 d^2(\boldsymbol{p}^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}).
\end{aligned}$$

For the j-th moment with $j \geq 2$,

$$\begin{aligned}
&\mathbb{E}\big|T^{-1}\sum_{t=1}^{T}\mathbb{F}_t^{(\tau)}\big|^j \leq T^{-1}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{X}_{1:t},\boldsymbol{Y}}\Big(\mathbb{E}_{X_{t+1}}\big(|\mathbb{F}_t^{(\tau)}|^j\big|\boldsymbol{X}_{1:t},\boldsymbol{Y}\big)\Big)\\
&\leq j!2^jT^{-1}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{X}_{1:t},\boldsymbol{Y}}\big(\mathbb{E}_{X_{t+1}}\big((\exp(|\mathbb{F}_t^{(\tau)}|/2)-1-|\mathbb{F}_t^{(\tau)}|/2)\big|\boldsymbol{X}_{1:t},\boldsymbol{Y}\big)\big)\\
&\leq j!2^ja_3\big(\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{X}_{1:t},\boldsymbol{Y}}\big(\|(p^{(\tau)}_{X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y}})^{\frac{1}{2}}-(p^0_{X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y}})^{\frac{1}{2}}\|_2^2\big)\big)\\
&\leq j!2^ja_1d^2(\boldsymbol{p}^{(\tau)}_{\boldsymbol{X}|\boldsymbol{Y}},\boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}),
\end{aligned}$$

where the first inequality follows from the Jensen's inequality. Then

$$P\big(d(\widehat{\boldsymbol{p}}^{\mathrm{f}}_{\boldsymbol{X}|\boldsymbol{Y}},\boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}})\geq\eta_{\mathrm{f}}\big)\leq 6\exp(-c_7n\eta_{\mathrm{f}}^2)+\exp(-\frac{\alpha}{8}c_9n\eta_{\mathrm{f}}^2+n\alpha\gamma_{\mathrm{f}}), \tag{32}$$

follows the same arguments as in the proof of Theorem 2.

For the indirect generation, let $p_t(\cdot)=p(\cdot|\boldsymbol{X}_{1:t})$ and $\mathbb{E}_t(\cdot)=\mathbb{E}(\cdot|\boldsymbol{X}_{1:t})$. Then,

$$\begin{aligned}
&d(\widehat{\boldsymbol{p}}^b_{\boldsymbol{X}|\boldsymbol{Y}},\boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}})=\Big(T^{-1}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{X}_{1:t}}\mathbb{E}_{\boldsymbol{Y}|\boldsymbol{X}_{1:t}}h^2\big(\widehat{p}^b_{X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y}},p^0_{X_{t+1}|\boldsymbol{X}_{1:t},\boldsymbol{Y}}\big)\Big)^{\frac{1}{2}}\\
&\leq 2\Big(T^{-1}\sum_{t=1}^{T}\Big(\mathbb{E}\big(h(\widehat{p}_{X_{t+1}|\boldsymbol{X}_{1:t}},p^0_{X_{t+1}|\boldsymbol{X}_{1:t}})+\big(\mathbb{E}_th^2(\widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}_{1:t+1}},p^0_{\boldsymbol{Y}|\boldsymbol{X}_{1:t+1}})\big)^{\frac{1}{2}}\big)^2\Big)\Big)^{\frac{1}{2}}\\
&\leq 2\Big(2T^{-1}\sum_{t=1}^{T}\Big(\mathbb{E}h^2(\widehat{p}_{X_{t+1}|\boldsymbol{X}_{1:t}},p^0_{X_{t+1}|\boldsymbol{X}_{1:t}})+\mathbb{E}h^2(\widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}_{1:t+1}},p^0_{\boldsymbol{Y}|\boldsymbol{X}_{1:t+1}})\Big)\Big)^{\frac{1}{2}}\\
&\leq 2\sqrt{2}\big(d(\widehat{\boldsymbol{p}}_{\boldsymbol{X}},\boldsymbol{p}^0_{\boldsymbol{X}})+d(\widehat{\boldsymbol{p}}_{\boldsymbol{Y}|\boldsymbol{X}},\boldsymbol{p}^0_{\boldsymbol{Y}|\boldsymbol{X}})\big),
\end{aligned}$$

where the first inequality follows from (26) by replacing $p(\cdot)$ as $p_t(\cdot)$, and $d^2(\widehat{\boldsymbol{p}}_{\boldsymbol{X}},\boldsymbol{p}^0_{\boldsymbol{X}})=$

