Text Grafting: Near-Distribution Weak Supervision for Minority Classes in Text Classification

Letian Peng, Yi Gu, Chengyu Dong, Zihan Wang, Jingbo Shang[*]
Department of Computer Science
University of California, San Diego
{lepeng, yig025, cdong, ziw224, jshang}@ucsd.edu

Abstract

For extremely weak-supervised text classification, pioneer research generates pseudo labels by mining texts similar to the class names from the raw corpus, which may end up with very limited or even no samples for the minority classes. Recent works have started to generate the relevant texts by prompting LLMs using the class names or definitions; however, there is a high risk that LLMs cannot generate in-distribution (i.e., similar to the corpus where the text classifier will be applied) data, leading to ungeneralizable classifiers. In this paper, we combine the advantages of these two approaches and propose to bridge the gap via a novel framework, *text grafting*, which aims to obtain clean and near-distribution weak supervision for minority classes. Specifically, we first use LLM-based logits to mine masked templates from the raw corpus, which have a high potential for data synthesis into the target minority class. Then, the templates are filled by state-of-the-art LLMs to synthesize near-distribution texts falling into minority classes. Text grafting shows significant improvement over direct mining or synthesis on minority classes. We also use analysis and case studies to comprehend the property of text grafting.

1 Introduction

Recent research has made rapid progress on extremely weak-supervised text classification (XWS-TC) (Wang et al., 2023), limiting the supervision to a brief natural-language description without any annotated samples. For example, text mining-based XWS-TC (Meng et al., 2020; Wang et al., 2021; Shen et al., 2021; Mekala et al., 2022; Zhao et al., 2023; Dong et al., 2023a) takes only class names or seed words from humans and discovers potential in-class texts following designated heuristics.

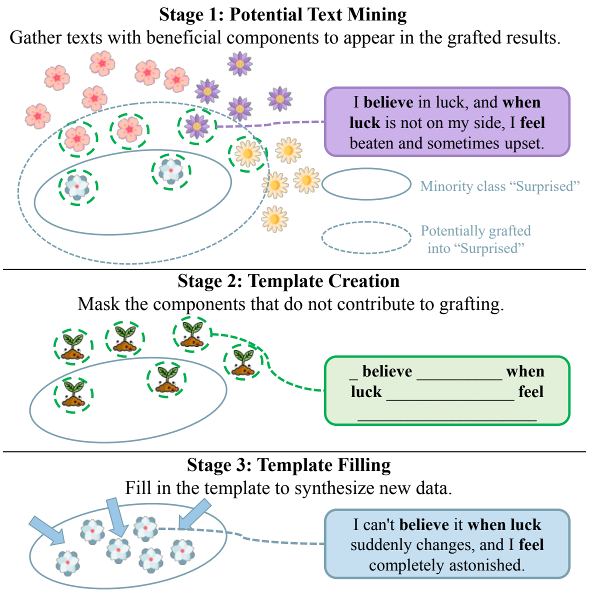

Stage 1: Potential Text Mining
Gather texts with beneficial components to appear in the grafted results.
I believe in luck, and when luck is not on my side, I feel beaten and sometimes upset.
Minority class "Surprised"
Potentially grafted into "Surprised"
Stage 2: Template Creation
Mask the components that do not contribute to grafting.
_ believe _________ when luck _______________ feel ____________________
Stage 3: Template Filling
Fill in the template to synthesize new data.
I can't believe it when luck suddenly changes, and I feel completely astonished.

Figure 1: The framework of text grafting.

Framework	Mining?	Train Data	Data Quality	In-Distribution
Text Mining	Text	Raw	Noisy	Yes
Data Synthesis	None	Generated	Clean	Hardly
Text Grafting (ours)	Template	Grafted	Clean	Mostly

Table 1: High-level comparison among three discussed XWS-TC frameworks.

Minority classes are arguably the most challenging part of XWS-TC. The class distribution in real-world datasets is often a long-tailed distribution (Zhang et al., 2023), with a non-trivial number of minority classes. These minority classes have a very small number of documents in the raw corpus, therefore, it is difficult to locate the right documents by mining-based methods, leading to noisy pseudo-labels. Under extreme circumstances, the mining-based methods may end up with no sample for minority classes.

A potential way to address this issue is data synthesis-based XWS-TC (Ye et al., 2022a,b; Peng and Shang, 2024), which hopes to generate in-class texts by prompting large language models (LLM) (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023a,b; Meta, 2024; Mesnard et al., 2024; OpenAI, 2024) with class names or defini-

[*] Corresponding author.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3741–3752
November 12-16, 2024 ©2024 Association for Computational Linguistics

tions. However, such synthesized texts may follow a distribution different from the corpus where the text classifier will be later applied (Mitchell et al., 2023), which makes the learned text classifier out-of-distribution, leading to poor performance.

This paper combines the advantages of mining-based and synthesis-based frameworks to propose a new framework, *text grafting*, which aims to obtain clean and near-distribution weak supervision for minority classes. As specified in Figure 1, text grafting incorporates three stages: (1) *Potential Text Mining* gathers raw texts with beneficial components to synthesize in-class texts for the target minority class. (2) *Template Creation* forms templates by masking the components that do not contribute to the in-class text synthesis. (3) *Template Filling* synthesizes in-class texts by filling in the blanks. Table 1 systematically compares the weak supervision obtained by different frameworks.

