SEGA: SkEtch-based Generative Data 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

Abstract

In this work, we propose a novel textual data augmentation method — **SEGA**, which is pre-trained on a large-scale general corpus based on a *reconstruction from sketch* learning objective, to generate paragraphs given textual sketches (i.e., textual spans, phrases or words, concatenated by mask tokens). During augmentation, SEGA first extracts a target-aware sketch from the original text and then generates a new context for the sketch, which is a much different approach compared with previous augmentation methods. The large-scale pre-training on general corpus makes SEGA a *ready-to-use* data augmentation tool for various NLP tasks without the need for further fine-tuning on downstream datasets. Extensive experiments on 6 text classification datasets demonstrate that SEGA significantly outperforms previous methods in both in-distribution (ID) and out-of-distribution (OOD) settings. Experiments on named entity recognition (NER) and machine reading comprehension (MRC) tasks also indicate that SEGA is a strong augmentation method for other NLP tasks.[1]

1 Introduction

The success of deep learning models is usually built on the large training data with good quality, which may not be easily available in real applications (Feng et al., 2021). Therefore, data augmentation techniques are widely used to improve models' generalization performance, especially in low-resource scenarios. In the field of natural language processing (NLP), plenty of augmentation methods have been proposed. For example, rule-based methods (Wei and Zou, 2019; Feng et al., 2020; Guo et al., 2022) define some text-editing rules such as replacement, insertion, and deletion for augmentation; Back-translation (Sennrich et al., 2016; Yu et al., 2018) methods utilize translation models to translate the text into other language(s) and then back to the original language; In recent years, thanks to the growing ability of large pre-trained language models (PLM), more and more PLM-based methods are proposed for generative text augmentation, including 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). These PTM-based generative augmentation methods usually involve an extra model fine-tuning step on downstream tasks.

By studying the characteristics of these methods, we claim that they 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) modify the original text by introducing small perturbations (e.g. by replacing/inserting some words). If the perturbations are too large, the generated samples may even hurt the performance. Therefore, the resulting new samples are semantically and structurally very similar to the original sample, which is limited in terms of diversity; 2) Aggressive methods aim to generate totally new training samples instead of just modifying the original ones. For example, LAMBADA (Anaby-Tavor et al., 2020) fine-tunes the 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 risk of generating undesirable samples.

Motivated by this gap, we propose a new approach between these two directions, where the key parts of the text are extracted and fixed then a generative model produces new contexts around these key parts. By doing so, the diversity is much greater than the conservative methods as larger parts of the

[†]Work done during the internship at MSRA NLC group
[*]Corresponding author, email: hlhuang@shufe.edu.cn
[1]Code and model are publicly available at http://github.com/beyondguo/SEGA

original text are altered. At the same time, the quality is more guaranteed than aggressive methods since the core semantics are preserved.

To achieve this, we pre-trained a conditional language model (CLM) on a large-scale general corpus using a novel *reconstruction from sketch* objective, where the CLM is learned to reconstruct a piece of text based on the extracted sketch. During augmentation, we extract the *target-aware sketch* and feed it into the pre-trained model to produce new training samples. We call this new approach as **SEGA** (**se**tch-based **g**enerative **a**ugmentation).

Large-scale pre-training of SEGA makes it 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 SEGA 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 illustrate that SEGA 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.

2 Related Work

2.1 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). In this work, we mainly focus on the input space data augmentation.

In the input space, rule-based methods are widely used for various tasks. Wei and Zou (2019) propose the EDA tool for text classification tasks which consists of four simple text-editing rules to modify original samples. Similar rule-based methods are proposed for named entity recognition (Dai and Adel, 2020) and natural language generation (Feng et al., 2020). Recently Guo et al. (2022) proposes the STA tool which further improves the performance of rule-based methods by modifying the text in a selective manner.

Apart from the rule-based methods, there are also plenty of generative model-based augmentation methods. Back-translation (Sennrich et al., 2016; Yu et al., 2018) methods augment text by first translating the text to other languages and then back to the original one. Paraphrasing (Gao et al., 2020; Damodaran, 2021) methods re-write the original sequences through fine-tuned seq2seq models. Inspired by (Kobayashi, 2018) which utilizes contextual information for word replacement, Wu et al. (2019); Kumar et al. (2020) utilize the masking mechanism in masked language models (MLM) like BERT (Devlin et al., 2019) to produce new samples by masking and replacing the words in the original text. Anaby-Tavor et al. (2020); Kumar et al. (2020) 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. Zhou et al. (2022b) proposes MELM for named entity recognition tasks, which first linearizes the sequences and then fine-tunes the model by masking the entities.