$\bar{\mathbb{E}}h^2(\widehat{p}_{X_{t+1}|\boldsymbol{X}_{1:t}}, p^0_{X_{t+1}|\boldsymbol{X}_{1:t}})$, $d^2(\widehat{\boldsymbol{p}}_{\boldsymbol{Y}|\boldsymbol{X}}, \boldsymbol{p}^0_{\boldsymbol{Y}|\boldsymbol{X}}) = \bar{\mathbb{E}}h^2(\widehat{p}_{\boldsymbol{Y}|\boldsymbol{X}_{1:t+1}}, p^0_{\boldsymbol{Y}|\boldsymbol{X}_{1:t+1}})$. Therefore,

$$
\begin{aligned}
&P\big(d(\widehat{\boldsymbol{p}}^b_{\boldsymbol{X}|\boldsymbol{Y}}, \boldsymbol{p}^0_{\boldsymbol{X}|\boldsymbol{Y}}) \geq 2\sqrt{2}(\eta_b + \eta_m)\big) \leq P(d(\widehat{\boldsymbol{p}}_{\boldsymbol{X}}, \boldsymbol{p}^0_{\boldsymbol{X}}) \geq \eta_m) + P(d(\widehat{\boldsymbol{p}}_{\boldsymbol{Y}|\boldsymbol{X}}, \boldsymbol{p}^0_{\boldsymbol{Y}|\boldsymbol{X}}) \geq \eta_b) \\
&\leq 7\exp(-c_7(n+\tilde{n})\eta_m^2) + \exp(-\frac{\alpha c_9}{8}(n+\tilde{n})\eta_m^2 + \alpha(n+\tilde{n})\gamma_m) \\
&\quad + 7\exp(-c_8 n\eta_b^2) + \exp(-\frac{\alpha c_9}{8}n\eta_b^2 + \alpha n\gamma_b),
\end{aligned}
$$

where the last inequality follows from (32). Similarly, bounds for $P(d(\widehat{\boldsymbol{p}}_{\boldsymbol{X}}, \boldsymbol{p}^0_{\boldsymbol{X}}) \geq \eta_m)$ and $d(\widehat{\boldsymbol{p}}_{\boldsymbol{Y}|\boldsymbol{X}}, \boldsymbol{p}^0_{\boldsymbol{Y}|\boldsymbol{X}})$ can be established. The desired result then follows. □

Proof of Corollary 2. It suffices to verify the entropy conditions in Corollary 1. For the direct generation, let $\boldsymbol{p}_{\boldsymbol{X}|\boldsymbol{Y}} = \{p(X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y}; \boldsymbol{\theta}_{\mathrm{f}})\}_{t=1}^T$ and $\bar{\boldsymbol{p}}_{\boldsymbol{X}|\boldsymbol{Y}} = \{p(X_{t+1}|\boldsymbol{X}_{1:t}, \boldsymbol{Y}; \bar{\boldsymbol{\theta}}_{\mathrm{f}})\}_{t=1}^T$ in $\mathcal{F}_{\mathrm{f},k}$. Then

