Improving Text Auto-Completion with Next Phrase Prediction

Dong-Ho Lee1,2, Zhiqiang Hu3, Roy Ka-Wei Lee3
1Department of Computer Science, University of Southern California
2Upstage AI Research, Republic of Korea
3Singapore University of Technology and Design
dongho.lee@usc.edu, zhiqiang_hu@mymail.sutd.edu.sg, roy_lee@sutd.edu.sg

Abstract

Language models such as GPT-2 have performed well on constructing syntactically sound sentences for text auto-completion task. However, such models often require considerable training effort to adapt to specific writing domains (e.g., medical). In this paper, we propose an intermediate training strategy to enhance pre-trained language models’ performance in the text auto-completion task and fastly adapt them to specific domains. Our strategy includes a novel self-supervised training objective called Next Phrase Prediction (NPP), which encourages a language model to complete the partial query with enriched phrases and eventually improve the model’s text auto-completion performance. Preliminary experiments have shown that our approach is able to outperform the baselines in auto-completion for email and academic-writing domains.

1 Introduction

Natural language interface (NLI) applications such as Personal assistants (e.g., Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana) and search engines (e.g., Google) have become an integral part of our everyday life. Among the many features in NLI applications, text auto-completion, which aims to suggest words, phrases, and sentences that complete the user’s textual input, is a common, but key feature. Smart reply (Kannan et al., 2016) and Smart compose (Chen et al., 2019) are two recent works that provide contextual assistance to aid users in completing everyday text such as emails, search engine inputs, etc.

While recent advances in deep neural models have shown impressive performance on the text auto-completion task, these models generally require a large amount of everyday text and huge amount of computing power for training to generate adequate suggestions (Chen et al., 2019). The challenge is compounded when we perform auto-completion in specific domains such as academic writing, which requires a large training corpus for specific expertise. Table 1 illustrates the difficulty in domain-specific auto-completion with the same amount of supervisions.

A potential solution to address the challenges in text auto-completion is exploiting the Decoder-only Transformer model such as GPT-2 (Radford et al., 2019). The model performs well on constructing syntactically sound sentences from partial query. However, GPT-2 requires a huge fine-tuning effort to construct sentences of expert domains. Figure 1 shows an example of GPT-2 auto-completion suggestions for computer science domain sentences before fine-tuning. Recently, text-to-text transformers such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) have demonstrated great potential in natural language generation (NLG) tasks by using

<table>
<thead>
<tr>
<th>Test Perplexity</th>
<th>Email</th>
<th>Academic Writing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-LSTM</td>
<td>1.88 ±0.05</td>
<td>3.17 ±0.03</td>
</tr>
</tbody>
</table>

Table 1: Perplexity Comparison of Bi-LSTM. Train Bi-LSTM for the language modeling with the same amounts (100K) of training instances for each domain. Perplexity of Academic writing domain is almost double of emails.

GPT-2 Suggestions

1. The approach for organizing the computation process on the gpu is described.
2. A reservoir is usually a recurrent neural network with fixed random connections.

Computer Science related article

1. The approach for organizing the computation of the values is identical for two types.
2. A reservoir is usually a recurrent neural activity or the activity of an associated memory.

Figure 1: Comparison of generated outputs. GPT-2 can generate syntactically sound, and semantically general sentence from partial query. However, it still needs to be fine-tuned a lot to generate semantically expert domain (e.g. Computer Science) focused sentence.
She bought a top and bottom from the strange little shop.

Generate next phrase: She bought a top and bottom from that strange little shop.

(A) a top and bottom
(B) that strange little shop

Overview of next phrase prediction. From the constituent tree, we retrieve the child phrases and group them according to their types (i.e., noun phrase (NP), verb phrase (VP), preposition phrase (PP)). Next, we randomly select a group that contains more than two phrases. Finally, we construct a generative QA style instance, where the phrases in the group are options to be selected as the correct next phrase for the input phrase.

2 Overview

In this section, we first formalize the auto-completion problem, and then introduce the workflow of our intermediate training strategy.

2.1 Problem Statement

Given partial query $p = [s^{(1)}, s^{(2)}, \ldots, s^{(n)}]$, an auto-completion returns $q = [t^{(1)}, t^{(2)}, \ldots, t^{(n)}]$, where $q$ is a syntactic and semantic extension of $p$. Specifically, every token of $p$ is a prefix of $q$, and every token of $q$ is a suffix of $p$; $[p; q]$ is a full sentence. We evaluate the auto-completion model’s performance on two attributes: (a) the soundness of $q$, and (b) the semantic similarity of $q$ with the ground truth.