Our proposed SEGA 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, SEGA introduces more diversity to the training set; Compared with AR-based open generation methods, SEGA 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 SEGA can be directly used for augmentation (though we also show fine-tuning can lead to further performance gains); Last but not least, SEGA is a more general approach applicable to a variety of NLP tasks while most of the traditional methods are task-specific.

2.2 Denoising Pre-Training

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., 2019) and ALBERT (Lan et al., 2019). BART (Lewis et al., 2020) uses a novel text infilling task

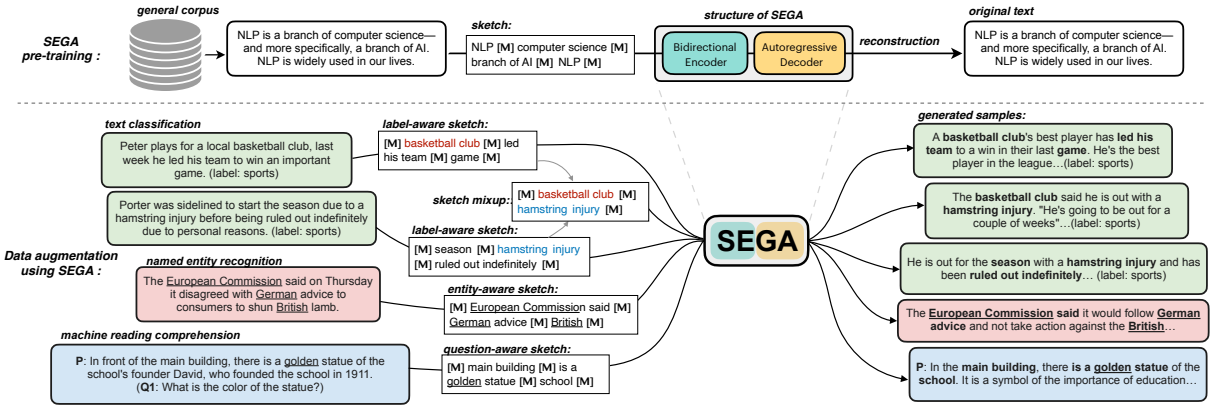


Figure 1: Illustrations of the pre-training and augmentation process of SEGA, where [M] refers to the model's mask token. The pre-training is conducted only once and then can be applied to various NLP tasks for data augmentation. During augmentation, we extract the *target-aware* sketches as SEGA's inputs.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

with a bigger masking ratio of 30% for pre-training, where small spans of text are corrupted for reconstruction. Inspired by the text infilling task, we propose the *reconstruction from sketch* task to pre-train our SEGA model to reconstruct the original text given only a sketch, which consists of the most informative parts of the text.

3 Methodology

3.1 SEGA Pre-Training

When writing an article, we usually start by drawing up a general framework containing the key elements we wish to mention (which we call as **sketch** in this work), on the basis of which we write the content. We want to train a conditional language model to mimic this process. This process can be viewed as a type of text reconstruction from corrupted text, which is widely used as the denoising pre-training objective for many PTMs like BERT and BART, as introduced in Section 2.2. However, these tasks only mask a small fraction of tokens from the original text (15% for BERT and 30% for BART), which are incompatible with our goal — to reconstruct the whole content based only on a sketch where most of the content may be corrupted, which can hardly be reconstructed by BERT or BART. Therefore, we propose *reconstruction from sketch* pre-training task, where a sketch is first extracted from a document and then train the language model to reconstruct the original text from the sketch.

To prepare the sketch, which should contain the core semantics and the outline of the original text, we propose an *extraction-projection-masking* procedure. First, we extract the keywords or key-phrases (up to 3-grams) using the unsupervised keywords extraction algorithm YAKE (Campos et al., 2020) from the original text, which roughly accounts for 1/5 of the text length. Then, these key parts are projected back to the original text. Note that one key-phrase may occur multiple times and different key-phrases have chances to be overlapped. Finally, each of the remaining parts is replaced with a *single mask token*, resulting in about 73%[2] masking on the original text on average, which is a much larger proportion than previous PTMs. We construct about 27 million (27,599,676) `<sketch,text>` pairs from the C4-realnewslike corpus (Raffel et al., 2019) for pre-training. SEGA uses a bidirectional encoder and an auto-regressive decoder, initialized with the weights of BART-large. An overview of the pre-training process is shown in the upper part of Figure 1.

