GENIUS: Sketch-based Language Model Pre-training via Extreme and Selective Masking for Text Generation and Augmentation

Biyang Guo[1†], **Yeyun Gong**[2], **Yelong Shen**[3]
Songqiao Han[1], **Hailiang Huang**[1*], **Nan Duan**[2*], **Weizhu Chen**[3*]
[1]AI Lab, SIME, Shanghai University of Finance and Economics
[2]Microsoft Research Aisa, [3]Microsoft Azure AI

Abstract

We introduce **GENIUS**: a conditional text **gen**erat**i**on model **u**sing **s**ketches as input, which can fill in the missing contexts for a given sketch (key information consisting of textual spans, phrases, or words, concatenated by mask tokens). GENIUS is pre-trained on a large-scale textual corpus with a novel *reconstruction from sketch* objective using an extreme and selective masking strategy, enabling it to generate diverse and high-quality texts given sketches. Comparison with other competitive conditional language models (CLMs) reveals the superiority of GENIUS's text generation quality. We further show that GENIUS can be used as a strong and ready-to-use data augmentation tool for various natural language processing (NLP) tasks. Most existing textual data augmentation methods are either too conservative, by making small changes to the original text, or too aggressive, by creating entirely new samples. With GENIUS, we propose **GeniusAug**, which first extracts the target-aware sketches from the original training set and then generates new samples based on the sketches. Empirical experiments on 6 text classification datasets show that GeniusAug significantly improves the models' performance in both in-distribution (ID) and out-of-distribution (OOD) settings. We also demonstrate the effectiveness of GeniusAug on named entity recognition (NER) and machine reading comprehension (MRC) tasks.[1]

1 Introduction

When writing an article, we usually start by drawing up a general framework containing the key elements or thoughts we wish to convey (which we call as **sketch** in this work), on the basis of which we write the whole content. Motivated by this, we want to train a conditional language model to mimic this process to generate a full text based on a sketch, where the sketch may only make up a very small part of the full text. We define this as the *sketch-based text generation*, which has rich potential applications such as human writing assistance (Shih et al., 2019), automatic story generation (Yao et al., 2019), or generating new samples as data augmentation for downstream NLP tasks (Wu et al., 2019; Kumar et al., 2020).

Sketch-based text generation can be viewed as the reconstruction of an **extremely-masked** text. Many pretrained transformer models (PTMs) such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), T5 (Raffel et al., 2019), and BART (Lewis et al., 2020), involve a text reconstruction objective during their pre-training. However, due to the low masking ratio (e.g. 15% for BERT/RoBERTa/T5 and 30% for BART), these models can only be used to recover a **mildly-masked** text, where a few tokens or small spans are missing. This makes them incompatible with our goal – to reconstruct the whole content based only on a sketch where most of the content may be corrupted. Zhu et al. (2019); Liu et al. (2019a); Donahue et al. (2020) also propose text-infilling models, but these models are either based on traditional architectures like RNNs (Hochreiter and Schmidhuber, 1997) or only trained on specific domains without large-scale pre-training, limiting their usage on downstream tasks.

In this work, we present a sketch-based text generation model – **GENIUS** (**GEN**erat**I**on **U**sing **S**ketch as input) – pretrained on a large-scale corpus using a novel *reconstruction from sketch* pre-training objective with an **extreme** and **selective** masking strategy. Compared with previous related methods, GENIUS can generate more diverse, detailed, and coherent text based on sketches. GENIUS also exhibits a strong attribute controlling ability to generate content towards certain attributes like sentiment or topic, which makes GENIUS more flexible for

[†]Work done during the internship at MSRA NLC group
[*]Corresponding author, email: hlhuang@shufe.edu.cn; nanduan@microsoft.com;wzchen@microsoft.com.
[1]Code and models are publicly available at https://github.com/microsoft/SCGLab and https://github.com/beyondguo/genius

conditional text generation and other applications.

We then illustrate the great potential of GENIUS for **textual data augmentation** as an important application of GENIUS. Data augmentation is widely used to enhance the generalization of deep learning models, especially when the training data is scarce or noisy (Feng et al., 2021). Various data augmentation methods have been proposed in the NLP field, such as rule-based text-editing methods (Wei and Zou, 2019; Feng et al., 2020; Guo et al., 2022), back-translation (Sennrich et al., 2016; Yu et al., 2018), masked language model-based (Wu et al., 2019; Kumar et al., 2020) and auto-regressive model-based methods (Anaby-Tavor et al., 2020; Kumar et al., 2020). However, we argue that most previous methods are *either too conservative or too aggressive*: 1) Conservative methods like EDA (Wei and Zou, 2019), MLM-based methods (Wu et al., 2019; Kumar et al., 2020) only make small modifications to the original text. If the modifications are too large, they may harm the model's performance. Therefore, the generated samples are semantically and structurally very close to the original ones, which limits the diversity; 2) Aggressive methods aim to create completely new training samples rather than just altering the original ones. For example, LAMBADA (Anaby-Tavor et al., 2020) fine-tunes GPT-2 (Radford et al., 2019) to learn the patterns of training data and generate new sentences conditioned on certain labels. This introduces large diversity to the training set but is less controllable in terms of data quality, which means it has a higher chance of producing undesirable samples.

With this observation, we propose a sketch-based data augmentation method named **GeniusAug** which balances between these two extremes, built upon the strong generation ability of GENIUS. With GeniusAug, we first extract the essential parts of the text and then use the GENIUS model to produce new contexts around these essential parts. By doing so, the diversity is much higher than the conservative methods as larger parts of the original text are changed. At the same time, the quality is more reliable than aggressive methods since the core semantics are retained. Large-scale pre-training of GENIUS enables GeniusAug to be a ready-to-use tool for augmentation without the need for further fine-tuning on downstream tasks (though we also show that fine-tuning may lead to extra performance gain). The flexible nature of GeniusAug also makes it a general data augmentation method that can be easily applied to various NLP tasks, including text classification, named entity recognition, and machine reading comprehension. Extensive experiments show that GeniusAug can generate high-quality training samples to boost the models' in-distribution (ID) and out-of-distribution (OOD) generalization performance, which outperforms traditional methods by large margins.

The rest of the paper is organized as follows. Section 2 introduces the pre-training method of GENIUS and the method of applying GeniusAug for data augmentation. Section 3 discusses the performances of other possible pre-training strategies for GENIUS and evaluates the sketch-based text generation quality in comparison with other related methods. Section 4 elaborates on the data augmentation experiments including text classification, named entity recognition, and MRC tasks. We discuss the related work in Section 5 and the limitations of our work in Section 6.

2 Methodology

2.1 GENIUS Pre-Training

We aim to train a conditional language model that can generate content from a sketch containing the key elements of the article. This is similar to the text reconstruction from corrupted text task, which is a common denoising pre-training objective for PTMs like BERT and BART (see Section 5.1). However, these PTMs only reconstruct a small fraction of masked tokens from the original text (15% for BERT and 30% for BART), while we want to reconstruct the whole content from a sketch that may mask most of the text, which is beyond the capability of BERT or BART. Therefore, we propose a new pre-training task, *reconstruction from sketch*, where we first extract a sketch from a document and then train the language model to recover the original text from the sketch.

To obtain a sketch that preserves the core semantics and the outline of the original text, we apply an *extraction-projection-masking* procedure. We use the unsupervised keywords extraction algorithm YAKE (Campos et al., 2020) to extract keywords or key-phrases (up to 3-grams) from the original text, which account for roughly 1/5 of the text length. We then project these key parts back to the original text, allowing for multiple oc-

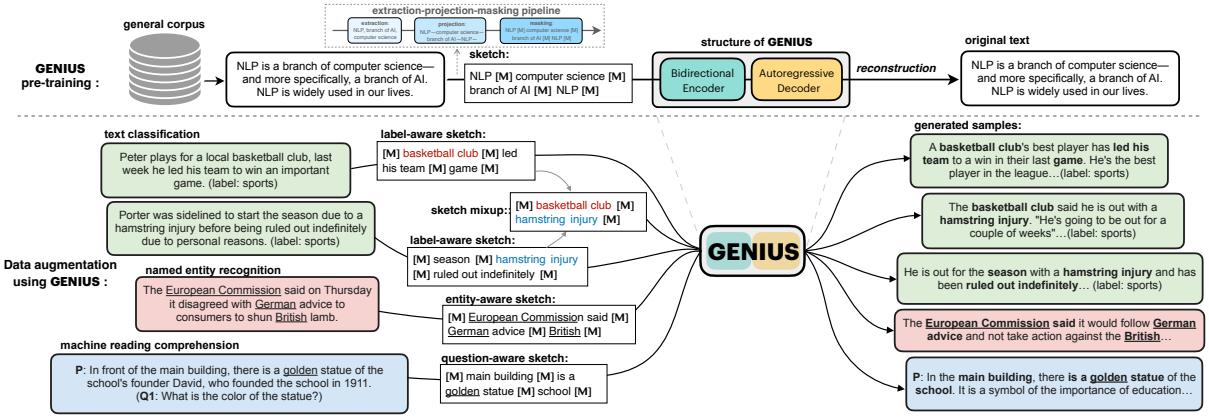


Figure 1: Illustrations of the pre-training and augmentation process of GENIUS, where [M] refers to the model's mask token. The pre-training is conducted only once and then can be applied to various NLP tasks for data augmentation. During augmentation, we extract the *target-aware* sketches as GENIUS's inputs. The *sketch mixup* shown in the figure is an example approach to further improve the diversity.

currences and overlaps of the key-phrases. We replace the rest of the text with a *single mask token*, resulting in an average masking ratio of about 73%[2]. We construct about 27 million (27,599,676) <sketch,text> pairs from the C4-realnewslike corpus (Raffel et al., 2019) for pre-training. We use a bidirectional encoder and an auto-regressive decoder, initialized with BART weights. To facilitate future related research on other languages, we have also pre-trained a Chinese version of GENIUS, and is developing a multilingual version, see Appendix A.1 for more details. The upper part of Figure 1 shows an overview of the pre-training process.