2.2 Workflow

The workflow consists of two main steps, starting with a pre-trained T5 model: (i) applying the proposed self-supervised objective NPP for intermediate-task training, and (ii) fine-tuning on target auto-completion task.

3 Next Phrase Prediction

The key idea of the next phrase prediction (NPP) objective is to train a text-to-text transformer to complete the partial query with adequate phrases. The underlying intuition of our proposed approach is as follows: (1) Phrases tend to express meaning beyond simple word concatenation. For example, noun phrase such as “Recurrent Neural Network” is constructed by three different words (“Recurrent”, “Neural”, “Network”), where each word has its own meaning. (2) Common phrases tend to be used on their own in the text. For instance, the prepositional phrase such as "in this paper" frequently appears in academic writing domain. Unlike existing language models that are trained to neglect such characteristics of phrases and predict the next word or span of the text, text auto-completion can be improved by performing phrase-completion.
completion as an intermediate training strategy in an effort to make the most of the phrase. Specifically, NPP involves two main steps: (i) Phrase Extraction, and (ii) Generative Question Answering (QA).

3.1 Phrase Extraction
We first begin by extracting phrases using constituency parsing to retrieve qualitative phrases. Given an input \( x = [x_1, x_2, \ldots, x_n] \), we first conduct constituency parsing using AllenNLP (Gardner et al., 2017) and extract the Noun Phrase (NP), Verb Phrase (VP), and Prepositional Phrase (PP). The extracted phrases are grouped into sets according to their types, denoted as \( S_{vp}, S_{np}, \) and \( S_{pp} \), respectively: \( S_{vp} = \{vp_1, vp_2, \ldots, vp_q\} \), \( S_{np} = \{np_1, np_2, \ldots, np_p\} \), and \( S_{pp} = \{pp_1, pp_2, \ldots, pp_q\} \). For each phrase, we only keep the node that does not have a child node of the same phrase type. For example, the sentence “She wants to eat pie.” has three VPs as follows:

1. wants to eat pie (VP) → wants (VBZ) to eat pie (VP)
2. to eat pie (VP) → to (TO) eat pie (VP)
3. eat pie (VP)

To construct \( S_{vp} \) for this sentence, we only consider "eat pie" as \( vp_i \) to avoid word overlap between phrases.

3.2 Generative QA
After retrieving the phrases, we train the language model to predict the correct next phrase in a generative QA task setting (Khashabi et al., 2020). Specifically, from \( S_{vp}, S_{np}, \) and \( S_{pp} \), we randomly choose a set \( S \) that has more than two phrases. To formulate the Generative QA task with the selected \( S \), here we present both the question and answer: If the answer is a randomly chosen phrase \( p \) from \( S \), then the question is composed of partial query \( q \) in which the chosen phrase \( p \) is an extension of \( q \) and all phrases in \( S \) as answer choices. The model is trained to output the correct phrase \( p \), given partial query \( q \) and answer choices \( S \). Figure 2 shows a real example of this format by choosing \( S_{vp} \) as \( S \), "a top and bottom" as \( q \), and "She bought" as \( p \).

4 Experiments

4.1 Details for intermediate training
We train a pre-trained T5-base model with NPP. We randomly sample 1M sentences from the English Wikipedia corpus\(^1\), which is used for pre-training BERT and its variants, as the source data for NPP. The corpus has about 1.2B tokens, which is considerably less than the 34B token used in T5, and 10B tokens used in GPT2\(^2\).

4.2 Target Dataset
To show the effectiveness of our proposed method, we utilize two domains of text corpus to create the text auto-completion datasets:

- **Email**: We utilize Enron email corpus\(^3\) for general domain which is written in English collected from internal communication within a large business organization.
- **Academic writing**: We collect the abstracts of academic articles from ArnetMiner (Tang et al., 2008). The articles are written in English and mainly from the Computer Science domain, which are extracted from DBLP\(^4\), ACM\(^5\), etc.

Table 2 summarizes the statistics of the datasets used in our experiments. For data processing, we first extract the sentences from these text corpus. For each sentence, we split into pairs \((p, q)\) by all word points. We consider \( p \) as partial phrase query to predict completion of the remaining phrase \( q \) in the sentence. Note that the \((p, q)\) formulation is used in fine-tuning the base models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emails</td>
<td>156,998</td>
<td>13,474</td>
<td>15,030</td>
</tr>
<tr>
<td>Academics</td>
<td>161,885</td>
<td>20,206</td>
<td>19,953</td>
</tr>
</tbody>
</table>

Table 2: Statistics of datasets.