Overall, SEGA pre-training has two major differences from previous denoising pre-training objectives: **Where to mask**: Unlike previous methods which mask random tokens/spans of the original text, we present a sketch extraction pipeline to extract the most informative parts of the text, leaving the remaining less-informative parts masked; **How much to mask**: Previous methods usually only mask a small fragment of text in reconstruction tasks, SEGA may mask up to 80% of the text.

There are also other possible designs for the sketch. In our preliminary experiments, we also compare with three other templates $\{T_1, T_2, T_3\}$: T_1 uses the simple concatenation of the keywords

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

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 *(Ours)*: NLP **[M]** computer science **[M]** branch of AI **[M]** NLP **[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 *(Ours)*	**28.77**	**17.89**	**26.74**

Table 1: Comparison of different sketch templates. **[M]** is the mask token.

sketch	**[M] use machine learning [M] AI techniques [M]**
SEGA 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 2: Examples of synthetic sentences generated by SEGA and the effect of adding specific prompts for "attribute controlling". The ***red italicized*** represents the text that is close to the given attribute.

joined by spaces, ordered by their importance given by the extractor. Note that T_1 is similar to the approach used for keywords-to-text (K2T) generation models (Gehrmann et al., 2021; Bhatia, 2021). However, K2T methods usually can only accept a few uni-grams as inputs and generate short sentences, which is not suitable for data augmentation scenarios. T_2 is slightly different from T_1, where the keywords are sorted by their original order; T_3 further allows multiple occurrences of a keyword or overlap between different words as in the original text. We call our template described above T_4, which is nearly the same as T_3 but the missing parts are replaced with single mask tokens. Examples of these templates are shown in Table 1. We pretrain SEGA using these different sketch templates and report their reconstruction performances on the dev set by ROUGE-1/2/L scores. The results show that our choice in this work helps the SEGA to learn better during pre-training. Compared with other templates, T_4 is closer to the text-infilling task of BART, which makes the SEGA easier to continue pre-training on BART. In the following experiments in this work, we utilize T_4 as the default sketch template. More comparisons of these different templates on downstream tasks will be studied in future work.

3.2 Data Augmentation with SEGA

Target-aware sketch extraction. The generation of SEGA is conditioned on the sketches of the original samples, therefore the input sketches have a significant impact on the quality of the generated text. To make the extracted sketches more task-relevant, we propose *target-aware sketch extraction* to extract the task-related information as the input to SEGA. 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. For text classification tasks, the TRI is the corresponding category or the category description; for NER tasks, the TRI is the entities; for machine reading comprehension tasks, the TRI is the corresponding question. We calculate the matching scores for each n-gram as follows:

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

where e_f is a fused embedding of the document and its TRI, s_i is the matching score for ith n-gram w_i. We utilize sentence-BERT (Reimers and Gurevych, 2019) as the encoder $\mathbf{E}$ to obtain the embeddings. We set $\lambda = 0.5$ and select the top 20% n-grams as the keywords/key-phrases according to the matching scores. These extracted target-related parts are then used to form the sketch following the steps in the sketch construction pipeline introduced in Section 3.1.

SEGA Generation. After obtaining the sketch of a training sample, we use the pre-trained SEGA to generate new samples conditioned on this sketch. We use beam search with random sampling to decode and generate the text. Table 2 illustrates an example sketch and its corresponding generated

text by SEGA. SEGA 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 SEGA to generate content towards certain attributes, as shown in the last four rows in Table 2. Note that we didn't specifically pre-train or fine-tune SEGA 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 SEGA is acquired from the *reconstruction from sketch* pre-training. This characteristic makes SEGA flexible to control the quality and diversity during data augmentation, especially for sentiment or topic classification.

More advanced options. Apart from the standard usage of SEGA 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 SEGA. 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 SEGA model.

4 Experiments

The main experiments are conducted on the text classification tasks in Section 4.1, which are also most widely studied in academic and industrial communities (Bayer et al., 2021). In Section 4.2, we show that SEGA is also applicable to other NLP tasks, including named entity recognition (NER) and machine reading comprehension (MRC).