To identify the words not contributing to the classification, we borrow the marginalization idea from LLM reasoning (Holtzman et al., 2021). We get the probability logit of each word in the raw text by instructing LLMs (relatively small, specifically Gemma (Mesnard et al., 2024)) to generate with or without the in-class as a requirement. The difference between the two logits represents the potential of each word to appear in the grafted text. As only words with high potential will be left, we use the average potential of top-$K\%$ words to represent the text potential score. The bottom-$(100-K)\%$ words will be masked to form the template for data synthesis. We rank the templates by their potential scores and select top-$T\%$ templates for the last template-filling stage. Finally, these selected templates are filled by prompting a state-of-the-art LLM, GPT-4o (OpenAI, 2024).

We compare the three mentioned frameworks on various raw corpora to classify different minority classes. The experiment results show text grafting can outperform state-of-the-art text mining and dataset synthesis methods. The ablation study verifies that all stages and the intermediate template contribute to the success of our proposed text grafting. The mask-and-filling scenario also shows its advantage over simple in-context generation, since it forces the LLM to incorporate components from the raw texts. We also involve an extreme situation where the target class does not appear in the raw corpus completely. Remarkably, text grafting shows its robustness to this extreme situation, indicating its applicability does not require the target class to appear in the raw corpus. This enables text grafting to work on a very small corpus which boosts efficiency.

Furthermore, we analyze and discuss the property of text grafting. We apply principal component analysis to visualize that the drafted texts are indeed near in-distribution. We also find the grafted texts are near-distribution enough that we do not need to synthesize negative samples as in traditional data synthesis, which reduces the cost. We also conduct a comprehensive hyperparameter analysis of our method. Interestingly, we found that The mask ratio is searched to be better set to a high value like 0.75 and the mined template number can be as small as 200. These case studies explore the advantages of text grafting in distribution approximation and its failure when the raw texts are near the distribution of LLM generation.

We summarize our contributions as follows,

- We propose a novel XWS-TC framework for minority classes, text grafting, combining the in-distribution advantage of text mining and the in-class advantage of data synthesis.
- We implement text grafting following the marginalization idea from LLM reasoning, utilizing the probability logits for template mining and masking.
- We provide comprehensive analysis and case studies to show the strength, property, and possible failure of text grafting.[1]

2 Related Works

Extremely Weak-Supervised Text Classification (XWS-TC) needs only minimal human guidance to label the text, such as a few rules by human experts that match the text to the labels (Wang et al., 2023). Mainstream XWS-TC methods can be divided into two categories: **Text Mining** and **Data Synthesis**.

Text Mining is a fundamentak task (Han and Kamber, 2000) for natural language processing. In XWS-TC, the text miner follows high-level rules from humans to annotate raw texts, which are used to train the text classifier. A mainstream rule is whether a seed word appears in the raw text (Mekala and Shang, 2020; Meng et al., 2020; Wang et al., 2021), categorized as seed methods. Another mining way is to prompt language models for logits that reflect the probability of texts falling

[1] The datasets and models used in the experiments are released in github.com/KomeijiForce/TextGrafting

in classes (Brown et al., 2020), which can be calibrated by several techniques (Holtzman et al., 2021; Zhao et al., 2021; Han et al., 2023). The strong performance of existing text mining methods is highly dependent on the precision of the class-indicative rules (Dong et al., 2023a), which is hard to maintain for minority classes.

Data Synthesis (He et al., 2022) addresses the precision degradation in text mining by directly prompting LLMs with the label names to generate in-class texts (Ye et al., 2022a; Peng and Shang, 2024). With the powerful generative ability of LLMs, the synthesized texts are generally clean (in-class) for training strong classifiers. However, synthesized texts hold LLM-specific patterns, discovered by LLM-generated text detectors (Mitchell et al., 2023; Wu et al., 2023). This pattern is hard to be eliminated even with in-context learning (Koike et al., 2024). Thus, synthesized texts are generally out-of-domain and consequently fine-tune a weaker classifier on the test set.

Minority Classes widely appear in classification datasets as a result of long-tailed distribution (Zhang et al., 2023; Henning et al., 2023). For minority classes with supervised annotations, techniques like re-sampling (Shen et al., 2016; Pouyanfar et al., 2018; Tepper et al., 2020) and data augmentation (Wei and Zou, 2019; Juuti et al., 2020; Tian et al., 2021; Chen et al., 2021). However, these methods are applied to unbalanced annotations, which are unavailable under XWS.

Counterfactual Augmentation refers to generating annotated data out of the dataset or raw corpus. Different from regular augmentation, counterfactual augmentation changes the reference, e.g., label flipping (Zhou et al., 2022; Peng et al., 2023). Counterfactual augmentation is also applied for text-to-text tasks like translation (Liu et al., 2021) or summarization (Rajagopal et al., 2022). Counterfactual augmentation shares the same requirement for known reference as regular augmentation. This paper explores a counterfactual augmentation method for unannotated raw text under XWS.

3 Text Grafting

3.1 Preliminary

XWS Minority Class Classification takes a raw corpus $\mathcal{D} = \{X_{(i)}\}_{i=1:|\mathcal{D}|}$ and the target minority class name c as the input to train a binary classifier $f(X)$ that discerns a text falling in c or not. We denote the j-th word in the i-th text of the raw corpus as $x_{(i,j)}$.