$$
\begin{aligned}
&\kappa^2(\boldsymbol{p}_{\boldsymbol{X}|\boldsymbol{Y}}, \bar{\boldsymbol{p}}_{\boldsymbol{X}|\boldsymbol{Y}}) \leq \bar{\mathbb{E}}\big\|\boldsymbol{\sigma}^{\frac{1}{2}}(\boldsymbol{W}^o_{\mathrm{f}}\boldsymbol{h}_{t-1}) - \boldsymbol{\sigma}^{\frac{1}{2}}(\bar{\boldsymbol{W}}^o_{\mathrm{f}}\bar{\boldsymbol{h}}_{t-1})\big\|_2^2 \\
&\leq \frac{1}{2}\bar{\mathbb{E}}\big\|\boldsymbol{W}^o_{\mathrm{f}}\boldsymbol{h}_{t-1} - \bar{\boldsymbol{W}}^o_{\mathrm{f}}\bar{\boldsymbol{h}}_{t-1}\big\|_2^2 \\
&\leq \bar{\mathbb{E}}\Big(\big\|(\boldsymbol{W}^o_{\mathrm{f}} - \bar{\boldsymbol{W}}^o_{\mathrm{f}})\boldsymbol{h}_{t-1}\big\|_2^2 + \big\|\bar{\boldsymbol{W}}^o_{\mathrm{f}}(\boldsymbol{h}_{t-1} - \bar{\boldsymbol{h}}_{t-1})\big\|_2^2\Big) \\
&\leq \bar{\mathbb{E}}\Big(\|\boldsymbol{W}^o_{\mathrm{f}} - \bar{\boldsymbol{W}}^o_{\mathrm{f}}\|_F^2\|\boldsymbol{h}_{t-1}\|_2^2 + \|\bar{\boldsymbol{W}}^o_{\mathrm{f}}\|_F^2\|\boldsymbol{h}_{t-1} - \bar{\boldsymbol{h}}_{t-1}\|_2^2\Big) \\
&\leq \Big(\frac{2k(4k)^{T+1}}{T(4k-1)^2}\Big)\big(\|\boldsymbol{W}^x_{\mathrm{f}} - \bar{\boldsymbol{W}}^x_{\mathrm{f}}\|_F^2 + 2r_{\mathrm{f}}\|\boldsymbol{W}^h_{\mathrm{f}} - \bar{\boldsymbol{W}}^h_{\mathrm{f}}\|_F^2 + 4kc_{\boldsymbol{Y}}\|\boldsymbol{W}^y_{\mathrm{f}} - \bar{\boldsymbol{W}}^y_{\mathrm{f}}\|_F^2\big) \\
&\quad + r_{\mathrm{f}}\|\boldsymbol{W}^o_{\mathrm{f}} - \bar{\boldsymbol{W}}^o_{\mathrm{f}}\|_F^2 \leq T^{-1}\max(2r_{\mathrm{f}}, 4kc_{15})^2(4k)^T\|\boldsymbol{\theta}_{\mathrm{f}} - \bar{\boldsymbol{\theta}}_{\mathrm{f}}\|_2^2,
\end{aligned}
$$

where the last inequality uses the fact that $\|\boldsymbol{h}_t - \bar{\boldsymbol{h}}_t\|_2^2 \leq \frac{(4k)^t-1}{4k-1}(2\|\boldsymbol{W}^x_{\mathrm{f}} - \bar{\boldsymbol{W}}^x_{\mathrm{f}}\|_F^2 + 4r_{\mathrm{f}}\|\boldsymbol{W}^h_{\mathrm{f}} - \bar{\boldsymbol{W}}^h_{\mathrm{f}}\|_F^2) + (4k)^t\|\boldsymbol{W}^y_{\mathrm{f}} - \bar{\boldsymbol{W}}^y_{\mathrm{f}}\|_F^2\|\boldsymbol{Y}\|_2^2$, which uses the fact that