4.1 Text Classification Tasks

4.1.1 Setup

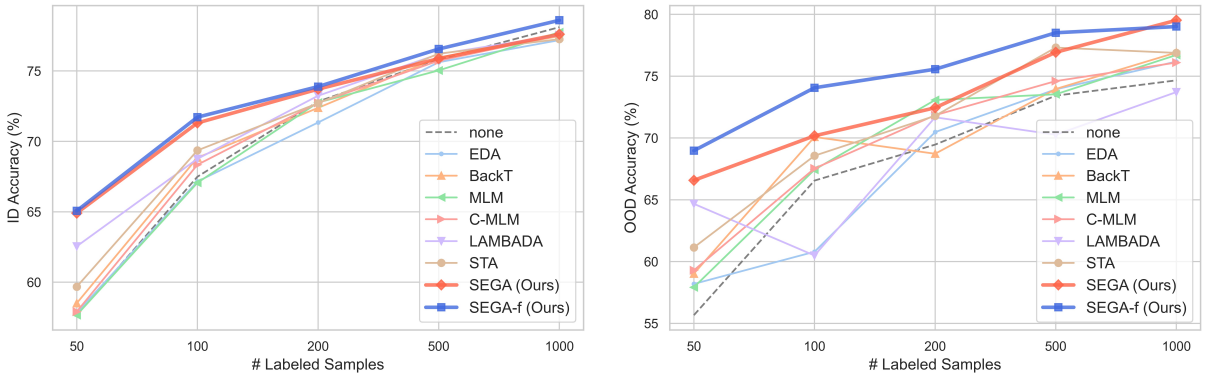
Datasets. For text classification, we conduct experiments on 6 widely used datasets, including four topic classification datasets **Yahoo** (Zhang et al., 2015), **20NG**, **BBC** (Greene and Cunningham, 2006), **Huff**[3] (Misra and Grover, 2021) and two sentiment classification datasets **SST2** (Socher et al., 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); (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 **SEGA** for data augmentation, we also compare with a fine-tuned version **SEGA-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.3). 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

[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*
Back-Trans (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*
SEGA (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*
SEGA-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 3: 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 2: Augmentation effectiveness when given different numbers of labeled samples in ID and OOD settings. We plot the averaged scores from all datasets.

experiment on the stronger *RoBERTa-large* (Liu et al., 2019) classifier, discussed in the later part.

4.1.2 Results

Table 3 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 2. The full results under all settings can be found in Appendix A.5.

In-distribution evaluations. Our proposed SEGA and SEGA-f both boost the performance of base classifiers in the ID evaluations, with average improvements of around 2% and 3% respectively. SEGA and SEGA-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 SEGA 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, SEGA 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, SEGA and SEGA-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

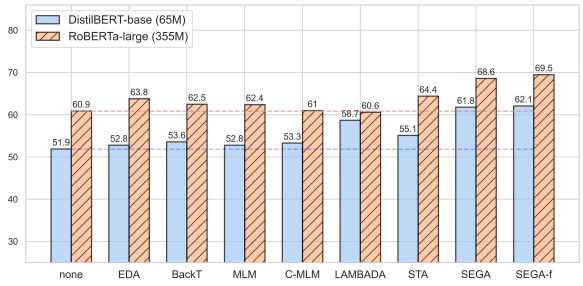


Figure 3: Performances on different classifiers – DistilBERT-base and a stronger RoBERTa-large. The results are averaged across all datasets on a $n = 50$ setting.

Dataset	*none*	*pre-sota*	*SEGA*	*SEGA-f*	***SEGA-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 4: SEGA-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.

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 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 SEGA and SEGA-f makes them very suitable for the OOD generalization: By masking the unimportant parts in the original text, SEGA 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 2 shows the performances of different methods with different numbers of labeled training samples. In the severe low-resource scenarios ($n = \{50, 100\}$), SEGA and SEGA-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 SEGA and SEGA-f continue to maintain huge advantages over others in the OOD setting.