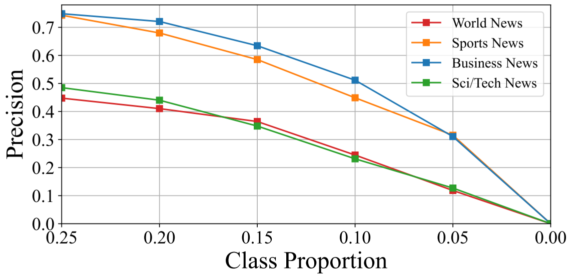


Figure 2: The precision of state-of-the-art text mining on same classes with different class proportions. "Precision" refers to the precision of the pseudo-labels. "Class Proportion" means the ratio of the texts of this class in the entire corpus after down-sampling.

Text Mining gathers in-class texts with high-level rules $g(X)$ that can precisely assign X to target class c. Example rules include whether X contains words indicating c (seed words) (Dong et al., 2023a) or X has top confidence to be in c by prompting LLMs (Brown et al., 2020) among $\mathcal{D}$. The mined $D^{(TM)} = \{X_{(i)}|g(X_{(i)})\}_{i=1:|\mathcal{D}|}$ is combined with some randomly sampled negative texts (due to the scarcity of c) to train $f(\cdot)$.

However, text miners fail in minority classes due to their low proportion in the raw corpus. By running a state-of-the-art text mining method (Dong et al., 2023a) on AG-News (Zhang et al., 2015) with class name proportion modified by sampling, we observe the mining precision drops sharply with the decrease of proportion, presented in Figure 2. Another concern is the class might be too minor that even no ground truth can be mined from the raw corpus, limiting the precision to 0% no matter how intuitive the mining rule is.

Data Synthesis does not annotate raw texts for classifier fine-tuning but directly prompts LLMs to generate in-class texts ($X' \sim \text{LLM}(I_c)$), where I_c is an instruction to write a text in class c. With the strong capability of state-of-the-art LLMs (OpenAI, 2024; Meta, 2024), the generated X' are highly confident to fall in class. Another advantage of data synthesis is the ability of LLMs to generate negative samples (Ye et al., 2022a; Peng and Shang, 2024). However, synthesized texts consist of patterns different from other sources (Mitchell et al., 2023), which indicates classifiers $f(\cdot)$ fine-tuned by synthesized texts are out-of-domain, consequently weaker in the classification task.

3.2 Overview of Text Grafting

As depicted in Figure 3, our text grafting is a hybrid method that combines the strengths of text mining and data synthesis. The core observation is that out-of-class texts can contain useful components for writing in-class texts. The text mining stage of text grafting aims to discover these potential components and formalize them as templates. In the data synthesis stage, the templates are filled by LLMs to produce in-class texts. With components from both raw texts and synthesis, the grafted texts are both in-class and near-distribution, which are supposed to fine-tune a better classifier than only text mining or data synthesis.

3.3 Implementation

In detail, the text mining stage includes **Potential Text Mining** and **Template Creation**, while in the data synthesis stage we conduct **Template Filling**. The text mining stage requires relatively small open-source LLMs with higher efficiency and accessible logits. Template Filling can utilize state-of-the-art LLMs even with API accessibility.

Potential Text Mining discovers texts with potential components to appear in the grafted texts. We evaluate the potential of each word $x_{(i,j)}$ in the raw text $X_{(i)}$ with regularized logits prompted from LLMs following the regularization idea in DC-PMI (Holtzman et al., 2021). The potential $\Delta p_{(i,j)}$ for $x_{(i,j)}$ is defined as the difference between the probability logit of $x_{(i,j)}$ prompted by an instruction with the class name (I_c) and an instruction for regularization (I_r). The difference can also be viewed as the probability of $x_{(i,j)}$ raised by incorporating the class name c into the instruction.

$$\Delta p_{(i,j)} = \log P_{\text{LLM}}(x_{(i,j)}|I_c) - \log P_{\text{LLM}}(x_{(i,j)}|I_r) \quad (1)$$

The words with top-$K\%$ Δp among the words in text X_i will remain in the template. Thus, the average of their Δp represents the potential (ΔP_i) of the template created based on X_i. As we are mining potential templates rather than directly in-class texts, the mining rate $K\%$ can be much larger than text mining.

$$\Delta P_i = \left\lceil \frac{1}{K\% \cdot |X_i|} \right\rceil \sum_{\Delta p_i \in \text{Top-}K\%(\Delta p_{1:|X_i|})} \Delta p_i \quad (2)$$

Then the texts are ranked by their grafting potential ΔP and texts with top-$N\%$ potential are mined to create the templates.

Function	Prompt
TM (I_c)	"Please write a <label> <style>."
TM (I_r)	"Please write a <style>."
DS	"Fill in the blanks in the template to produce a <label> <style>."

Table 2: The prompts used in text grafting. In prompts, **<label>** refers to the label names like "Surprised" while **<style>** represents the distribution like "Tweet".

Template Creation simply masks the words with bottom-$(100 - K)\%$ potential Δp by blank tokens "_" and uses the top-$K\%$ as template part. Text X_i is thus converted to template T_i, which is prepared for LLMs to fill in during the data synthesis stage. As the example in Figure 3, the components with the top potential to be in a grafted "Surprised" remain in the template such as "believe", "when luck", "feel". These components support the data synthesis to better write an in-class text while keeping the style in distribution with the writing structure from the raw corpus.