$$
\begin{aligned}
\|\boldsymbol{h}_t - \bar{\boldsymbol{h}}_t\|_2^2 &\leq \|(\boldsymbol{W}^x_{\mathrm{f}} - \bar{\boldsymbol{W}}^x_{\mathrm{f}})\mathbf{1}_{[X_t]} + \boldsymbol{W}^h_{\mathrm{f}}\boldsymbol{h}_{t-1} - \bar{\boldsymbol{W}}^h_{\mathrm{f}}\bar{\boldsymbol{h}}_{t-1}\|_2^2 \\
&\leq 2\|\boldsymbol{W}^x_{\mathrm{f}} - \bar{\boldsymbol{W}}^x_{\mathrm{f}}\|_F^2 + 4r_{\mathrm{f}}\|\boldsymbol{W}^h_{\mathrm{f}} - \bar{\boldsymbol{W}}^h_{\mathrm{f}}\|_F^2 + 4k\|\boldsymbol{h}_{t-1} - \bar{\boldsymbol{h}}_{t-1}\|_2^2,
\end{aligned}
$$

and $\|\boldsymbol{h}_0 - \bar{\boldsymbol{h}}_0\|_2^2 \leq \|\boldsymbol{W}_{\mathrm{f}}^y - \bar{\boldsymbol{W}}_{\mathrm{f}}^y\|_F^2 \|\boldsymbol{Y}\|_2^2$.

Hence, $H(u, \mathcal{F}_{\mathrm{f},k}) \leq \Lambda_{\mathrm{f}} \log\big(\frac{3\max(2r_{\mathrm{f}}, 4kc_{15})(4k)^{(T+1)/2}/T^{1/2}}{u}\big)$ and the entropy condition is met by setting $\epsilon_{\mathrm{f}} = \big(\frac{\Lambda_{\mathrm{f}}}{n} \log\big(\frac{\max(r_{\mathrm{f}}, 2c_{15})2^T/T^{1/2}n}{\Lambda_{\mathrm{f}}}\big)\big)^{1/2}$.

For the indirect generation, it suffices to verify the entropy conditions in Corollary 2. Let $\boldsymbol{p}_{\boldsymbol{X}} = \{P_{\boldsymbol{X}}(X_{t+1}|\boldsymbol{X}_{1:t}; \boldsymbol{\theta}_m)\}_{t=1}^{T-1}$ and $\bar{\boldsymbol{p}}_{\boldsymbol{X}} = \{P_{\boldsymbol{X}}(X_{t+1}|\boldsymbol{X}_{1:t}; \bar{\boldsymbol{\theta}}_m)\}_{t=1}^{T-1}$. Note that $\boldsymbol{h}_0 = \bar{\boldsymbol{h}}_0 = \boldsymbol{0}_{r_m}$ and

$$
\begin{aligned}
\kappa^2(\boldsymbol{p}_{\boldsymbol{X}}, \bar{\boldsymbol{p}}_{\boldsymbol{X}}) &\leq \bar{\mathbb{E}}\Big(\|\boldsymbol{W}_m^o - \bar{\boldsymbol{W}}_m^o\|_F^2 \|\boldsymbol{h}_{t-1}\|_2^2 + \|\bar{\boldsymbol{W}}_m^o\|_F^2 \|\boldsymbol{h}_{t-1} - \bar{\boldsymbol{h}}_{t-1}\|_2^2\Big) \\
&\leq \frac{2k(4k)^{T+1}}{T(4k-1)^2}\big(\|\boldsymbol{W}_m^x - \bar{\boldsymbol{W}}_m^x\|_F^2 + 2r_{\mathrm{f}}\|\boldsymbol{W}_m^h - \bar{\boldsymbol{W}}_m^h\|_F^2\big) + r_{\mathrm{f}}\|\boldsymbol{W}_m^o - \bar{\boldsymbol{W}}_m^o\|_F^2 \\
&\leq 2r_m T^{-1}(4k)^T \|\boldsymbol{\theta}_m - \bar{\boldsymbol{\theta}}_m\|_2^2.
\end{aligned}
$$

Then, $H(u, \mathcal{F}_{m,k}) \leq \Lambda_m \log\big(\frac{3r_m(4k)^{(T+1)/2}T^{-1/2}}{u}\big)$ and the entropy condition is met by setting $\epsilon_m = O_p\Big(\big(\frac{\Lambda_m}{n+\tilde{n}} \log(\frac{r_m(n+\tilde{n})2^T T^{-1/2})}{\Lambda_m}\big)\Big)^{1/2}\Big)$.