Effectiveness of fine-tuning. Compared with SEGA, SEGA-f achieves better results in most datasets. During the pre-training process, the model inputs are *general* sketches, while in the SEGA fine-tuning we are using *target-aware* sketches, this helps the model 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 each dataset, as shown in Figure 3. Surprisingly, SEGA/SEGA-f *helps the weaker DistilBERT-base classifiers to outperform the strong RoBERTa-large classifiers* (comparing the orange bar of "none" and blue bars of "SEGA" and "SEGA-f"). SEGA/SEGA-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 **SEGA-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 SEGA-mixup can further boost the performance for most experimented tasks. Note that SEGA-mixup doesn't need a fine-tuning step. There are other possible ways to improve the diversity of SEGA-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

n	**NER** (CoNLL03) 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
***SEGA** (Ours)*	**49.17**	**61.12**	**66.10**	74.69	**62.77**
n	**MRC** (SQuAD) 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
Back-Trans (2018)	**19.26**	26.13	36.21	50.31	32.98
***SEGA** (Ours)*	19.03	**28.60**	**37.02**	**51.83**	**34.12**

Table 5: Data augmentation for the NER and MRC tasks with different labeled sizes.

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 **Back-Trans** are used. The other settings are the same as text classification tasks.

4.2.2 Results

Named entity recognition. The upper part of Table 5 summarizes the comparison for the CoNLL03 task with different sizes of labeled data. When the number of labeled data $n \in \{50, 100, 200\}$, SEGA outperforms all the baselines by a large margin. When n reaches 500, SEGA 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, SEGA can generate more diverse and natural text, which is helpful for the NER model to learn new patterns for entity recognition. Compared with MELM, SEGA 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 SEGA and MELM is that SEGA aims to diversify the context while MELM focuses on introducing more entities. We will explore the combination of SEGA and MELM in future work to further improve the performance.

Machine reading comprehension. The lower part of Table 5 shows the exact match (EM) scores for SQuAD in different resource levels. SEGA achieves the best results when $n = \{100, 200, 500\}$ and comparable scores with Back-Trans when $n = 50$. The two baselines SR and Back-Trans are both effective augmentation methods among all training sizes. Compared with the baselines, SEGA 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, SEGA is orthogonal to these QDA methods: SEGA 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 SEGA to generate more diverse training samples for MRC and QA tasks.

4.2.3 Ablation Study

SEGA pre-training. SEGA 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 SEGA – 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 SEGA's pre-training by directly using BART-large for our proposed sketch-based generation, without the SEGA-like pre-training, denoted as "− PT". We experiment on the Huff dataset and Huff⇒BBC task as shown in Table 6. Compared with SEGA or SEGA-f, using BART for sketch-based generation achieves significantly worse results, especially in the OOD or low-resource setting. The pre-training

ID (Huff)						
n	50	100	200	500	1000	avg. (Δ)
none	71.44	78.99	78.57	81.48	85.35	79.17
SEGA	**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 (-2.36)
SEGA-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 (-2.89)

OOD (Huff⇒BBC)						
n	50	100	200	500	1000	avg. (Δ)
none	39.62	63.58	65.20	65.90	77.30	62.32
SEGA	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 (-11.24)
SEGA-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 (-13.30)

Table 6: Ablation study on the effectiveness of SEGA's large-scale pre-training

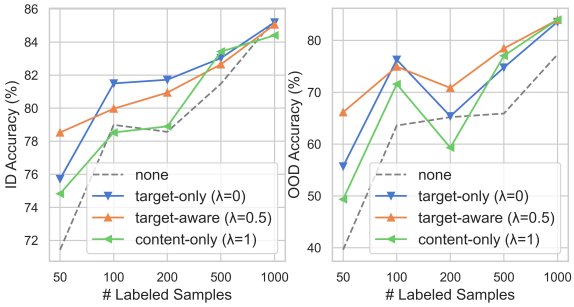


Figure 4: Comparison of different sketch extraction strategies on the Huff dataset with different labeled sizes.

of BART determines it can only reconstruct small textual corruption. In comparison, SEGA 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 4 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 Discussions

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

1) Compared with rule-based methods, SEGA 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) SEGA'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 SEGA 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.4.

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

6 Conclusion

In this paper, we present a sketch-based generative data augmentation method called SEGA, which is pre-trained on a large-scale corpus with a novel *reconstruction from sketch* pre-training task. SEGA can generate coherent and diverse paragraphs given a textual sketch and can be directly applied to various NLP tasks for data augmentation without further fine-tuning on downstream datasets. Extensive experiments reveal the strong performance of SEGA on classification, NER, and MRC tasks. In

future work, we will further study the effectiveness of SEGA on other languages and the potential of *reconstruction from sketch* pre-training to 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.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *arXiv preprint arXiv:2107.03158*.