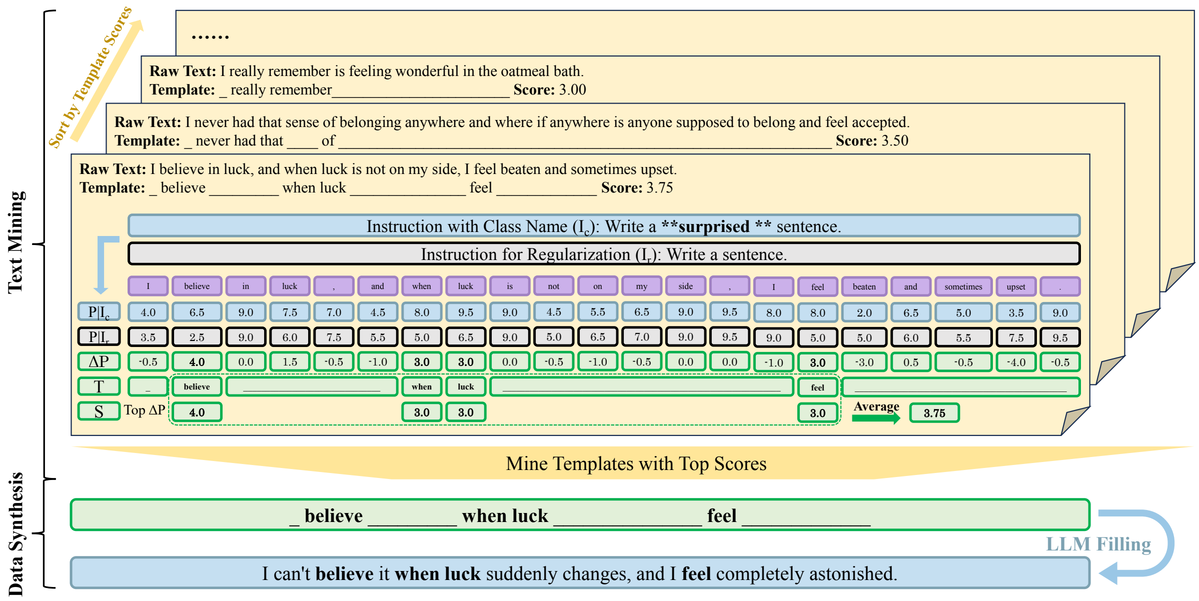
Template Filling prompts an LLM to fill in the blanks in T, which produces a grafted text that generally falls in the target class c. Referring to the example in Figure 3, the LLM well utilizes the writing structure in the template and fills in the blanks to produce the in-class text. As the template keeps the writing structure of the raw corpus, the grafted text is quite similar to the original one but flipped into the target minority class.

Specific prompts in these stages are shown in Table 2, where the label and distribution information is filled to support the text grafting.

4 Experiments

4.1 Evaluation

Datasets We take several minority classes from popular text classification datasets to evaluate the performance of different XWS-TC methods on minority classes. We include 1) TweetEval (Barbieri et al., 2020) and Emotion (Saravia et al., 2018), which contain minority emotion classes "Optimism" (8.9%) and "Surprised" (3.6%); 2) 20 News (Lang, 1995), which contains minority news topic "Religion" (3.3%) and "Politics" (4.1%); 3) BigPatent (Sharma et al., 2019), which contains minority patent class "Mechanical Engineering" (7.0%). The raw corpus is down-sampled to $10,000$ samples to improve experiment efficiency and save budget costs. We use the F1 score as the metric for evaluation.



Raw Text: I really remember is feeling wonderful in the oatmeal bath.
Template: _ really remember_________________________ **Score:** 3.00

Raw Text: I never had that sense of belonging anywhere and where if anywhere is anyone supposed to belong and feel accepted.
Template: _ never had that ____ of ____ **Score:** 3.50

Raw Text: I believe in luck, and when luck is not on my side, I feel beaten and sometimes upset.
Template: _ believe _________ when luck _____________ feel _____________ **Score:** 3.75

Instruction with Class Name (I_c): Write a ****surprised** ** sentence.

Instruction for Regularization (I_r): Write a sentence.

	I	believe	in	luck	,	and	when	luck	is	not	on	my	side	,	I	feel	beaten	and	sometimes	upset	.
$P\|I_c$	4.0	6.5	9.0	7.5	7.0	4.5	8.0	9.5	9.0	4.5	5.5	6.5	9.0	9.5	8.0	8.0	2.0	6.5	5.0	3.5	9.0
$P\|I_r$	3.5	2.5	9.0	6.0	7.5	5.5	5.0	6.5	9.0	5.0	6.5	7.0	9.0	9.5	9.0	5.0	5.0	6.0	5.5	7.5	9.5
ΔP	-0.5	**4.0**	0.0	1.5	-0.5	-1.0	**3.0**	**3.0**	0.0	-0.5	-1.0	-0.5	0.0	0.0	-1.0	**3.0**	-3.0	0.5	-0.5	-4.0	-0.5
T	_	**believe**	___				**when**	**luck**	___							**feel**	___				
S Top ΔP		**4.0**					**3.0**	**3.0**								**3.0**	Average → **3.75**				

_ believe _________ when luck _______________ feel _____________

I can't **believe** it **when luck** suddenly changes, and I **feel** completely astonished.