Moreover, if $\boldsymbol{Y} \in \{0,1\}^K$,

$$
\begin{aligned}
\kappa^2(\boldsymbol{p}_{\boldsymbol{Y}|\boldsymbol{X}}, \bar{\boldsymbol{p}}_{\boldsymbol{Y}|\boldsymbol{X}}) &\leq \bar{\mathbb{E}}\big\|\big(\boldsymbol{\sigma}\big(\boldsymbol{\theta}_b \mathcal{E}(\boldsymbol{X}_{1:t})\big)\big)^{\frac{1}{2}} - \big(\boldsymbol{\sigma}\big(\bar{\boldsymbol{\theta}}_b \mathcal{E}(\boldsymbol{X}_{1:t})\big)\big)^{\frac{1}{2}}\big\|_2^2, \\
&\leq \bar{\mathbb{E}}\big\|\boldsymbol{\sigma}^{\frac{1}{2}}\big(\boldsymbol{\theta}_b \mathcal{E}(\boldsymbol{X}_{1:t})\big) - \boldsymbol{\sigma}^{\frac{1}{2}}\big(\bar{\boldsymbol{\theta}}_b \mathcal{E}(\boldsymbol{X}_{1:t})\big)\big\|_2^2 \leq \frac{1}{2}\bar{\mathbb{E}}\big\|(\boldsymbol{\theta}_b - \bar{\boldsymbol{\theta}}_b)\mathcal{E}(\boldsymbol{X}_{1:t})\big\|_2^2 \\
&\leq \frac{1}{2}\big\|\boldsymbol{\theta}_b - \bar{\boldsymbol{\theta}}_b\big\|_F^2 \bar{\mathbb{E}}\big\|\mathcal{E}(\boldsymbol{X}_{1:t})\big\|_2^2.
\end{aligned}
$$

Similarly, $H(u, \mathcal{F}_{b,k}) \leq \Lambda_b \log\big(\frac{3\sqrt{kc_{16}}}{u\sqrt{2}}\big)$ and the entropy condition is met by setting $\epsilon_b = O_p\Big(\big(\frac{\Lambda_b}{n} \log(\frac{\sqrt{c_{16}}n}{\Lambda_b})\big)^{1/2}\Big)$.

If $\boldsymbol{y} \in \mathbb{R}^K$, then

$$
\begin{aligned}
\kappa^2(\boldsymbol{p}_{\boldsymbol{Y}|\boldsymbol{X}}, \bar{\boldsymbol{p}}_{\boldsymbol{Y}|\boldsymbol{X}}) &= \bar{\mathbb{E}}\Big(1 - \exp\big(-\frac{1}{8}\|(\boldsymbol{\theta}_b - \bar{\boldsymbol{\theta}}_b)\mathcal{E}(\boldsymbol{X}_{1:t})\|_2^2\big)\Big), \\
&\leq \frac{1}{8}\bar{\mathbb{E}}\big\|(\boldsymbol{\theta}_b - \bar{\boldsymbol{\theta}}_b)\mathcal{E}(\boldsymbol{X}_{1:t})\big\|_2^2 \leq \frac{1}{8}\big\|\boldsymbol{\theta}_b - \bar{\boldsymbol{\theta}}_b\big\|_F^2 \bar{\mathbb{E}}\big\|\mathcal{E}(\boldsymbol{X}_{1:t})\big\|_2^2, \qquad (33)
\end{aligned}
$$

implying that $H(u, \mathcal{F}_{b,k}) \leq \Lambda_b \log\big(\frac{3\sqrt{kc_{16}}}{2\sqrt{2}u}\big)$, and the entropy condition holds when $\epsilon_b = O_p\big(\big(\frac{\Lambda_b}{n}\log(\frac{\sqrt{c_{16}}n}{\Lambda_b})\big)^{1/2}\big)$. This completes the proof. □