Gagan Bhatia. 2021. Key-to-text generation based on t5. https://github.com/gagan3012/keytotext.

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.

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)*.

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.

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*.

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.

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*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

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*.

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. 2019. 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*.

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.

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.

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)*, pages 6382–6388, 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.

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*.

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.

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.

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 SEGA 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 `l` ranging from 50 to 200 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 ± 7.05.

SEGA is a seq2seq model, with a bidirectional encoder and an auto-regressive decoder. SEGA uses the same structure of BART (Lewis et al., 2020) and is initialized with the weights of BART-large. SEGA is optimized using AdamW (Loshchilov and Hutter, 2017) optimizer with learning rate 5.6e-5 and weight decay 0.01. We pre-train the SEGA model for 3 epochs with batch size 16 using 8 NVIDIA V100 cards which takes a few days.

We also pre-trained `SEGA-base` for faster inference and Chinese versions `SEGA-large-chinese` and `SEGA-base-chinese`. We host our models on the Huggingface Model Hub:

- SEGA-large (this paper): https://huggingface.co/beyond/sega-large
- SEGA-base: https://huggingface.co/beyond/sega-base
- SEGA-base-chinese: https://huggingface.co/beyond/sega-base-chinese

Other variants will be uploaded in future.

A.2 Baselines Implementation

The baselines we compare in our experiments can be divided into two groups: rule-based and LM-

based.

Rule-based methods:

- EDA (Wei and Zou, 2019): a rule-based method composed of four text editing operations: synonyms replacement/insertion, random swap/deletion;
- STA (Guo et al., 2022): a method specifically designed for text classification tasks. STA first identifies the word roles according to semantic and statistical information and then applies different text-editing operations on words with specific roles. According to their results, STA outperforms EDA and some strong LM-based methods in low-resource settings;
- MR (Dai and Adel, 2020): replaces the NER entities with other entities of the same type in the training set;
- Rule-mix (Dai and Adel, 2020): specifically proposed for NER tasks, consists of four different operations: mention replacement, synonyms replacement, token replacement and segment shuffling.

LM-based:

- MLM (Kumar et al., 2020): utilizing the mask language modeling ability of PTMs to substitute words with others randomly drawn according to the MLM predicted distribution based on the current context;
- C-MLM (Wu et al., 2019; Kumar et al., 2020): further fine-tuning the MLM model conditioned on the sequence label, which makes the generated text more task-related;
- Back-Trans (Sennrich et al., 2016; Yu et al., 2018; Silfverberg et al., 2017): translate a sequence into another language and then back into the original language;
- LAMBADA (Anaby-Tavor et al., 2020): fine-tune GPT-2 with the original examples prepended with their labels, and then generate examples by feeding the fine-tuned model certain labels. After generation, a basic classifier trained on the original data is required to denoise the generated samples by only keeping the generated samples with high confidence;
- MELM (Zhou et al., 2022b): first linearizes the sequences and then fine-tunes the PLMs with a masked entity language modeling task, then new sequences can be generated by substituting the entities in the samples.

A.3 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]

[4]https://github.com/huggingface/transformers/

for model training and evaluating.

A.4 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 SEGA. We use the **a**ttribute-**c**ontrolling (see Section 3.2) to make the generated samples closer to their corresponding labels. Specifically, we add the label text as the prompt 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 SEGA pre-training. Therefore, we concatenate the consecutive sequences into longer text before sketch extraction to make the inputs more compatible with SEGA. Since SEGA 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 SEGA 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 SEGA 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).

tree/main/examples/pytorch/question-answering

A.5 Full Results on Text Classification

Table 7 gives the full results on all the text classification tasks, including the ID and OOD evaluations under different low-resource levels. SEGA or SEGA-f may not be the best method for each specific setting (eg. $n = 50$, BBC, ID), but achieves the highest average scores for all settings.

A.6 Computation Infrastructure & Augmentation Efficiency

We use 8 NVIDIA V100 cards for SEGA pre-training and 1 V100 card for all downstream tasks (classification, NER, MRC). In data augmentation process, SEGA 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.

B More Examples

Table 9 gives examples of SEGA-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 SEGA can generate much more diverse new samples than other methods, while also preserving the core semantics of the original samples.