Figure 3: The overview of text grafting with the minority class **"Surprised"** in the Emotion dataset as an example. Text grafting includes two stages: **1) Text (Template) Mining:** Create scored templates and select the ones with the top scores. **2) Data Synthesis:** Prompt the LLM to fill in the templates to synthesize in-class texts.

Baselines We include various text mining and data synthesis methods as the baselines for comparison to illustrate the advantage of our text grafting.

Text mining methods include,

- **Prompting Confidence** (Brown et al., 2020), which is a prompting method that directly queries an LLM whether the text falls in the target minority class, and uses the probability logit of answering "yes" for ranking. Considering the class minority, the mining rate is set to 1%.
- **Debiased Seed Word** (Dong et al., 2023a), which is the current state-of-the-art XWS-TC method. This method uses a seed word (the same as the label name) to match the target minority class and then drops the seed word from the context to eliminate spurious correlation. Then the texts are filtered by text selection (Mekala et al., 2022) to produce the final mined texts.

Data synthesis methods include,

- **ZeroGen** (Ye et al., 2022a), which directly prompts the LLM to synthesize texts in or out of the target minority class.
- **In-Context Generation** (Dong et al., 2023b), which uses raw texts as the in-context examples to generate texts with a similar writing style as the raw corpus.
- **Incubator** (Peng and Shang, 2024), which uses instruction-tuned LLMs and in-context learning based on annotated instruction-to-dataset samples to generate data points for fine-tuning.

All text synthesis methods synthesize 1000 texts as positive (in the target minority class) or negative samples (out of the target minority class, 2000 in total).

The LLM used for text mining is a popular and advanced open-source LLM, Gemma (Mesnard et al., 2024) (`gemma-1.1-7b-it`) with accessible possibility logits. The LLM used for data synthesis is the state-of-the-art LLM, GPT-4o (OpenAI, 2024).

Grafting Hyperparameters The mining rates of our text grafter are set to 25% ($K\%$) for potential components in templates and 10% ($N\%$) for potential templates. Thus, the synthesized data number is less than 1000, not more than the data number from pure data synthesis.

Fine-tuning Hyperparameters We fine-tune a RoBERTa-Large (Liu et al., 2019) as the classifier with the AdamW (Loshchilov and Hutter, 2019) as the optimizer whose learning rate is initialized to 1×10^{-5}. The classifier is fine-tuned by 10 epochs with batch size 8 and 20% training data are split for validation to select the best-performing checkpoint. All the experiment results are achieved by an average of 5 runs. The two stages in text grafting apply the same LLM as text mining and data synthesis.

Dataset		**TWEET**	**PATENT**	**EMOTION**	**20NEWS**		**Average**
Distribution		Tweet	Patent	Tweet	News		
Minority Class		Optimism	Mechanical	Surprised	Religion	Politics	
Class Proportion		8.9%	7.0%	3.6%	3.3%	4.1%	
Supervised		45.88	34.30	32.28	24.10	32.27	35.14
Text Mining (TM)	Prompting Confidence	17.93	14.59	7.00	6.50	15.77	12.81
	Debaised Seed Word	19.15	20.46	8.78	11.47	19.53	15.88
Data Synthesis (DS)	ZeroGen	10.82	24.17	7.19	6.97	17.60	13.35
	Incubator	22.46	20.86	7.44	23.96	24.48	19.84
	In-Context Generation	16.24	24.53	22.24	21.98	24.13	21.83
TM+DS	Text Grafting **(Ours)**	**32.70**	**25.42**	**27.46**	**25.32**	**27.32**	**27.64**
Ablation	w/o Mining	26.54	16.74	24.32	17.69	15.16	20.09
	w/o Synthesis (DC-PMI)	17.86	11.34	7.34	4.33	4.28	9.03
	w/ Random Masking	30.11	19.07	23.37	23.57	26.65	24.55
	w/ MF → ICG	21.31	20.58	15.33	23.60	25.06	21.18
Zero-Occur	Debaised Seed Word	0.00	17.66	5.88	8.79	20.73	10.61
	In-Context Generation	18.84	23.15	19.50	20.63	24.11	21.25
	Text Grafting **(Ours)**	**30.61**	**25.27**	**31.08**	**26.15**	**25.54**	**27.73**

Table 3: Text mining performance (F1 Score) for minority classes among different datasets.

Method	**EMOTION**	**TNEWS**
Language	English	Chinese
Debaised Seed Word	19.14	22.84
+ Text Grafting	**31.30**	**28.61**

Table 4: Results (Macro F1 Score) on end-to-end XWS-TC for different languages. Emotion (English) contains minority classes "Surprised" and "Love" while TNEWS (Chinese) has a minority class "Stock".

4.2 Main Result

The main results from our experiments are presented in Table 3. The comparison inside text mining methods shows the advantage of the seed method over the prompt method, consistent with the findings of Wang et al.. The comparison among text synthesis methods reflects the importance of knowledge about the distribution of the corpus, as in-context generation outperforms other baselines with raw texts as an example for synthesis. Finally, text grafting outperforms all the baselines, which verifies the benefit of text grafting to produce in-class and near-distribution texts.