Lemma 2. *Let* $v_n(f) = n^{-1/2}\sum_{i=1}^n \big(v\big(f(\boldsymbol{X}^i), f^0(\boldsymbol{X}^i)\big) - \mathbb{E}v\big(f(\boldsymbol{X}^i), f^0(\boldsymbol{X}^i)\big)\big)$, *assume that there exist some generic constants* $a_2 > 0$ *and* $a_3 > 0$, *for* $j \geq 2$, *such that*

$$\mathbb{E}\big|v\big(f(\boldsymbol{X}), f^0(\boldsymbol{X})\big)\big|^j \leq a_2 j! 2^j d^2(f, f^0),$$

and for any $\delta > 0$, *if*

$$\int_{\delta/2^8}^{2^{1/2}\delta^{1/2}} H^{1/2}(u, \mathcal{V}_\delta)du \leq a_3 n^{1/2}\delta,$$

where $\mathcal{V}_\delta = \big\{v\big(f, f^0\big) : d^2(f, f^0) \leq \delta, f \in \mathcal{F}\big\}$, *then there exist some constants* $a_4 > 0$ *and* $a_5 > 0$ *depending on* a_2 *and* a_3 *such that*

$$P^*\Big(\sup_{d^2(f,f^0)\leq\delta; f\in\mathcal{F}} v_n(f) \geq a_4 n^{1/2}\delta\Big) \leq 3\exp(-a_5 n\delta), \tag{34}$$

where P^* *is the outer probability measure corresponding to* $p^0_{\boldsymbol{X}}$.

Proof of Lemma 2. The result follows from Lemmas 5 and Lemma 7 in [38], by replacing the Hellinger distance in Lemma 5 as a generic distance $d(\cdot, \cdot)$. □

References

[1] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[2] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] R. Caruana, S. Lawrence, and C. L. Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems*, pages 402–408, 2001.

[5] J. Cheng and M. Lapata. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*, 2016.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] F. Cozman and I. Cohen. Risks of semi-supervised learning. *Semi-supervised learning*, pages 56–72, 2006.

[8] F. G. Cozman, I. Cohen, and M. Cirelo. Unlabeled data can degrade classification performance of generative classifiers. In *Flairs conference*, pages 327–331, 2002.

[9] F. G. Cozman, I. Cohen, and M. C. Cirelo. Semi-supervised learning of mixture models. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 99–106, 2003.

[10] B. Dai, X. Shen, and J. Wang. Embedding learning. *Journal of the American Statistical Association*, (in press), 2020.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2018.

[12] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054, 2019.

[13] V. Gjorgjioski, D. Kocev, and S. Džeroski. Comparison of distances for multi-label classification with pcts. In *Proceedings of the Slovenian KDD Conference on Data Mining and Data Warehouses (SiKDD'11)*, 2011.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[15] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, 2016.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[17] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[18] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[19] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

[20] I. Langkilde. Forest-based statistical sentence generation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 170–177. Association for Computational Linguistics, 2000.

[21] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

[22] S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*, 2018.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[24] T. Mikolov, W.-T. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[25] V. Mullachery and V. Motwani. Image captioning. *arXiv preprint arXiv:1805.09137*, 2018.

[26] R. Nallapati, I. Melnyk, A. Kumar, and B. Zhou. Sengen: Sentence generating neural variational topic model. *arXiv preprint arXiv:1708.00308*, 2017.

[27] Y. Ollivier, C. Tallec, and G. Charpiat. Training recurrent networks online without backtracking. *arXiv preprint arXiv:1507.07680*, 2015.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[29] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[31] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[32] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[33] X. Shen. On the method of penalization. *Statistica Sinica*, 8(2):337–357, 1998.