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

		ID evaluation								OOD evaluation					
n	**Method**	**Huff**	**BBC**	**IMDB**	**SST2**	**Yahoo**	**20NG**	*avg.*	Δ	H⇒B	B⇒H	I⇒S	S⇒I	avg.	Δ
50	none	71.44	94.94	58.78	67.22	29.81	23.78	*57.66*	–	39.62	61.97	56.48	64.58	55.66	–
	EDA	73.47	95.10	58.40	65.83	30.43	23.76	*57.83*	0.17	46.92	66.57	60.64	58.64	58.19	2.53
	BackTrans	73.52	94.58	62.57	64.77	31.13	24.48	*58.51*	0.85	45.84	61.11	67.06	62.08	59.02	3.36
	MLM	72.45	95.04	59.21	64.08	31.07	24.18	*57.67*	0.01	42.30	64.52	60.56	64.28	57.92	2.25
	C-MLM	72.18	**95.60**	57.59	66.51	32.07	23.47	*57.91*	0.24	41.90	68.55	60.40	66.32	59.29	3.63
	LAMBADA	76.00	91.50	62.51	75.11	**44.13**	26.06	*62.55*	4.89	58.86	64.05	62.38	73.40	64.67	9.01
	STA	76.72	94.96	57.48	69.91	33.16	25.86	*59.68*	2.02	57.52	65.87	59.72	61.44	61.14	5.48
	SEGA	**78.53**	94.90	**68.74**	73.65	40.23	**33.42**	*64.91*	7.25	66.14	67.27	**68.20**	64.68	66.57	10.91
	SEGA-f	77.99	95.30	65.54	**76.40**	44.02	31.15	***65.07***	7.40	**66.70**	**67.40**	65.20	**76.58**	**68.97**	13.31
100	none	78.99	95.50	75.66	72.09	42.24	40.45	*67.49*	–	63.58	63.55	75.02	64.06	66.55	–
	EDA	79.11	93.50	78.16	68.49	44.04	39.61	*67.15*	-0.34	65.06	47.03	76.22	54.84	60.79	-5.77
	BackTrans	80.50	94.50	79.64	71.93	45.00	41.46	*68.84*	1.35	71.00	**71.52**	76.42	61.36	70.08	3.52
	MLM	80.16	**96.20**	70.77	71.49	43.06	40.67	*67.06*	-0.43	74.20	66.42	68.62	60.46	67.43	0.87
	C-MLM	79.81	95.60	78.33	70.94	43.47	41.87	*68.34*	0.85	69.40	68.18	74.34	58.20	67.53	0.98
	LAMBADA	79.64	89.30	78.29	72.22	**49.44**	43.67	*68.76*	1.27	51.76	48.31	76.48	65.42	60.49	-6.06
	STA	80.73	95.30	80.33	73.46	45.62	40.72	*69.36*	1.87	70.64	62.93	76.14	64.52	68.56	2.01
	SEGA	79.97	94.82	**80.73**	75.53	49.31	47.54	*71.31*	3.83	74.94	63.84	76.06	65.86	70.17	3.62
	SEGA-f	**81.01**	94.76	77.31	**79.01**	49.11	**49.10**	***71.72***	4.23	**76.18**	65.93	**77.44**	**76.70**	**74.06**	7.51
200	none	78.57	**96.40**	**83.38**	79.20	47.74	51.71	*72.83*	–	65.20	57.09	78.00	77.56	69.46	–
	EDA	77.18	93.60	81.61	78.62	45.17	51.84	*71.34*	-1.50	70.50	57.57	78.06	75.74	70.47	1.01
	BackTrans	80.75	94.40	80.20	**81.38**	47.39	50.15	*72.38*	-0.46	67.70	51.84	77.10	78.24	68.72	-0.74
	MLM	81.71	96.10	80.14	80.71	46.43	51.46	*72.76*	-0.07	71.30	64.94	78.32	77.70	73.07	3.60
	C-MLM	**82.78**	**96.40**	81.25	80.37	44.56	51.11	*72.74*	-0.09	62.44	**70.61**	79.22	75.10	71.84	2.38
	LAMBADA	82.84	94.60	81.02	78.83	**51.09**	51.08	*73.24*	0.41	75.40	55.17	77.40	78.70	71.67	2.20
	STA	79.66	94.70	81.33	79.20	48.93	52.45	*72.71*	-0.12	67.20	63.42	78.90	77.64	71.79	2.33
	SEGA	80.94	95.60	81.73	79.70	50.87	53.40	*73.71*	0.87	70.84	61.23	77.10	80.58	72.44	2.98
	SEGA-f	81.00	96.00	82.49	79.40	50.80	**53.67**	***73.89***	1.06	**77.24**	63.61	**80.48**	**80.92**	**75.56**	6.10
500	none	81.48	96.86	85.12	80.46	**53.87**	57.22	*75.83*	–	65.90	68.18	80.36	79.22	73.42	–
	EDA	83.05	96.90	84.94	80.83	52.06	55.96	*75.62*	-0.21	76.40	61.06	80.76	77.48	73.93	0.51
	BackTrans	82.92	96.60	84.17	82.66	52.59	56.81	*75.96*	0.12	78.00	62.60	78.64	76.60	73.96	0.55