However, there is still a significant gap between the performance of supervised classification and XWS-TC even with text grafting. This indicates the grafted texts still have differences with the raw corpus distribution for further improvement.

4.3 Ablation Study

Table 3 also includes the ablation study results for text grafting in the *Ablation* columns. The first comparison focuses on the necessity of text mining and data grafting in the pipelines of text grafting. **Without Mining** removes the template score-based sorting and lets the LLM fill in randomly selected templates, which significantly underperforms the initial grafting. **Without Synthesis** does not create templates for data synthesis, but directly uses the Δp averaged over all words to mine texts for fine-tuning, equal to DC-PMI (Holtzman et al., 2021). The result is similar to the Prompting Confidence method, which shows the limitation of text mining for minority classes. Then we emphasize the necessity of intermediate templates. **With Random Masking** randomly masks the mined texts instead

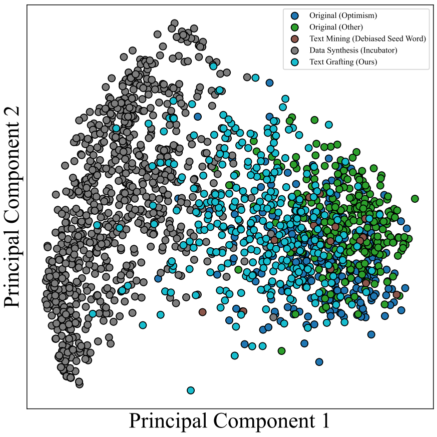


Figure 4: The visualization of text distributions from different methods.

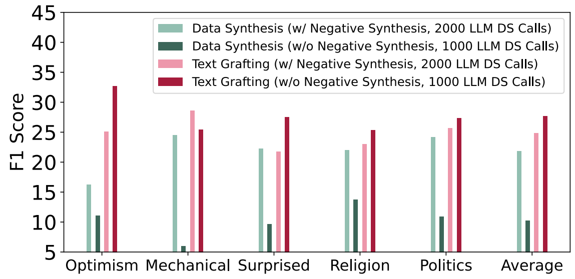


Figure 5: The analysis on the necessity of negative data synthesis.

of following the word-level potential Δp, which also results in a performance drop. **With Mask Filling → In-Context Generation** takes the mined texts as the in-context examples, which result in a similar performance as the one without mining, indicating the importance of template creation and filling. Based on these ablation results, our grafting framework is shown to be essential for achieving optimal performance by effectively combining data synthesis, text mining, and templates.

4.4 Further Analysis

Q1: How does Text Grafting Benefit End-to-End XWS-TC? Table 4 shows how text grafting can be integrated into end-to-end XWS-TC pipelines for different languages. We include the English Emotion dataset with "Surprised" and "Love" as the minority classes and the Chinese TNEWS dataset (Xu et al., 2020) with a minority class "Stock". For the minority classes, texts are synthesized by grafting while other classes apply the traditional debiased seed word method. The result shows text grafting improves end-to-end XWS-TC on different languages, which verifies the cross-lingual benefit of integrating text grafting into XWS-TC pipelines to handle minority classes.

Q2: What if the class proportion is 0%? In the *Zero-Occur* part of Table 3, we also include the discussed extreme situation when the raw corpus does not contain any text falling in the target minority class. A dramatic drop appears in the performance of text mining as there is no ground truth that any miner can get. The data synthesis and text grafting methods are robust to this change as they do not require the existence of ground truth examples. Thus, text grafting is verified to be applicable to raw corpus without the target minority class. Thus, text grafting can be based on a small subset of the corpus which might not contain the target minority class to boost efficiency.

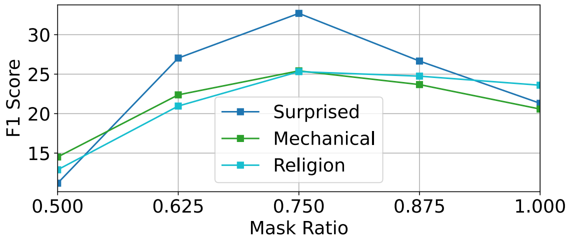


Figure 6: Analysis of the effect of mask ratio.

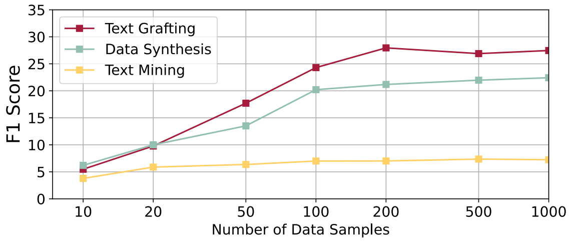


Figure 7: Analysis of the effect of data number.

Q3: How are grafted texts "near-distribution"? In Figure 4, we apply semantic text embeddings (Gao et al., 2021) to represent the texts mined or synthesized by different methods. These embeddings are then reduced to 2-dimension by principal component analysis (F.R.S., 1901) for visualization. We use the "Optimism" class of the TweetEval benchmark and compare the most competitive methods (Debiased Seed Word, Incubator, Text Grafting) of different frameworks. We can observe that text mining only discovers a limited proportion of in-class texts. The synthesized texts fall into a very different domain from the raw corpus, which fine-tunes an out-of-domain classifier with limited generalizability. In contrast, the grafted texts are much more near-distribution, contributing to the performance of the fine-tuned classifier.