The pre-training task of GENIUS differs from previous denoising pre-training objectives in two aspects: **Extreme masking**: We mask up to 80% of the text, while previous methods usually mask a small proportion of text; **Selective masking**: We mask the less-informative parts of the text based on a sketch extraction pipeline, instead of masking random tokens/spans of the original text.

We will show that the following points are crucial for GENIUS's pre-training: 1) the sketch should contain the key elements of the original text instead of randomly selected ones; 2) the order and occurrences of these elements should remain unchanged; 3) the missing parts are replaced with single mask tokens, which echos the text-infilling configuration of GENIUS's backbone – BART. These designs alleviate the difficulty of high masking reconstruction, as well as the catastrophic forgetting problem (French, 1999) during the continued pre-training. Experiments and discussions about GENIUS's sketch design are placed in Section 3.

[2]We sampled 1 million documents from our training set, the average masking ratio (%) is 72.97 ± 7.05

2.2 Data Augmentation with GENIUS

Data augmentation is an important application for natural language generation (NLG) models, which is also a valuable evaluation of whether the generated text can be used in real applications. Here we introduce **GeniusAug** which utilizes the strong ability of GENIUS for data augmentation.

Target-aware sketch extraction. To generate useful augmentation samples for downstream tasks using GENIUS, we need to feed the model with sketches that contain the information relevant to the task. We design a method called *target-aware sketch extraction* to select the task-related parts from the original samples as the input sketches. Given a document d, all its n-grams (1 to 3-grams) $[w_1, w_2, ..., w_m]$ and its *target-related information* (**TRI**) t, we use a pre-trained encoder $\mathbf{E}$ to obtain their embeddings e_d, $[v_1, v_2, ..., v_m]$, and e_t respectively. The TRI depends on the task: for text classification, it is the category or its description; for NER, it is the entities; for machine reading comprehension, it is the question. We use sentence-BERT (Reimers and Gurevych, 2019) as the encoder $\mathbf{E}$. Next, we compute the similarity between each n-gram and a fused embedding of the document and its TRI, which is defined as:

$$e_f = \lambda e_d + (1-\lambda) e_t$$
$$s_i = \frac{e_f \cdot v_i}{\|e_f\| \|v_i\|}, i = 1, 2, ...m$$

where e_f is the fused embedding, s_i is the similarity score for the ith n-gram w_i. We set $\lambda = 0.5$ and

sketch	[M] **use machine learning** [M] **AI techniques** [M]
GENIUS *output*	How do you **use machine learning** and other **AI techniques**? What are the benefits and disadvantages of AI?
prompt = "Medicine"	How do you **use machine learning** and **AI techniques** to help ***patients***? I am a software engineer. I have been working in AI for over 10 years. I am passionate about ***helping patients with their health problems.***
prompt = "Finance"	How do you **use machine learning** in your ***business***? **AI techniques** are a big part of the digital ***transformation of the economy.***
prompt = "Good news"	you ***can*** now **use machine learning** and **AI techniques** to ***help you*** get the most out of your new job.
prompt = "Bad news"	you ***can't*** **use machine learning** to do what you want. ***It's not possible*** to use **AI techniques** to predict what's going to happen in a particular situation.

Table 1: Examples of synthetic sentences generated by GENIUS and the effect of adding specific prompts for "attribute controlling". The ***red italicized*** represents the text that is close to the given attribute.

choose the top 20% n-grams with the highest similarity scores as the keywords/key-phrases. Finally, we construct the sketch from these selected parts following the steps in Section 2.1.

Generating new training samples. After obtaining the sketch of a training sample, we use the pre-trained GENIUS to generate new samples conditioned on this sketch. We use beam search with random sampling to decode and generate the text. Table 1 illustrates an example sketch and its corresponding generated text by GENIUS. GENIUS is able to fill in the blanks ([M]) with multiple words or long spans, which is different from BERT or BART that can only fill in one or a few words. In the meanwhile, the key parts of the sketch remain in the generated text, which guarantees that the generated text won't have a large semantic shift from the original text.

Attribute controlling. By adding a topic or sentiment prompt before the sketch, we can further control GENIUS to generate content towards certain attributes, as shown in the last four rows in Table 1. Note that we didn't specifically pre-train or fine-tune GENIUS with an attribute-conditioned generation task, like CTRL (Keskar et al., 2019) or (Ziegler et al., 2019), nor did we use extra attribute models to control the generation like PPLM (Dathathri et al., 2019). The attribute controlling ability of GENIUS is acquired from the *reconstruction from sketch* pre-training. This characteristic makes GENIUS flexible to control the quality and diversity during data augmentation, especially for sentiment or topic classification.

More advanced options. Apart from the standard usage introduced above, we find there exist other interesting ways to generate more diverse and high-quality training samples. For example, inspired by the Mixup technique (Zhang et al., 2018), we propose **sketch mixup** to *combine the target-related parts from multiple training samples* to form the sketch, which is also shown in Figure 1. Our experiments show this approach can bring further performance gains for some tasks compared with standard usages of GENIUS. This inspires us that the *sketch designing* can be an interesting and worthy future research topic to further exploit the ability of the pre-trained GENIUS model.

3 Experiments: Sketch-based Generation Pre-training Strategies and Evaluation

3.1 Comparison of Different Strategies for GENIUS's Pre-training

In Section 2.1 we introduce our default sketch design for GENIUS's pre-training. Now we compare different possible sketch templates for pre-training: T_1 is a simple concatenation of the key elements joined by spaces, ordered by their importance given by the extractor. T_2 sorts the elements by their original order, and T_3 allows multiple occurrences and overlaps of key elements as in the original text. Our default sketch template described before is named T_4, which further replaces each of the missing parts with a single mask token. We also compare with T_4-*random*, which extracts random n-grams for sketch construction as the only difference with T_4. Table 2 shows examples of these templates.

We pre-train GENIUS with these different sketch templates and evaluate their reconstruction performance on the dev set by ROUGE-1/2/L scores. The overall performance is ordered by T_4-*random* $< T_1 < T_2 < T_3 < T_4$, which illustrates that our choice of T_4 helps GENIUS learn better during pre-training. Comparison between T_2 and T_1 shows the benefits of using the original order; comparison between T_3 and T_2 shows the benefits of keeping the original occurrences; comparison between T_4 and T_3 shows the importance of being consistent with BART by filling the missing parts with mask tokens; T_4-*random* also possess these characteristics, but the random masking strategy makes model hard to learn in this high-masking setting, resulting in the lowest ROUGE scores among these choices. Therefore, we use T_4 as the default sketch template in this work.

passage: **NLP** is a branch of **computer science**—and more specifically, a **branch of AI**. **NLP** is widely used in our lives.
keywords/ key-phrases *(sorted by importance)*: [NLP, branch of AI, computer science]
T_1: NLP branch of AI computer science
T_2: NLP computer science branch of AI
T_3: NLP computer science branch of AI NLP
T_4 *(default)*: NLP `[M]` computer science `[M]` branch of AI `[M]` NLP `[M]`
T_4*-random (extract random n-grams)*: `[M]` a branch `[M]` science `[M]` more specifically `[M]`

Template	ROUGE-1	ROUGE-2	ROUGE-L
T_1	28.32	16.90	24.05
T_2	28.56	17.42	26.41
T_3	28.70	17.31	26.52
T_4 *(default)*	**28.77**	**17.89**	**26.74**
T_4*-random*	17.82	16.41	17.78

Table 2: Comparison of different sketch templates for GENIUS's pre-raining. `[M]` is the mask token.

3.2 Quality Evaluation and Comparison for Sketch-based Text Generation

Previous studies like blank infilling task (Shen et al., 2020) mainly evaluate how the reconstructed text *resembles* the original one though CER (Morris et al., 2004) or BLEU (Papineni et al., 2002) scores, the sketch-based text generation, however, is different since it favors more *diversity* and there isn't a ground-truth to a sketch due to its high masking ratio. Therefore, we mainly evaluate how well the model can join the key elements in a fluent and informative way. We extract 1000 topic-related sketches from the HuffPost news dataset (Misra and Grover, 2021) from five topics (politics, sports, entertainment, tech, and business) using the same approach in Section 2.2, and evaluate the sketch-based generation using the following metrics: **perplexity** (Jelinek et al., 1977) given by GPT-2 to measure the text fluency; **clf-error**, inspired by the Inception Score (Barratt and Sharma, 2018) used for image generation evaluation, we trained a BERT-based classifier on 50K HuffPost news from the same topics as a scorer to measure how well the generated text can represent the original topic. The classification error is reported for this metric; **sketch-lost** which measures how the original sketch is retained in the generated text, calculated by the average of both word-level and fragment-level missing percentage; **recall** measures how much n-grams from the original text are restored in the generated text. We report the average recall of unigram, bigram, and the longest common subsequence levels; **diversity** is calculated by the percentage of new words introduced in the generated samples compared with the original samples. We also report the relative **length** of the generated text compared with the original text.