[34] X. Shen and W. H. Wong. Convergence rate of sieve estimates. *Annals of Statistics*, pages 580–615, 1994.

[35] S. Smale and D.-X. Zhou. Estimating the approximation error in learning theory. *Analysis and Applications*, 1(01):17–41, 2003.

[36] I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[37] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[38] W. H. Wong, X. Shen, et al. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *Annals of Statistics*, 23(2):339–362, 1995.

[39] T. Yang and C. E. Priebe. The effect of model misspecification on semi-supervised classification. *IEEE transactions on pattern analysis and machine intelligence*, 33(10):2093–2103, 2011.

[40] D. Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[41] H.-F. Yu, F.-L. Huang, and C.-J. Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75, 2011.

[42] D. Zhou, T. Hofmann, and B. Schölkopf. Semi-supervised learning on directed graphs. In *Advances in Neural Information Processing Systems*, pages 1633–1640, 2004.

[43] Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100, 2018.

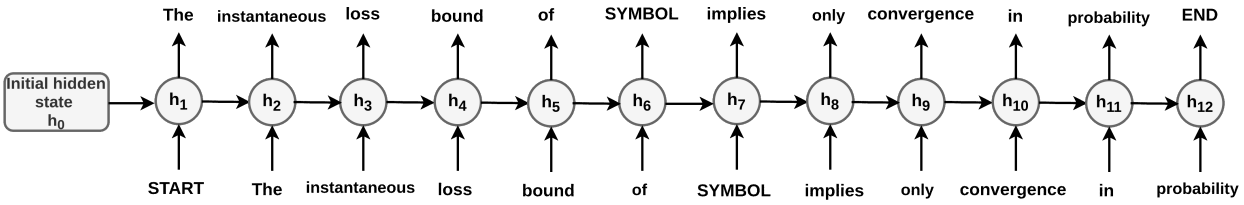

Figure 1: A generated sentence by indirect and direct RNN generators in (20) and (15), where the RNN architecture is displayed, in which sentence "*The instantaneous loss bound of SYMBOL implies only convergence in probability*" with topic "MISC" is consecutively generated by words, $\boldsymbol{h}_t$ is the hidden node of RNNs in (20) and (15), and $\boldsymbol{h}_0$ is the initial hidden state, which is zero under (15) and "MISC" under (20).

Table 1: Eleven next-word-prediction sequences with associated with a sentence.

Topic	**Sentence**
MISC	*The loss bound of SYMBOL implies convergence in probability.*
1.	Null Null Null Null Null Null Null Null Null Null START → The
2.	Null Null Null Null Null Null Null Null Null START The → loss
3.	Null Null Null Null Null Null Null Null START The loss → bound
4.	Null Null Null Null Null Null Null START The loss bound → of
5.	Null Null Null Null Null Null START The loss bound of → SYMBOL
6.	Null Null Null Null Null START The loss bound of SYMBOL → implies
7.	Null Null Null Null START The loss bound of SYMBOL implies → convergence
8.	Null Null Null START The loss bound of SYMBOL implies convergence → in
9.	Null Null START The loss bound of SYMBOL implies convergence in → probability
10.	Null START The loss bound of SYMBOL implies only convergence in probability → .
11.	START The loss bound of SYMBOL implies only convergence in probability . → END

Table 2: Test errors in loss functions–Entropy, BLEU_l, and SF-BLEU_l (standard errors in parentheses) of various generators based on 20 random partitions of the UCI sentence categorization text corpus. Here "Separate RNN", "Indirect", "Direct", "Direct-GPT2" and "Coupled" denote the separate RNN, indirect, and direct generators based on the RNN-LSTM architecture, the direct generator based on the RNN-GPT architecture, and the coupled generator, while Indirect-label or Coupled-label refers to the generation without unlabeled data.