Q4: Is Negative Data Synthesis Necessary? For data synthesis-based methods, the synthesis of negative data is an essential stage in the pipeline, which doubles the calls for LLM to synthesize texts. In text grafting, we efficiently use the raw texts as the negative examples. Thus, we explore the necessity of negative synthesis by evaluating the performance of data synthesis (In-Context Generation) and text grafting with or without negative data synthesis with the results presented in Figure 5.

Based on the results, we observe negative data synthesis is very necessary to pure data synthesis as the performance drops dramatically by removing

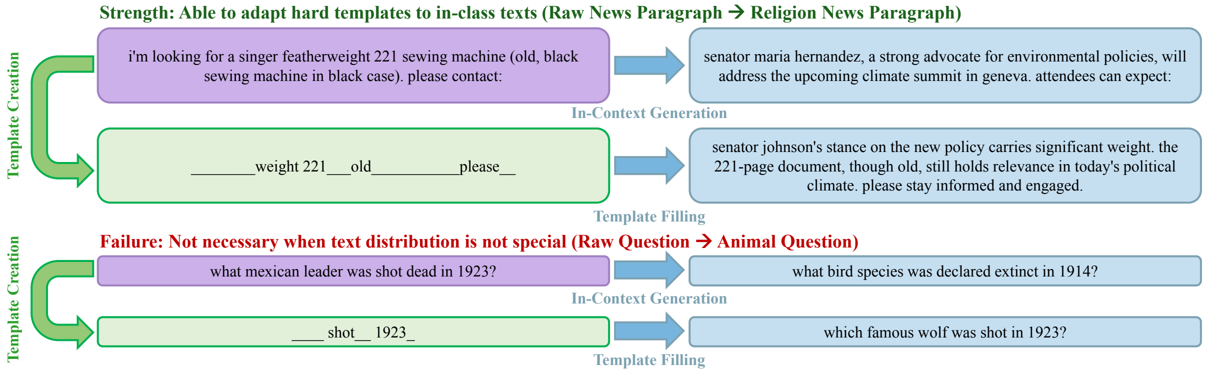

Strength: Able to adapt hard templates to in-class texts (Raw News Paragraph → Religion News Paragraph)
Template Creation
i'm looking for a singer featherweight 221 sewing machine (old, black sewing machine in black case). please contact:
senator maria hernandez, a strong advocate for environmental policies, will address the upcoming climate summit in geneva. attendees can expect:
In-Context Generation
________weight 221___old___________please__
senator johnson's stance on the new policy carries significant weight. the 221-page document, though old, still holds relevance in today's political climate. please stay informed and engaged.
Template Filling
Failure: Not necessary when text distribution is not special (Raw Question → Animal Question)
Template Creation
what mexican leader was shot dead in 1923?
what bird species was declared extinct in 1914?
In-Context Generation
____ shot__ 1923_
which famous wolf was shot in 1923?
Template Filling

Figure 8: A case study on the strength and possible failure of text grafting.

this stage. In contrast, text grafting without negative data synthesis works even better, indicating that our text grafting can work more efficiently by reducing the effort to call LLM at double times. We attribute this efficiency to the near-distribution property of the grafted texts, which makes the discrimination between them and the original raw texts no longer degrade to the classifying of text sources (Mitchell et al., 2023).

Q5: What mask ratio to choose? In Figure 6, we analyze the mask ratio used in text grafting. Within the considered set of mask ratios, $\{0.5, 0.625, 0.75, 0.875, 1.0\}$, the best-performing ratio is 0.75 among different datasets, the same as the setup in our experiments. We can also observe a trend of performance decrease when the mask ratio becomes away from 0.75. This indicates a too-high masking ratio will make the synthesized text deviate from the domain of raw corpus (100% leads to in-context generation). On the other hand, a too-low mask ratio will limit the synthesizer to generate in-class texts, which might cause more severe performance drops.

Q6: How many templates to mine? In Figure 7, we further analyze the necessary number of templates to train a strong classifier, which can guide the efficient application of text grafting. The result of the "surprised" class shows about 200 samples can reach the best performance, which results in about $0.2 budget for each class (OpenAI, 2024).

We also present how the efficiency of text mining (Debiased Seed Word) and data synthesis (In-Context Generation) is affected by sample numbers. Text mining cannot fine-tune a well-performing classifier due to severe noise in minority class mining. Data synthesis shows a similar scaling trend as text grafting but generally underperforms text grafting.

Function	Prompt
TM-V1 (I_c)	"Please write a <style> with attribute <label>."
TM-V1 (I_r)	"Please write a <style>."
TM-V2 (I_c)	"Provide me with a <label> <style>."
TM-V2 (I_r)	"Provide me with a <style>."
DS-V1	"Complete the empty fields in the template to generate a <label> <style>."
DS-V2	"Complete the template by filling in the blanks to create a <label> <style>."

Table 5: The prompt variants for robustness evaluation.

Variant	Original	V1	V2
DS	27.64	27.74	27.35
TM	27.64	28.12	27.79

Table 6: Results on prompt variants.