We compare with the following existing models: **BART-large-infill** directly utilized the text-infilling ability of BART-large (Lewis et al., 2020) for generation; **T5-CommonGen** (Gehrmann et al., 2021) trains the T5 (Raffel et al., 2019) on the CommonGen (Lin et al., 2020) dataset for keywords-to-text generation; **CBART-large** (He, 2021) is a lexically constrained generation model that can gradually generate complete sentences given some keywords; **ILM-ngram** (Donahue et al., 2020) is the ngram infilling version of **ILM** which formulates the training sequences into a special blank-infilling structure and trains the GPT-2 model to fill the blanks. We also evaluate the performances of GENIUS using different sketch templates, denoted as **GENIUS-Tx-base**. Considering the superiority of T_4 discussed in Section 3.1, we pre-train a larger version using T_4 initialized with BART-large with longer training sequences, denoted as **GENIUS-T4-large**, which is also the default version of GENIUS in this work. Note that all models including GENIUS are not further fine-tuned on the HuffPost dataset. We use the publicly available model checkpoints of previous methods for evaluation.

According to the results in Table 3, GENIUS achieves the lowest perplexity, recall and clf-error compared with others. This means GENIUS can generate more fluent and semantic-preserving samples than other methods. These extracted sketches from the 5-class HuffPost dataset can be viewed as a corrupted training set where only some key information remains. GENIUS helps to reconstruct the corrupted training set and achieves the lowest classification error. The generated samples from GENIUS are also more diverse than other methods, except ILM-ngram. However, ILM-ngram has the highest perplexity and high clf-error. ILM-ngram is only pre-trained on 100K book texts, which partly explains the poor performance on a new domain. ILM uses a 15% masking ratio during training, which also limits its ability in this high-masking reconstruction task. T5-CommonGen and CBART are limited to a few unigrams as input, and the output is limited to short sentences, resulting in high

	Perplexity(↓)	Clf-error(↓)	Sketch-lost(↓)	Recall(↑)	Diversity(↑)	Length
raw sketch	218.22	9.32	0.00	20.21	0.00	0.25
BART-large-infill (Lewis et al., 2020)	56.04	7.49	0.96	22.41	5.37	0.37
T5-CommonGen (Gehrmann et al., 2021)	43.24	12.73	41.57	15.06	8.48	0.43
CBART-large (He, 2021)	50.58	10.08	7.88	22.32	8.07	0.56
ILM-ngram (Donahue et al., 2020)	90.25	12.2	**0.03**	24.71	**30.58**	1.47
GENIUS-T1-base	26.13	8.97	10.66	24.71	8.59	0.73
GENIUS-T2-base	27.63	9.08	7.43	22.77	6.55	0.56
GENIUS-T3-base	24.95	8.20	1.99	24.71	9.16	0.72
GENIUS-T4-base	19.32	8.12	0.83	25.35	9.23	0.77
GENIUS (GENIUS-T4-large)	**18.09**	**7.09**	0.69	**29.51**	21.18	1.71

Table 3: Evaluation for the sketch-based text generation using different methods. All models are not fine-tuned on the evaluation dataset. GENIUS generates more fluent and semantic-preserving text than other methods while also with high diversity.

English Task:	
sketch	[M] **machine learning** [M] **my research interest** [M] **data science** [M]
BART-large-infill	The **machine learning** aspect of **my research interest** in **data science**.
T5-CommonGen	**my interest** in **machine learning** is based on **data** from my **research** .
CBART-large	Using **machine learning** and **my research** , I developed a deep **interest** in **data science** , which I wanted to pursue .
ILM-ngram	Big **machine learning** My research agenda for the summer is to **my research interest** learn about the atomic bomb. **data science** This summer I went to the math lab to study.
GENIUS	I am a Ph.D. student in **machine learning**, and **my research interest** is in **data science**. I am interested in understanding how humans and machines interact and how we can improve the quality of life for people around the world.

Chinese Task:	
sketch	[M] **酸菜鱼火锅** [M] **很美味，味道绝了** [M] **周末真开心** [M]
BART-chinese	这**酸菜鱼火锅**真的**很美味，味道绝了**这**周末真开心**啊
GENIUS-chinese	今天吃了**酸菜鱼火锅**，真的**很美味，味道绝了**，不愧是我的第一次尝试，这个**周末真开心**，不知道大家有没有吃过呢，一起来看看吧。

Figure 2: Examples of different methods for sketch-based text generation. We illustrate both the English (default) and Chinese versions of GENIUS. The **bold** represents the key fragments from the sketch, while the blue represents the newly generated contexts.

perplexity and clf-score when inputting sketches which may consist of multi-granularity elements. The comparison of different GENIUS-Tx-base models echos the results in Section 3.1 that our default sketch template T_4 is superior to other templates, with better scores in all metrics. GENIUS-T4-large further improves the fluency and diversity by using larger model weights and longer training sequences.

Figure 2 shows some generated examples by different models, for both English and Chinese tasks. One noticeable difference between GENIUS and previous methods is that GENIUS can generate longer text with more details. This feature is inherited from GENIUS's extreme-masking pre-training where the model is asked to reconstruct large parts of the original text. ILM also generates longer text but is quite in-fluent. T5-CommonGen and CBART can generate relatively fluent text but may destroy the structure of the input sketch.

4 Experiments: Data Augmentation for Various NLP Tasks with GeniusAug

In this section, we will show that our proposed GeniusAug method can effectively use sketch-based text generation for data augmentation, improving the downstream model performances for various NLP tasks, including text classification, NER, and MRC.

4.1 Text Classification

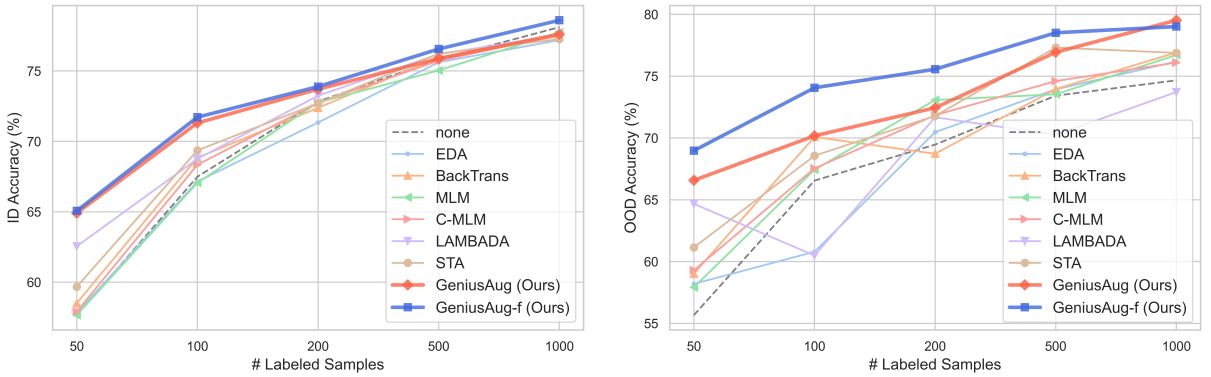
4.1.1 Setup

Datasets. We conduct experiments on 6 widely used datasets, including four topic classification datasets **BBC** (Greene and Cunningham, 2006), **Huff**[3] (Misra and Grover, 2021), **Yahoo** (Zhang et al., 2015), **20NG**, and two sentiment classification datasets **SST2** (Socher et al.,

[3]The original Huff dataset contains 41 categories. To facilitate the ID and OOD comparison, we only choose 5 categories that are the same as the BBC dataset.

	ID evaluation							*OOD evaluation*				
Method	**Huff**	**BBC**	**Yahoo**	**20NG**	**IMDB**	**SST2**	*avg.*	**H⇒B**	**B⇒H**	**I⇒S**	**S⇒I**	*avg.*
none	79.17	**96.16**	45.77	46.67	77.87	76.67	*70.39*	62.32	62.00	74.37	73.11	*67.95*
EDA (2019)	79.20	95.11	45.10	46.15	77.88	75.52	*69.83*	67.48	58.92	75.83	69.42	*67.91*
BackTrans (2018)	80.48	95.28	46.10	46.61	78.35	76.96	*70.63*	67.75	63.10	75.91	72.19	*69.74*
MLM (2020)	80.04	96.07	45.35	46.53	75.73	76.61	*70.06*	66.80	65.39	73.66	73.06	*69.73*
C-MLM (2020) *	80.60	96.13	45.40	46.36	77.31	76.91	*70.45*	64.94	**67.80**	74.98	71.78	*69.87*
LAMBADA (2020) *	81.46	93.74	**50.49**	47.72	78.22	78.31	*71.66*	68.57	52.79	75.24	76.04	*68.16*
STA (2022)	80.74	95.64	46.96	47.27	77.88	77.80	*71.05*	71.39	64.82	74.72	73.62	*71.13*
GeniusAug (Ours)	81.43	95.74	49.60	50.38	**80.16**	78.82	*72.68*	74.87	66.85	76.02	74.76	*73.13*
GeniusAug-*f* (Ours) *	**81.82**	95.99	50.42	**50.81**	79.40	**80.57**	***73.17***	**76.18**	66.89	**77.45**	**80.36**	***75.22***

Table 4: In-distribution (ID) and out-of-distribution (OOD) evaluations of different augmentation methods, where H, B, I, and S stand for BBC, Huff, IMDB, and SST2 respectively. The stared methods (*) need fine-tuning on the downstream tasks. Underline means significant improvements over the *none* baseline with paired student's t-test, $p < 0.05$.