4.3 Base models
We compare our proposed approach with other pre-trained language generation models. We fine-tuned the following models on our training data in a sequence-to-sequence format: (1) **GPT-2** (Radford et al., 2019) is the pre-trained GPT-2 large model, which has 774M parameters. For fine-tuning, we condition the model on the format \( p = q \). For inference, we sample from the fine-tuned GPT-2 model after a prompt of the partial phrase \( p \) with beam search, and cleaning the samples by post-processing. Then, we use the first sample as the output sentence. (2) **T5** (Raffel et al., 2020) is the pre-trained T5-base model, which has 220M parameters. For fine-tuning, we prepend the prefix:

---

1.\(^1\)https://dumps.wikimedia.org/enwiki/latest/
2.\(^2\)Assuming the average token size is four characters.
3.\(^3\)https://www.cs.cmu.edu/~./enron/
4.\(^4\)https://dblp.org/
5.\(^5\)https://www.acm.org/
Table 3: Experimental Results. The first group of models are baselines which are not intermediately trained. Last group of models are intermediately-trained with different objectives. Best models are bold within each metric.

<table>
<thead>
<tr>
<th>Model / Metrics</th>
<th>Emails</th>
<th>Academic Writing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU-4</td>
<td>METEOR</td>
</tr>
<tr>
<td>GPT-2 (Radford et al., 2019)</td>
<td>1.1</td>
<td>6.6</td>
</tr>
<tr>
<td>T5 (Raffel et al., 2020)</td>
<td>2.8</td>
<td>6.8</td>
</tr>
<tr>
<td>NSP+T5</td>
<td>3.0</td>
<td>6.9</td>
</tr>
<tr>
<td>NPP+T5 (Ours)</td>
<td>3.2</td>
<td>7.1</td>
</tr>
</tbody>
</table>

Table 4: Generated Examples of Academic Writing. For the same partial queries from academic writing dataset, we compare the generated completions between T5 and NPP+T5. Underlines are overlap words between the original completion and generated completions.

"generate next phrase:" to partial query p and feed into the model to generate completion q. (3) NSP+T5 is intermediately-trained based on T5-Base using Next Sentence Prediction (NSP), which is used in BERT (Devlin et al., 2019) pre-training. (4) NPP+T5 is intermediately-trained based on T5-base using Next Phrase Prediction (NPP), which is our proposed objective.

4.4 Evaluation Metrics

To evaluate the syntactic and semantic soundness of generated sentences, we exploit several widely used automatic metrics to assess the performance, such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016). These metrics evaluate whether the model is able to generate semantically expert domain focused sentence by measuring surface similarities and associations between system generations and original text.

4.5 Experimental Results

Table 3 shows the experimental results of text auto-completion on the email and academic writing datasets. We observed that the model intermediately trained with our objective outperforms the base models on both datasets.

Specifically, our approach, NPP+T5, outperforms NSP+T5 by a margin from 0.2 to 0.3 BLEU/METEOR/SPICE score, suggesting that predicting the next phrase is more effective than predicting next sentence in text auto-completion task. Moreover, we also observe that NPP+T5 outperforms GPT-2 even though the number of parameters in NPP+T5 is less than half of GPT-2. The experimental results demonstrated the flexibility of our proposed approach, which can serve as "plug-and-play" for any text-to-text transformer models and enhance their performance in the text auto-completion task.

Table 4 shows the comparison of generated suggestions for the same partial query between T5 and NPP+T5. We can observe that the completions by NPP+T5 are generally more acceptable in terms of semantic similarity between generated completions and original text.

5 Conclusion

In this paper, we propose a novel intermediate training strategy that encourages the model to complete the partial query with enriched phrases and eventually improving the performance of the text auto-completion system. Our proposed approach enhances state-of-the-art language model’s performance by intermediately training it with our next phrase prediction self-supervised objective. Preliminary experiments have shown that our approach is able to outperform the baselines in auto-completion for email and academic-writing domains with only around 1.2B tokens of training. For future work, we aim to experiment our proposed approach on text auto-completion in more writing domains and develop a demonstration system to better showcase our approach in text auto-completion.
References