	MLM	79.98	96.40	83.58	82.09	52.11	56.08	*75.04*	-0.80	70.40	64.25	80.34	79.12	73.53	0.11
	C-MLM	82.63	96.48	84.06	82.59	52.64	55.95	*75.73*	-0.11	76.20	65.85	79.92	76.42	74.60	1.18
	LAMBADA	83.25	96.84	84.44	82.25	53.07	57.32	*76.20*	0.36	76.12	45.18	80.20	79.56	70.26	-3.15
	STA	**83.39**	**97.02**	84.90	81.97	52.51	57.56	*76.23*	0.39	**83.10**	68.43	77.32	80.40	77.31	3.90
	SEGA	82.65	96.88	84.22	80.99	53.11	57.44	*75.88*	0.05	78.46	**70.24**	78.52	80.51	76.93	3.52
	SEGA-f	83.14	96.96	**85.35**	**83.21**	52.55	**58.17**	***76.56***	0.73	81.24	67.70	**81.86**	**83.22**	**78.51**	5.09
1000	none	85.35	**97.12**	**86.41**	84.36	55.21	60.21	*78.11*	–	77.30	59.21	82.00	80.12	74.66	–
	EDA	83.20	96.44	86.28	83.85	53.80	59.57	*77.19*	-0.92	78.50	62.36	83.46	80.38	76.18	1.52
	BackTrans	84.71	96.34	85.17	84.06	54.38	60.14	*77.47*	-0.64	76.22	68.42	80.32	82.66	76.91	2.25
	MLM	85.90	96.60	84.95	84.70	54.07	60.27	*77.75*	-0.36	75.82	66.84	80.44	83.74	76.71	2.05
	C-MLM	85.62	96.56	85.33	84.15	54.23	59.39	*77.55*	-0.56	74.76	65.79	81.02	82.84	76.10	1.45
	LAMBADA	85.55	96.48	84.82	83.14	54.72	60.46	*77.53*	-0.58	80.72	51.25	79.76	83.10	73.71	-0.95
	STA	83.20	96.22	85.37	84.45	54.58	59.74	*77.26*	-0.85	78.50	63.43	81.50	84.08	76.88	2.22
	SEGA	85.05	96.48	85.36	84.22	54.47	60.09	*77.61*	-0.50	**83.98**	**71.68**	80.24	82.18	**79.52**	**4.86**
	SEGA-f	**85.94**	96.92	86.30	**84.84**	**55.64**	**61.98**	***78.60***	0.50	79.56	69.82	**82.26**	**84.40**	79.01	4.35

Table 7: The full results of the different augmentation methods on all text classification tasks under different resource levels. Testbed: DistilBERT-base classifier.

SEGA (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: **SEGA**'s augmentation efficiency on a single V100, with different settings.

sketch	<mask> **Conference on Empirical Methods** <mask> **submission of research papers** <mask> **Deep Learning** <mask>
SEGA 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>
SEGA 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 SEGA 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.
Back-Trans	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...
SEGA's sketch	*Sports:* <mask> lot of leagues <mask> in various sports <mask>
SEGA	~~*Sports:*~~ There are a lot of leagues in various sports. Here are some of the best.
SEGA-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.
SEGA-mixup's sketch	*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>
SEGA-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 SEGA 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
```

SEGA 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: SEGA generated samples for NER task. The last row shows an example where SEGA 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 SEGA:
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 SEGA:
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: SEGA generated samples for SQuAD. Given a paragraph, SEGA 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.