(a) ID (b) OOD

Figure 3: Augmentation effectiveness when given different numbers of labeled samples in ID and OOD settings. We plot the averaged scores from all datasets.

2013), **IMDB** (Maas et al., 2011). We experiment on a low-resource setting where $n = \{50, 100, 200, 500, 1000\}$ train/dev samples are randomly selected from the original train/dev sets of these datasets in our experiments. We use the original full test sets of these datasets for evaluation, which we call in-distribution (**ID**) evaluation. To further evaluate the model's generalization ability, we also design 4 groups of out-of-distribution (**OOD**) generalization tasks between the two movie review sentiment classification tasks – IMDB and SST2, and the two news classification tasks – BBC and Huff, following the experimental design in (Hendrycks et al., 2020), where the model is first trained in one dataset and then directly evaluated on the other dataset without additional fine-tuning.

Baselines. The following augmentation methods are compared: rule-based **EDA** (Wei and Zou, 2019) and **STA** (Guo et al., 2022); **BackTrans** (Silfverberg et al., 2017) uses the translation models from (Tiedemann, 2020) of four languages (de/ru/es/zh) in our experiments; **MLM** utilizes the masked language modeling (MLM) for words replacement; **C-MLM** (Kumar et al., 2020) further fine-tunes a conditional MLM model by prepending the label to each sequence during MLM training. Note that **MLM** and **C-MLM** use BERT-base in their original work, while we use the stronger RoBERTa-large in our experiments; **LAMBADA** (Anaby-Tavor et al., 2020) fin-tunes a conditional GPT-2 model to generate new samples by giving labels as prompts. Apart from directly using **GeniusAug** for data augmentation, we also compare with a fine-tuned version **GeniusAug-f** which is further fine-tuned on the downstream training set, where the model learns to reconstruct the original training sample given the target-aware sketch. With each augmentation method, we scale up the training set to 2-5 times the original size and select the best model on the dev set for evaluation. All the augmentation methods are based on the same base text classifier using the same training hyper-parameters and model selection criteria (see Appendix A.2). In the main experiments, the base classifier is

DistilBERT-base (Sanh et al., 2019), which is an efficient lightweight Transformer model distilled from BERT. We also experiment on the stronger *RoBERTa-large* (Liu et al., 2019b) classifier, discussed in the later part.

4.1.2 Results

Table 4 reports the averaged scores of $n = \{50, 100, 200, 500, 1000\}$ in both ID and OOD settings. The performances at each training size are shown in Figure 3.

In-distribution evaluations. Our proposed GeniusAug and GeniusAug-f both boosts the performance of base classifiers in the ID evaluations, with average improvements of around 2% and 3% respectively. GeniusAug and GeniusAug-f also outperform other baselines in most experiments by large margins. Among the baselines, rule-based STA and conditional generation-based LAMBADA are competitive methods, while other approaches bring only marginal gains or even degradation. Benefiting from a word roles recognition process and selective augmentation manner, STA can prevent the core semantics from being changed during augmentation while also introducing small perturbations to the original samples. However, rule-based operations may result in unnatural sentences and also limit the diversity. The strong reconstruction ability makes GeniusAug superior to STA by generating more fluent and diverse samples. LAMBADA can learn to generate diverse and coherent samples that belong to certain categories. However, since the generation is only conditioned on a label, the semantics of the generated text are more likely to be skewed. In comparison, GeniusAug aims to generate more complementary contexts for a given sketch, thus better guaranteeing the quality of the generation.

Out-of-distribution evaluations. OOD generalization is more challenging than ID evaluation since it requires the model to generalize to unknown distribution(s). In this setting, GeniusAug and GeniusAug-f bring much higher gains over non-augmentation baselines than in the ID evaluations, with average improvements of around 5% and 7% respectively. STA performs the best among the baselines thanks to its selective operations to protect core words and remove potential noise, which are helpful for generalization. LAMBADA exhibits severe degradation in the BBC⇒Huff generalization task. By checking the performances of LAMBADA at all training sizes, we find the

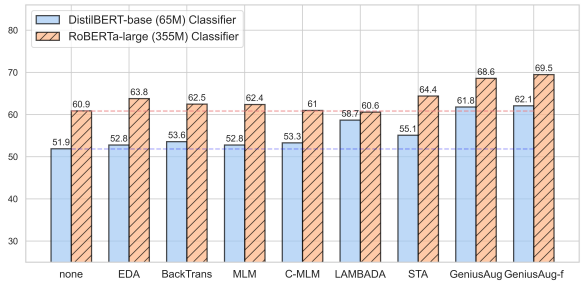


Figure 4: Performances on different classifiers – DistilBERT-base and a stronger RoBERTa-large. The results are averaged across all datasets on a $n = 50$ setting.

OOD generalization performance of LAMBADA is even getting worse when more training data is provided. This phenomenon indicates that LAMBADA may have learned the dataset bias during fine-tuning on the source dataset, which harms the OOD generalization ability. The nature of GeniusAug and GeniusAug-f makes them very suitable for the OOD generalization: By masking the unimportant parts in the original text, GeniusAug prevents the potential noise to be enhanced during generation; by keeping the key parts unchanged, the core semantics of the target is preserved during generation, reducing the risk of semantic drift.

Performance at the different resource levels. Figure 3 shows the performances of different methods with different numbers of labeled training samples. In the severe low-resource scenarios ($n = \{50, 100\}$), GeniusAug and GeniusAug-f are significantly stronger than other baselines. With labeled data getting richer, the gap between these methods is getting smaller in the ID setting, but GeniusAug and GeniusAug-f continue to maintain huge advantages over most baselines in the OOD setting.

Effectiveness of fine-tuning. Compared with GeniusAug, GeniusAug-f achieves better results in most datasets. During GENIUS's pre-training process, the model inputs are *general* sketches, while in the fine-tuning procedure we are using *target-aware* sketches, this helps the GENIUS to generate more target-related contents.

Makes strong classifier stronger. In the above experiments, we use DistilBERT-base for the classifier, which has around 65 million parameters. Here we also evaluate the augmentation effectiveness using a much stronger RoBERTa-large classifier, with more than 355 million parameters. Due to the limited time/resources, we currently only experiment with $n = 50$ for

Dataset	*none*	*pre-sota*	*GA*	*GA-f*	***GA-mixup***
Huff	71.44	76.72	78.53	77.99	**78.74**
BBC	94.94	95.60	94.90	**95.30**	95.14
IMDB	58.78	62.57	68.74	65.54	**69.58**
SST2	67.22	75.11	73.65	**76.40**	71.42
Yahoo	29.81	44.13	40.23	44.02	**45.50**
20NG	23.78	26.06	33.42	31.15	**38.77**
avg.	57.66	63.37	64.91	65.07	**66.53**

Table 5: GA denotes GeniusAug. GeniusAug-mixup combines the key information from various samples to boost the diversity of synthetic data. The experiments are conducted at $n = 50$ for each dataset.

each dataset, as shown in Figure 4. Surprisingly, GeniusAug/GeniusAug-f *helps the weaker DistilBERT-base classifiers to outperform the strong RoBERTa-large classifiers* (comparing the orange bar of "none" and blue bars of "GeniusAug" and "GeniusAug-f"). GeniusAug/GeniusAug-f also makes the RoBERTa-large classifiers stronger, improving their average accuracy scores by 8-9% and outperforming other baselines by at least 4%.

Boosting the diversity via *Sketch Mixup*. In the previous experiments, each sketch is extracted from a single training sample, which somehow is limited in diversity. Inspired by the Mixup (Zhang et al., 2018), we propose **GeniusAug-mixup** to generate samples based on the mixed-up sketch that *combines the target-related parts from multiple training samples*. By doing so, the generated samples exhibit larger differences from the original samples, while still being label-preserving. Experiments on all the datasets ($n = 50$) show that GeniusAug-mixup can further boost the performance for most experimented tasks. Note that GeniusAug-mixup doesn't need a fine-tuning step. There are other possible ways to improve the diversity of GeniusAug-generated dataset, such as applying synonyms replacement or changing the element order on the sketches, which we will explore in future work.

4.2 Augmentation for Other NLP Tasks

4.2.1 Setup

We use the **CoNLL03** (Sang and Meulder, 2003) dataset for the named entity recognition (NER) task and **SQuAD** (Rajpurkar et al., 2016) for the machine reading comprehension (MRC) task. We sample the *first* $n \in \{50, 100, 200, 500\}$ labeled samples from the original datasets for our experiments. We use CoNLL03's original full test set for evaluation. For SQuAD, since the test set is not publicly available, we report the results on the development set. We report the F1 score for CoNLL03 and the exact match (EM) score for SQuAD. The base NER/MRC models are all based on *BERT-base*.