Method	Entropy	BLEU_1-loss	BLEU_2-loss	BLEU_3-loss	BLEU_4-loss
$\boldsymbol{Y}$*: categorical label*					
Separate RNN	9.317(.040)	0.895(.010)	0.926(.008)	0.954(.007)	0.971(.005)
Indirect	7.424(.049)	0.768(.003)	0.854(.002)	0.885(.002)	0.914(.002)
Indirect-label	8.839(.060)	0.831(.008)	0.878(.005)	0.899(.004)	0.923(.003)
Direct	9.537(.054)	0.823(.008)	0.872(.005)	0.895(.005)	0.919(.004)
Direct-GPT2	8.684(.051)	0.900(.006)	0.954(.002)	0.970(.001)	0.981(.001)
Coupled	7.424(.049)	0.768(.003)	0.854(.002)	0.885(.002)	0.914(.002)
Coupled-label	8.644(.050)	0.880(.008)	0.932(.008)	0.949(.007)	0.963(.006)
		SF-BLEU_1-loss	SF-BLEU_2-loss	SF-BLEU_3-loss	SF-BLEU_4-loss
Separate RNN		0.076(.010)	0.208(.027)	0.271(.036)	0.303(.043)
Indirect		0.105(.006)	0.296(.009)	0.416(.012)	0.502(.013)
Indirect-label		0.138(.008)	0.363(.022)	0.472(.029)	0.545(.036)
Direct		0.139(.006)	0.372(.019)	0.487(.026)	0.561(.032)
Direct-GPT2		0.053(.006)	0.159(.019)	0.255(.031)	0.320(.040)
Coupled		0.105(.006)	0.296(.009)	0.416(.012)	0.502(.013)
Coupled-label		0.082(.011)	0.233(.028)	0.342(.038)	0.417(.045)
Method	Entropy	BLEU_1-loss	BLEU_2-loss	BLEU_3-loss	BLEU_4-loss
$\boldsymbol{Y}$*: continuous label based on Doc2Vec* [23, 24]					
Indirect	7.641(.036)	0.768(.005)	0.851(.003)	0.883(.003)	0.912(.003)
Indirect-label	8.512(.041)	0.912(.010)	0.937(.008)	0.949(.007)	0.960(.005)
Direct	9.102(.050)	0.916(.010)	0.939(.007)	0.950(.005)	0.961(.004)
Coupled	7.641(.036)	0.768(.005)	0.851(.003)	0.883(.003)	0.912(.003)
Coupled-label	8.512(.041)	0.912(.010)	0.937(.008)	0.949(.007)	0.960(.005)
		SF-BLEU_1-loss	SF-BLEU_2-loss	SF-BLEU_3-loss	SF-BLEU_4-loss
Indirect		0.097(.005)	0.261(.008)	0.361(.010)	0.440(.012)
Indirect-labeled		0.064(.010)	0.165(.026)	0.211(.035)	0.232(.040)
Direct		0.079(.014)	0.202(.037)	0.252(.046)	0.271(.050)
Coupled		0.097(.005)	0.261(.008)	0.361(.010)	0.440(.012)
Coupled-label		0.064(.010)	0.165(.026)	0.211(.035)	0.232(.040)

Table 3: An abstract generated by the coupled generator based on one random partition of the UCI benchmark text corpus for sentence categorization. Here five sentences (1)-(5) correspond to five categories: AIM, OWN, CONTRAST, BASIS, MISC, with the first five words of each sentence prespecified. All sentences are grammatically legitimate except "**improves**" in (4) suffers from an error, and **kolmogorov** in (1) and **israelis** in (3) should be capitalized. These errors are correctable by a grammar checker.

(1) The paper extends research on the theory of choice rules. (2) We test our predictions using the ideas and the notion of **kolmogorov** *complexity bound on the number of examples of the data sets. (3) The results demonstrate that* **israelis** *models can be used to provide new results for classification accuracy. (4) We show that implementation concerns the performance of the learning algorithm for* **improves** *the optimal predictor of the prediction. (5) The effect of balance is described by a high level of events and the objects that are shared.*