For NER, Synonyms Replacement (**SR**), Mention Replacement (**MR**) (Dai and Adel, 2020), **Rule-mix**(Dai and Adel, 2020), and **MELM** (Zhou et al., 2022b) are used as baselines. For MRC, **SR** and **BackTrans** are used. The other settings are the same as text classification tasks.

NER (CoNLL03)					
n	50	100	200	500	*avg.*
non-aug	39.28	54.97	63.88	73.78	57.98
SR (2019)	41.15	56.79	61.90	73.98	58.46
MR (2020)	47.75	58.85	61.76	73.15	60.38
Mix-rule (2020)	45.78	55.56	61.60	72.55	58.87
MELM (2022b)	46.45	53.05	60.96	**77.36**	59.46
GeniusAug *(Ours)*	**49.17**	**61.12**	**66.10**	74.69	**62.77**
MRC (SQuAD)					
n	50	100	200	500	*avg.*
non-aug	15.74	21.67	31.19	47.98	29.15
SR (2019)	18.45	25.35	35.98	50.86	32.66
BackTrans (2018)	**19.26**	26.13	36.21	50.31	32.98
GeniusAug *(Ours)*	19.03	**28.60**	**37.02**	**51.83**	**34.12**

Table 6: Data augmentation for the NER and MRC tasks with different labeled sizes.

4.2.2 Results

Named entity recognition. The upper part of Table 6 summarizes the comparison for the CoNLL03 task with different sizes of labeled data. When the number of labeled data $n \in \{50, 100, 200\}$, GeniusAug outperforms all the baselines by a large margin. When n reaches 500, GeniusAug outperforms previous methods, except MELM. The rule-based methods SR, MR, and Mix-rule can improve the recognition performance when the labeled data size is small ($n \in \{50, 100\}$). However, these methods may generate non-fluent sequences, or make the lexical pattern around entities unnatural, which explains why these methods harm the performance when their training size becomes larger ($n \in \{200, 500\}$). MELM utilizes a novel masked entity language modeling task to train the model for predicting new entities, which achieves the best F1 score when $n = 500$. However, when the training size is small, MELM may not be able to train a satisfactory language model for generating proper new entities, which degrades the performance ($n \in \{100, 200\}$). Compared with rule-based methods, GeniusAug can generate more di-

verse and natural text, which is helpful for the NER model to learn new patterns for entity recognition. Compared with MELM, GeniusAug doesn't rely on a fine-tuning process and thus can achieve better results when the training size is extremely small. A major difference between GeniusAug and MELM is that GeniusAug aims to diversify the context while MELM focuses on introducing more entities. We will explore the combination of GeniusAug and MELM in future work to further improve the performance.

Machine reading comprehension. The lower part of Table 6 shows the exact match (EM) scores for SQuAD in different resource levels. GeniusAug achieves the best results when $n = \{100, 200, 500\}$ and comparable scores with BackTrans when $n = 50$. The two baselines SR and BackTrans are both effective augmentation methods among all training sizes. Compared with the baselines, GeniusAug brings more diversity to the context around the answer, which helps improve the understanding ability of MRC models. There are also question data augmentation (QDA) methods specifically designed and pre-trained for MRC and QA tasks (Alberti et al., 2019; Liu et al., 2020). We don't compare these methods in our experiments, since they need relatively large data for pre-training, which are not applicable in our low-resource setting. In addition, GeniusAug is orthogonal to these QDA methods: GeniusAug focuses on augmenting the context around the current answer, while QDA methods aim to generate new questions for the current context. Therefore, QDA methods can be combined with GeniusAug to generate more diverse training samples for MRC and QA tasks.

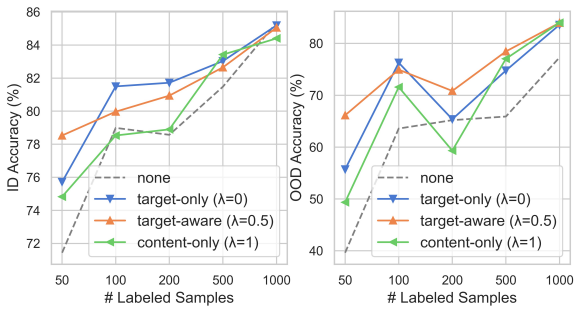


Figure 5: Comparison of different sketch extraction strategies on the Huff dataset with different labeled sizes.

ID (Huff)						
n	50	100	200	500	1000	avg.
none	71.44	78.99	78.57	81.48	85.35	79.17
GeniusAug	**78.53**	79.97	80.94	82.65	85.05	81.43
– *PT*	72.92	76.73	79.89	81.82	83.94	79.06
GeniusAug-f	77.99	**81.01**	81.00	**83.14**	85.94	**81.82**
– *PT*	69.85	77.32	79.87	81.61	**85.98**	78.93
OOD (Huff⇒BBC)						
n	50	100	200	500	1000	avg.
none	39.62	63.58	65.20	65.90	77.30	62.32
GeniusAug	66.14	74.94	70.84	78.46	**83.98**	74.87
– *PT*	40.70	64.28	62.38	72.22	78.58	63.63
GeniusAug-f	**66.70**	**76.18**	**77.24**	**81.24**	79.56	**76.18**
– *PT*	42.40	63.06	62.00	68.18	78.80	62.89

Table 7: Ablation study on the effectiveness of GENIUS's large-scale pre-training for data augmentation

4.2.3 Ablation Study

GENIUS pre-training. GENIUS is pre-trained using a novel *reconstruction from sketch* objective, which enables the model to reconstruct sentences or paragraphs given only a few segments. The backbone of GENIUS – BART (Lewis et al., 2020) can also be used to reconstruct text where certain short spans (ranging from 1-3 words) are masked. Therefore, we can verify the effectiveness of GENIUS's pre-training by directly using BART-large's infilling ability for our proposed sketch-based generation or sketch-based fine-tuning, without the GENIUS-like pre-training, denoted as "– PT". We experiment on the Huff dataset and Huff⇒BBC task as shown in Table 7. Compared with GeniusAug or GeniusAug-f, using BART for sketch-based generation achieves significantly worse results, especially in the OOD or low-resource setting. The pre-training of BART determines it can only reconstruct small textual corruption. In comparison, GeniusAug uses a much larger masking ratio during pre-training, resulting in the strong ability to generate complete and coherent context around the sketch. These results show the importance of our proposed *reconstruction from sketch* pre-training.

Target-aware sketch extraction. Recall that our target-aware sketch extraction uses a fused embedding from both the document and the target label, by using a hyper-parameter $\lambda = 0.5$ to balance the information from both sides. Now we compare the *content-only* and *task-only* strategies by setting $\lambda = 1$ and $\lambda = 0$ respectively. Figure 5 shows the comparison of these different strategies

on the Huff dataset with different training sizes. Results reveal that target-aware is more robust than the other two strategies in both ID and OOD scenarios, while content-only is relatively the worst across all settings. Extracting the sketches based only on the content may lose some key information related to downstream tasks, while based only on the target may result in a lack of diversity. It is beneficial to consider both sides for sketch construction, which shows the effectiveness of our proposed target-aware sketch extraction. Note that $\lambda = 0.5$ may not be optimal for every dataset therefore we encourage tuning this hyper-parameter according to the specific task.

5 Related Work

5.1 Reconstruction for Corrupted Text

Many recent pretrained transformer models (PTMs) are built on denoising pre-training, with text reconstruction as (one of) their pre-training objective(s). BERT (Devlin et al., 2019) first uses a masked language modeling (MLM) task in pre-training, which masks 15% of the tokens in the original text and asks the model to predict the missing tokens. The MLM task and the 15% masking ratio are also used by BERT's successors, like RoBERTa (Liu et al., 2019b) and ALBERT (Lan et al., 2019). BART (Lewis et al., 2020) uses a novel text infilling task with a bigger masking ratio of 30% for pre-training, where small spans of text are corrupted for reconstruction. MASS (Song et al., 2019) uses a higher masking ratio (50%) for seq2seq pre-training, but is limited to sentence level with only one consecutive fragment being masked for each sentence, which makes MASS unable to reconstruct the text based on a sketch that may have several blanks. GLM (Du et al., 2022) pre-trains a self-attention Transformer using an autoregressive blank infilling objective, also with a 15% random masking. However, GLM predicts the masked spans in an iterative manner where the current prediction should be based on previously predicted spans, limiting the efficiency of sketch-based text generation. To the best of our knowledge, GENIUS is the first language model pre-trained with *extreme and selective masking* on the large-scale general corpus.

Our work is also related to text-infilling methods, including Zhu et al. (2019)'s, TIGS (Liu et al., 2019a), blank language models (Shen et al., 2020), ILM (Donahue et al., 2020), and lexically constrained generation methods like POINTER (Zhang et al., 2020) and CBART (He, 2021). Keywords-to-text methods like T5-CommonGen (Gehrmann et al., 2021) also shares some similarity. However, these methods are not designed for the sketch-based text generation task defined in this work. They are not suitable for this task for *at least one* of the following reasons: 1) limited to unigrams as input; 2) mainly trained for short-sentence reconstruction; 3) fixed blank-filling length; 3) traditional model architecture without pre-training; 4) no pre-training on general corpus for public use. These limitations will result in high sketch-lost or in-fluency of the generated text. We compare with some of the typical methods in Section 3, which shows the superiority of our proposed GENIUS.

Recently Zeng et al. (2022) released a giant pre-trained model – GLM-130B, containing 130 billion parameters. GLM-130B also has strong abilities for blank-filling tasks. Due to the limited time and resources, we didn't compare with it in the experiments part. In Appendix B.1, we show some examples generated by GLM-130B by calling their online API and analyze the advantages and disadvantages of both GENIUS and GLM-130B.

5.2 Data Augmentation in NLP

Data augmentation techniques are extensively studied in all kinds of NLP tasks. Most of these methods conduct augmentation on the input space by generating new training samples (Feng et al., 2021), while some other methods are applied to the feature space, such as embedding mixup (Guo et al., 2019; Sun et al., 2020; Chen et al., 2020). In this work, we mainly focus on the input space data augmentation. Rule-based methods are widely used for various tasks, including text classification tasks (Wei and Zou, 2019; Guo et al., 2022), named entity recognition (Dai and Adel, 2020) and natural language generation (Feng et al., 2020). Apart from the rule-based methods, there are also plenty of generative model-based augmentation methods, such as back-translation (Sennrich et al., 2016; Yu et al., 2018), paraphrasing (Gao et al., 2020; Damodaran, 2021), contextual augmentation (Wu et al., 2019; Kumar et al., 2020) that utilize the masking mechanism in masked language models (MLM) and open-ended generation methods (Anaby-Tavor et al., 2020; Kumar et al., 2020) that use auto-regressive (AR) models like GPT-2 (Radford et al., 2019) to generate new text by fine-tuning the model to generate the original text conditioned on the label.

Our proposed GeniusAug is also on the line of generative augmentation methods, but is different from previous generative methods in the following ways: Compared with methods like MLM-based or paraphrasing methods, GeniusAug introduces more diversity to the training set; Compared with AR-based open generation methods, GeniusAug is more controllable in the content and quality of generated samples; Methods like C-MLM (Wu et al., 2019; Kumar et al., 2020), LAMBADA (Anaby-Tavor et al., 2020) and MELM (Zhou et al., 2022b) also involve an extra fine-tuning step on the downstream datasets, making them more inconvenient for deployment, while GeniusAug can be directly used for augmentation (though we also show fine-tuning can lead to further performance gains); Last but not least, GeniusAug is a more general approach applicable to a variety of NLP tasks while most of the traditional methods are task-specific.

6 Discussions & Limitations

6.1 GENIUS Model.

In this work, we present the sketch-based text generation task, a pre-trained model GENIUS specifically designed for this task, and a novel GeniusAug method that effectively applies sketch-based text generation to data augmentation scenarios. Sketch-based text generation task might be a new task for previous methods, which partly explains why they are inferior to GENIUS in our experiments. Therefore, we don't expect GENIUS to be the state-of-the-art (SOTA) model for all conditional language generation tasks.

Our current research on GENIUS also has some limitations, which are under our future research. For example, using more data and a larger backbone model can hopefully further improve the performance for more general use; the length of the generated text for each masked place cannot be controlled, limiting the usage for precise controlling; more experiments and analysis of using different pre-training strategies (such as the strategy used in ILM (Donahue et al., 2020)) should be studied.

6.2 GeniusAug.

Though GeniusAug shows advantages over a bunch of previous augmentation methods in a variety of NLP tasks, we don't pursue making GeniusAug the go-to augmentation tool for each specific task. The following limitations should be considered when using GeniusAug for data augmentation:

1) Compared with rule-based methods, GeniusAug needs more computation resources since it is based on a large pre-trained language model. We also release lighter versions trained on BART-base, but we haven't evaluated them on augmentation tasks; 2) GeniusAug's sketch extraction process is less compatible with tasks that require logical reasoning, such as natural language inference (NLI). This is because the extracted sketches may not be a good representation of the logical relationships contained in the text, which may result in noisy samples. For these tasks, we recommend utilizing the in-context learning of GPT-3 (Brown et al., 2020), or more fine-grained methods like FlipDA (Zhou et al., 2022a); 3) When using GeniusAug for NER data augmentation, new entities may be generated, which may lead to the unlabeled entity problem. To tackle this issue, an extra filtering step should be involved or utilize a modified loss function as described in Section A.3.

Therefore, we encourage combining different augmentation methods according to the characteristics of the downstream tasks.

7 Conclusion

In this paper, we present a sketch-based text generative model called GENIUS, which is pre-trained on a large-scale corpus with a novel extreme-and-selective masking strategy. Based on GENIUS, we propose a novel textual data augmentation method named GeniusAug which can generate diverse and high-quality texts given textual sketches extracted from the training set. GeniusAug can be directly applied to various NLP tasks without further fine-tuning on downstream datasets. Extensive experiments reveal the strong performance of GeniusAug on classification, NER, and MRC tasks. In future work, we will further explore other applications of GENIUS and study the potential of using extreme-and-selective masking strategy to further improve current pre-trained language models.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Shane Barratt and Rishi Sharma. 2018. A note on the inception score. *arXiv preprint arXiv:1801.01973*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Prithiviraj Damodaran. 2021. Parrot: Paraphrase generation for nlu.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. Association for Computational Linguistics.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press.

Biyang Guo, Songqiao Han, and Hailiang Huang. 2022. Selective text augmentation with word roles for low-resource text classification. *arXiv preprint arXiv:2209.01560*.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Xingwei He. 2021. Parallel refinements for lexically constrained text generation with bart. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8653–8666.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution

robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yangming Li, Shuming Shi, et al. 2020. Empirical analysis of unlabeled entity problem in named entity recognition. In *International Conference on Learning Representations*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. Commongen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840.

Dayiheng Liu, Jie Fu, Pengfei Liu, and Jiancheng Lv. 2019a. Tigs: An inference algorithm for text infilling with gradient search. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4146–4156.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Jiancheng Lv, Nan Duan, and Ming Zhou. 2020. Tell me how to ask again: Question data augmentation with controllable rewriting in continuous space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5798–5810.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Rishabh Misra and Jigyasa Grover. 2021. *Sculpting Data for ML: The first act of Machine Learning*.

Andrew Morris, Viktoria Maier, and Phil Green. 2004. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: language-independent named entity recognition. *conference on computational natural language learning*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020. Blank language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5186–5198.

Yong-Siang Shih, Wei-Cheng Chang, and Yiming Yang. 2019. Xl-editor: Post-editing sentences with xlnet. *ArXiv*, abs/1910.10479.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*.

Jörg Tiedemann. 2020. The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pages 84–95. Springer.

Bright Xu. 2019. Nlp chinese corpus: Large scale chinese corpus for nlp.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and William B Dolan. 2020. Pointer: Constrained progressive text generation via insertion-based generative pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8649–8670.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022a. Flipda: Effective and robust data augmentation for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8646–8665.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022b. Melm: Data augmentation with masked entity language modeling for low-resource ner. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262.

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text infilling. *arXiv preprint arXiv:1901.00158*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Experiment Details

A.1 GENIUS Pre-training

To extract the sketch of a given document, we utilize YAKE (Campos et al., 2020), a lightweight unsupervised automatic keyword extraction method to select the most relevant keywords. We set `max_ngram=3` and `topk=max(l/5,10)` where `l` is the length of the document (number of words). Then a projection and a masking process are applied to obtain the sketch of the document. We sampled 27 million paragraphs (with less than 15 sentences in each paragraph for the base model and length ranging from 50 to 200 for the large model) as the training set from the `realnewslike` split of C4 dataset (Raffel et al., 2019). Note that `topk=l/5` doesn't mean the masking ratio is 80% (4/5), since the keywords may occur multiple times in the document or be contained within other keywords. We calculated the masking ratio of 1 million <sketch,text> pairs randomly sampled from our training set, the average proportion (%) is 72.97 $\pm$ 7.05.

GENIUS is a seq2seq model, with a bidirectional encoder and an auto-regressive decoder. GENIUS uses the same structure of BART (Lewis et al., 2020) and is initialized with the weights of BART-base or BART-large for GENIUS-base and GENIUS-large respectively. GENIUS is optimized using AdamW (Loshchilov and Hutter, 2017) optimizer with learning rate 5.6e-5 and weight decay 0.01. We pre-train the GENIUS model for 3 epochs with batch size 32 using 8 NVIDIA V100 cards which takes a few days for the base model and around a week for the large model.

The Chinese version GENIUS-base-chinese is trained on BART-base-chinese (Shao et al., 2021), using 10 million passages from the CLUE corpus (Xu, 2019) for pre-training. A multilingual version is also under development.

All the code and models will be publicly available at https://github.com/beyondguo/genius and https://github.com/microsoft/SCGLab.

A.2 Datasets & Model Settings

For text classification, the training and validation sets are randomly sampled from the original datasets with $n \in \{50, 100, 200, 500, 1000\}$. We use the AdamW (Loshchilov and Hutter, 2017) optimizer with learning rate 5e-5 for training and use early-stopping with patience=10 to choose the best model. We run all experiments with 5 random seeds and report the average performance.

For named entity recognition, we use the classical CoNLL03 dataset (Sang and Meulder, 2003) for evaluation. Notice that the consecutive sequences of the CoNLL03 dataset may come from the same article, thus may share some common entities. Previous works usually randomly sample n sequences from the dataset, which may come from up to n different articles, resulting in a sampled dataset with plenty of diverse entities. We claim that this approach does not correspond to a true low-resource scenario for NER tasks. Therefore, in this work, instead of randomly choosing n samples from the dataset, we use the *first-n* samples from the dataset to construct training and validation sets, which are consecutive sequences coming from m articles ($m << n$). We experiment on $n \in \{50, 100, 200, 500\}$ for this task. We use BERT-base(Devlin et al., 2019) as the NER model, use AdamW optimizer with learning rate 2e-5 and linear learning rate scheduling for training. For each augmentation method, we train the model for 40 epochs and choose the best model according to the F1 score on the validation set.

For machine reading comprehension, we use the widely used SQuAD (Rajpurkar et al., 2016) dataset. The test set of SQuAD is not publicly available. To evaluate the performance in the test set, we need to corporate with the authors of SQuAD leaderboard. Since we mainly focus on the augmentation for low-resource setting in this paper where only a very small fraction of training data is used, we choose to only evaluate our method on the public development set, and use the same set of hyper-parameters for fair comparison among all the baselines. We use BERT-base as the basic model and the scripts provided by Huggingface[4] for model training and evaluating.

A.3 Data Pre-Processing for Different NLP Tasks

Text classification. For text classification tasks, we use the categories of the samples as the TRI for target-aware sketches extraction during the augmentation of GENIUS. We use the **a**ttribute-**c**ontrolling (see Section 2.2) to make the generated samples closer to their corresponding labels. Specifically, we add the label text as the prompt

[4] https://github.com/huggingface/transformers/tree/main/examples/pytorch/question-answering

before the sketch, joined by ":" or "</s>" in between.

NER. For NER, we use the entities in the training sequences as the TRI for target-aware sketch extraction, by which the extracted key-phrases are usually the textual spans that contain or semantically similar to the entities. The samples from the CoNLL03 dataset are usually short sentences, which are much shorter than the paragraphs used for GENIUS pre-training. Therefore, we concatenate the consecutive sequences into longer text before sketch extraction to make the inputs more compatible with GENIUS. Since GENIUS is a generative model which have potential to generate new entities that don't exist in the training set, resulting in unlabeled entity problem. To address this issue, we borrow the approach in (Li et al., 2020) by only labeling the entities that occur in the training set, leaving other tokens labeled with "X" (which don't contribute to the loss function).

MRC. For MRC task, we use the widely used SQuAD dataset for experiment. Each training example of SQuAD is a triple of (p, q, a) where p is a multi-sentence paragraph that contains the answer a to the question q. We use GENIUS to generate new paragraphs while keeping the question q unchanged. To make the original answer a accessible and reasonable for the generated paragraphs, we also keep the sentence where the answer occurs (noted as s_a) nearly unchanged and only augment the preceding (p_{pre}) and following text (p_{post}) of s_a. We use the current question as the TRI to extract sketches from p_{pre} and p_{post}, noted as s_{pre} and s_{post} respectively. Then the sketch to the GENIUS model is the concatenation of $[s_{pre}, s_a, s_{post}]$. For all augmentation methods, we also filter the augmented samples by the basic model (non-aug) to remove the potential noisy samples, as suggested by (Liu et al., 2020).

A.4 Computation Infrastructure & Augmentation Efficiency

We use 8 NVIDIA V100 cards for GENIUS pre-training and 1 V100 card for all downstream tasks (classification, NER, MRC). In data augmentation process, GENIUS can generate about 700 samples per minute for `max_len=60` and about 440 samples per minute for `max_len=200` with `batch_size=32` on a single V100 card. The speed can be improved by increasing the `batch_size` or using more GPUs. Table 8 shows the augmentation speed in different settings.

GENIUS (large)'s Augmentation Efficiency (single V100)		
batch size	**max_len**	**speed (#/minute)**
32	60	698.4
32	200	444.4
64	60	1000.0
64	200	685.7

Table 8: **GENIUS**'s augmentation efficiency on a single V100, with different settings.

B More Examples

B.1 Sketch-based Generation Examples

We provide some examples to subjectively compare the quality of BART, GENIUS and the giant GLM-130B (Zeng et al., 2022) model in Figure 6. GLM-130B is a bilingual model with about 130B (130,000M) parameters pre-trained on over 400 billion text tokens, which is much larger (about 300-1000 times larger) than GENIUS models and uses much more general data for pre-training. Therefore, we expect GLM-130B to be a very strong model for blank-filling tasks. Sketch 1-3 are for English tasks, which show that GLM-130B can also generate fluent sentences given the sketches. The results of GLM-130B are generated by calling their online API[5]. BART can also connect the fragments in a natural way, but can only fill in very few words. GENIUS also handles these inputs well, with even more content inserted than GLM-130B. For the Chinese tasks, we find both BART and GLM-130B generate in-fluent or meaningless text for sketch 4 and 5. Chinese sketch-based reconstruction is more difficult since the tokens are processed at the character level while in English it's word/subword level. However, GENIUS manages to generate coherent text based on these extremely masked sketches. Sketch 6 and 7 are traditional blank-filling tasks with only one mask. All three methods can fill in the blank in a fluent way. However, due to the relatively limited training corpus, BART and GENIUS are unable to give the "precise answer" into the blank in sketch 7, which requires extra knowledge about the concepts described in the context. GLM-130B instead can give the correct answer based on its super large-scale pre-training and model weights.

[5]https://huggingface.co/spaces/THUDM/GLM-130B

B.2 Augmentation Examples

Table 9 gives examples of GENIUS-generated sentences and the effect of using "attribute controlling".

Table 10 illustrates some examples generated by different augmentation methods for text classification tasks. EDA and STA are rule-based methods, while the others are LM-based methods. According to the examples, we can see that GENIUS can generate much more diverse new samples than other methods, while also preserving the core semantics of the original samples.

Table 11 shows some GENIUS-generated examples for CoNLL03 task. GENIUS introduces new contexts for existing entities. Table 12 shows the examples for SQuAD. Based on the target-aware sketch extraction, GENIUS can generate different new paragraphs according to the different questions.

1	*sketch*	___ play basketball ___ a strong team ___ University ___
	BART-large (406M)	7. We **play basketball** against **a strong team** in **University** Park.
	GLM-130B (130,000M)	to go **play basketball** for such **a strong team** at the **University** of Texas.
	GENIUS-large (406M)	I **play basketball** for **a strong team** at the **University**. I like to play in the gym. I am very athletic and have a great attitude. I have good grades and a good sense of humor.
2	*sketch*	___ Conference on Empirical Methods ___ submission of research papers ___ Deep Learning ___
	BART-large (406M)	The **Conference on Empirical Methods** invites the **submission of research papers** in **Deep Learning** and Machine Learning
	GLM-130B (130,000M)	International Joint **Conference on Empirical Methods** in Natural Language Processing for **submission of research papers** on NLP and **Deep Learning**. The conference…
	GENIUS-large (406M)	The **Conference on Empirical Methods** is a forum for the **submission of research papers**. It is open to anyone interested in **Deep Learning**, machine learning, and statistical inference. The conference will be held in New York, NY…
3	*sketch*	___ interview ___ The Associated Press ___ Trump announced another White House run, Pence declined ___ former president ___. But he ___ positioned himself ___ potential alternative ___ Republicans ___ conservative leadership ___ Trump era.
	BART-large (406M)	In an **interview** with **The Associated Press** after **Trump announced another White House run, Pence declined** to endorse the **former president**. **But he positioned himself** as a **potential alternative** to **Republicans** for **conservative leadership** in the **Trump era**. —AP
	GLM-130B (130,000M)	an exclusive **interview** with, **The Associated Press** before Donald **Trump announced another White House run, Pence declined** to criticize the **former president**'s comments . **But he** also firmly **positioned himself** as a **potential alternative** for disaffected **Republicans** looking for **conservative leadership** in the **Trump era**.
	GENIUS-large (406M)	In an **interview** with **The Associated Press**, before **Trump announced another White House run, Pence declined** to say whether he would run against the **former president**. **But he** has **positioned himself** as a **potential alternative**, saying **Republicans** need more **conservative leadership** in the **Trump era**."I don't know what the answer is," he said…
4	*sketch*	今天__篮球__上海财经大学__
	BART-base-chinese (116M)	**今天**的**篮球**是**上海财经大学**篮球
	GLM-130B (130,000M)	**今天**是个好日子阳光**篮球**青春**上海财经大学**之
	GENIUS-base-chinese (116M)	**今天**，我们来看看中国**篮球**协会与**上海财经大学**联合举办的这次比赛。
5	*sketch*	___自行车___数显屏___骑行___健身___体型___舒适___
	BART-base-chinese (116M)	多功能**自行车**电脑**数显屏**多种多样**骑行**方式**健身**方式**体型**轻便**舒适**安全
	GLM-130B (130,000M)	**自行车** 自行车 **数显屏** 自行车 **骑行** **健身** 健身车 健身器 **体型** 健身车 **舒适** 健身车
	GENIUS-base-chinese (116M)	这款**自行车**配备了**数显屏**，无论是**骑行**还是**健身**都非常方便，可以根据自己的**体型**和身高来选择，让驾驶更加**舒适**
6	*sketch*	今天的篮球____上海财经大学举办
	BART-base-chinese (116M)	**今天的篮球**在**上海财经大学举办**
	GLM-130B (130,000M)	**今天的篮球**联赛是由**上海财经大学举办**
	GENIUS-base-chinese (116M)	**今天的篮球**比赛，由**上海财经大学举办**
7	*sketch*	意大利米兰市古城堡旁的凯旋门，是1807年为纪念____而建。
	BART-base-chinese (116M)	**意大利米兰市古城堡旁的凯旋门，是1807年为纪念**凯旋**而建。**
	GLM-130B (130,000M)	**意大利米兰市古城堡旁的凯旋门，是1807年为纪念**拿破仑胜利**而建。**
	GENIUS-base-chinese (116M)	**意大利米兰市古城堡旁的凯旋门，是1807年为纪念**他的父亲**而建**，…

Figure 6: Examples of generated text from BART, GLM-130B and GENIUS in both English and Chinese versions. GLM-130B is a bilingual model with about 130B (130,000M) parameters, while GENIUS-large is English only with about 406M parameters and GENIUS-base-chinese is about 116M parameters, much smaller than the GLM-130B model. We use "_" to represent the mask token for clearer illustration. The **bold** are elements from the sketch and the blue are newly generated contexts. Sketch 1-8 are typical sketches defined in this work which contain some key information with many masks, while sketch 6-7 are traditional blank-filling tasks where only one mask is waiting for prediction.

sketch	<mask> **Conference on Empirical Methods** <mask> **submission of research papers** <mask> **Deep Learning** <mask>
GENIUS *generated*	The **Conference on Empirical Methods** is a forum for the **submission of research papers** in the field of **Deep Learning**. The conference is open to all interested parties. It will be held at Stanford University...
sketch	<mask> **the European Union** <mask> **month by EU** <mask> **Farm Commissioner Franz** <mask>
GENIUS *generated*	Farmers in **the European Union** will have to pay more for their produce if they want to be included in the EU's new farm bill, which is due to be approved next **month by EU** lawmakers. EU **Farm Commissioner Franz** Hoppe said on Wednesday...
prompt= ***"Environment: "***	~~Environment:~~ What does **the European Union** need to do? A new report released this **month by EU**'s **Farm Commissioner Franz** Hauser says that the EU **needs to do more to protect the environment**. He says that there is a ...
prompt= ***"Technology: "***	~~Technology:~~ What does it mean for **the European Union**? A new report released this **month by EU**'s **Farm Commissioner Franz** X says that the EU's agricultural sector is in danger of **losing its competitiveness due to a lack of technological innovation**. The report...
prompt= ***"Good News: "***	~~Good News:~~ The U.S. and **the European Union** **are getting closer to a deal** on farm subsidies. The agreement, which was reached last **month by EU** Agriculture and **Farm Commissioner Franz** Haase, will **give farmers a 1.5 percent increase in their subsidies**...
prompt= ***"Bad News: "***	~~Bad News:~~ **the European Union** **will not be able to raise the minimum wage** to 15 an hour next **month by EU** standards. **Farm Commissioner Franz** Haass said that the EU **would not raise the wage because it was not in the EU's interest**. The minimum...

Table 9: Examples of sketches and GENIUS generated samples. The 5-8 rows illustrate the impact of "attribute controlling". After the generation, we delete the prompt text at the beginning. The **black bold** represents the key parts. The **red bold** represents the text that are close to the meaning of the given prompts.

Original Text	I only have eight myself, but I know a few people who have at least 15. They have played a lot of leagues, even in a short amount of years, but in various sports. **[label: Sports]**
EDA	I only have eight myself, but I know a few people who have at least 15. They have played a lot of leagues, even in a short amount of, but in various sports.
STA	I only have eight myself but I know few people who have at least 15 They have played a lot of leagues, even in a short amount of years, but in various sports.
BackTrans	I only have eight people, but I know a few people who have at least 15. They have played a lot of leagues, even in a few years, but in various sports.
MLM	I only have eight myself, but I know a few people who have at least 15. They all played a bunch of leagues, even in a short amount of years, but in various sports
C-MLM	~~*Sports:*~~ Me only got eight myself, but I have a few buddies who have at least twelve. They have played a lot of leagues, but in a set amount of years time but in various sports.
LAMBADA	~~*Sports:*~~ I don't think it would be a bad idea to have him on the team. If he gets hurt, it would certainly hurt him a little bit. But if he does get hurt, it would hurt him a lot more than any other guy on the team. If he gets hurt in the playoffs, it would hurt him a lot more than any other guy on the team. But if he does win, it would hurt him a lot more than any other guy on the team. And if he does lose, it would hurt him...
sketch	*Sports:* <mask> lot of leagues <mask> in various sports <mask>
GeniusAug	~~*Sports:*~~ There are a lot of leagues in various sports. Here are some of the best.
GeniusAug-f	~~*Sports:*~~ I have played in a lot of leagues, in various sports. I'm not into any of them. But I'm a big fan of football. It's a sport that gives you something to do besides playing football. If you want to try something different, try something new.
sketch mixup	*Sports:* <mask> leagues <mask> High sped <mask> knee and <mask> liverpool rox <mask> leg <mask> various sports <mask> knee <mask> of leagues <mask> knee and straight <mask> Super Bowl XL <mask> sports <mask> rear knee and <mask> lot of leagues <mask> legs <mask> in various sports <mask>
GENIUS-mixup	~~*Sports:*~~ Which leagues have the High sped up right knee and the liverpool rox right leg? In various sports, the right knee is the most important. In a lot of leagues, the left knee and straight leg are the best. In Super Bowl XL and other sports, it is the right rear knee and right leg that are the most dangerous. In most of the games, the high speed of the leg is the best thing for the player. A lot of teams have the high sped up legs in various sports.

Table 10: Data augmentation for text classification. During the augmentation, C-MLM, LAMBADA and GeniusAug all start with a label prompt. The prompt text is removed after generation.

Original samples from CoNLL03:

```
EU    rejects German call to boycott British lamb .
B-ORG O       B-MISC O    O  O       B-MISC  O    O
Germany 's representative to the European Union 's veterinary committee Werner Zwingmann said on Wednesday
B-LOC   O  O              O  O   B-ORG    I-ORG O  O          O         B-PER  I-PER     O    O  O
```

GeniusAug generated samples:

```
The German government says the idea is unacceptable and that the EU    should reject it .
O   B-MISC O          O    O   O    O  O            O   O    O   B-ORG O      O      O  O
```

```
The spokesman added that the EU    was not considering imposing a ban on the meat .
O   O          O     O    O   B-ORG O   O   O           O        O O   O  O   O    O
```

```
Britain and France are also considering a ban on the sale of tobacco to minors .
B-LOC   O   B-LOC  O   O    O           O O   O  O   O    O  O       O  O      O
```

```
The measures have been proposed by EU    Farm and EU    Commissioner Franz Fischler .
O   O        O    O    O        O  B-ORG O    O   B-ORG O            B-PER I-PER    O
```

```
`` But Fischler agreed that the EU    s standing on the Spanish Farm is not a problem .
O  O   B-PER    O      O    O   B-ORG O O        O  O   B-MISC  O    O  O   O O       O
```

```
It says the meat is being exported to China , Brazil , India and the US .  # unlabeled entity problem
O  O    O   O    O  O     O        O  B-LOC O O      O O     O   O   O  O
```

Table 11: GeniusAug generated samples for NER task. The last row shows an example where GENIUS generates new entities like "India", "Brazil" that don't exist in the training data, which results in the unlabeled entity problem. We take the approach in (Li et al., 2020) to deal with this issue.

Original sample from SQuAD:
Paragraph: Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. *It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858.* At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.
Question: To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?

New paragraph generated by GeniusAug:
Paragraph: The basilica has a golden statue of the Virgin Mary and a statue of Christ, which is hidden away behind the basilica. *It is a replica of the grotto at Lourdes, France where the virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858.* The basilica contains 3 statues and a modern stone statue of Mary, which was built in the late 1800s. There are also a number of other relics. The relics are kept in a special room on the main floor of the church. The church is open to the public from 9:00am to 4:00pm Monday Friday and 9:30am to 3:00 pm on Saturday and Sunday. The museum is open from 10:00 am to 5:30 pm on weekdays.

Original sample from SQuAD (same paragraph with a *different* question):
Paragraph: Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. *Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes".* Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.
Question: What is in front of the Notre Dame Main Building?

New paragraph generated by GeniusAug:
Paragraph: In front of the school and the Main Building is a gold dome and a golden statue of the Madonna. *Immediately in front of our Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes" on the top of the dome.* In the Basilica of the Immaculate Conception, there is a small chapel with a small altar. Also behind the basilica is a grotto at Lourdes. There is a statue of Mary in the center of the grotto. We recommend that you take a bus to the nearest bus stop, as there is ample parking. We suggest that you arrive early in the morning to avoid long lines.

Table 12: GeniusAug generated samples for SQuAD. Given a paragraph, GeniusAug can generate different new paragraphs according to the given question, based on the target-aware sketch extraction. The underlined words are the ground truth answers, the *italicized* sentences are the key sentences that contain the answer, which are kept in the sketch during augmentation.