Q7: Is grafting robust to prompt templates? We rerun the experiments in Table 3 with prompt variants to evaluate the robustness of text grafting. We apply the prompt variants in Table 5 and illustrate the averaged F1 score over the 5 minority categories in Table 6. The result verifies our text grafting framework robust to specific prompt design.

5 Case Study

In Figure 8, we depict workflows of text grafting in comparison with in-context generation to illustrate the strength of grafting and possible failure.

Strength of text grafting is the ability of state-of-the-art LLMs to fill in hard templates as shown in the first case. While the template is not easy to be grafted into the target "Politics" class, the LLM comes up with the methodology to synthesize such a text. The text is also more similar in writing style to the original text than the in-context generation, which depicts the benefit from text grafting.

Failure of text grafting can happen when the corpus does not have a writing style very far from the way that LLMs can imitate. As shown in the second case, the LLM can synthesize the animal question without the intermediate template on the TREC corpus (Li and Roth, 2002), which reduces the necessity of text grafting. The XWS-TC of the minority class "Animal" on this corpus also shows a similar performance between data synthesis (F1 Score = 53.88) and text grafting (F1 Score = 53.46), which again emphasizes "near-distribution" to be an essential motivation to use text grafting.

6 Conclusion and Future Work

We introduced text grafting, a technique to generate in-distribution texts for minority classes using LLMs. By mining high-potential masked templates from the raw corpus and filling them with state-of-the-art LLMs, we achieve significant improvements in classifier performance on minority classes. Our analysis and case studies demonstrate the effectiveness of text grafting in enhancing text synthesis for minority classes. Future work will concentrate on improving the precision of template mining and the extension of text grafting to other tasks like information extraction.

Limitation

Despite the presented strengths in the paper, there are still several limitations in the text grafting pipeline. As a hybrid method, text grafting requires a large raw corpus more than data synthesis and LLM calls more than text mining. Other limitations of text grafting also succeed from text mining and data synthesis, such as the dependency on LLM ability (for mining and synthesis). Thus, the application scope for text grafting depends on how LLM comprehends the class name semantics. The performance of different classes might also be biased to the LLM ability in different classes.

References

Francesco Barbieri, José Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1644–1650. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Junya Chen, Zidi Xiu, Benjamin Goldstein, Ricardo Henao, Lawrence Carin, and Chenyang Tao. 2021. Supercharging imbalanced data learning with energy-based contrastive representation transfer. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21229–21243.

Chengyu Dong, Zihan Wang, and Jingbo Shang. 2023a. Debiasing made state-of-the-art: Revisiting the simple seed-based weak supervision for text classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 483–493. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023b. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Karl Pearson F.R.S. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Jiawei Han and Micheline Kamber. 2000. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2023. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2022. Generate, annotate, and learn: NLP with synthetic text. *Trans. Assoc. Comput. Linguistics*, 10:826–842.

Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 523–540. Association for Computational Linguistics.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7038–7051. Association for Computational Linguistics.

Mika Juuti, Tommi Gröndahl, Adrian Flanagan, and N. Asokan. 2020. A little goes a long way: Improving toxic language classification despite data scarcity. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2991–3009. Association for Computational Linguistics.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. OUTFOX: llm-generated essay detection through in-context learning with adversarially generated examples. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 21258–21266. AAAI Press.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Qi Liu, Matt J. Kusner, and Phil Blunsom. 2021. Counterfactual data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 187–197. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2022. LOPS: learning order inspired pseudo-label selection for weakly supervised text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4894–4908. Association for Computational Linguistics.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 323–333. Association for Computational Linguistics.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9006–9017. Association for Computational Linguistics.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`.

Letian Peng and Jingbo Shang. 2024. Incubating text classifiers following user instruction with nothing but LLM. *CoRR*, abs/2404.10877.

Letian Peng, Yuwei Zhang, and Jingbo Shang. 2023. Generating efficient training data via llm-based attribute manipulation. *CoRR*, abs/2307.07099.

Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S. Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, and Mei-Ling Shyu. 2018. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018, Miami, FL, USA, April 10-12, 2018*, pages 112–117. IEEE.

Dheeraj Rajagopal, Siamak Shakeri, Cícero Nogueira dos Santos, Eduard H. Hovy, and Chung-Ching Chang. 2022. Counterfactual data augmentation improves factuality of abstractive summarization. *CoRR*, abs/2205.12416.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. Taxoclass: Hierarchical multi-label text classification using only class names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4239–4249. Association for Computational Linguistics.

Li Shen, Zhouchen Lin, and Qingming Huang. 2016. Relay backpropagation for effective learning of deep convolutional neural networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, volume 9911 of *Lecture Notes in Computer Science*, pages 467–482. Springer.

Naama Tepper, Esther Goldbraich, Naama Zwerdling, George Kour, Ateret Anaby-Tavor, and Boaz Carmeli. 2020. Balancing via generation for multi-class text classification improvement. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1440–1452. Association for Computational Linguistics.

Jiachen Tian, Shizhan Chen, Xiaowang Zhang, Zhiyong Feng, Deyi Xiong, Shaojuan Wu, and Chunliu Dou. 2021. Re-embedding difficult samples via mutual information constrained semantically oversampling for imbalanced text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3148–3161. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3043–3053. Association for Computational Linguistics.

Zihan Wang, Tianle Wang, Dheeraj Mekala, and Jingbo Shang. 2023. A benchmark on extremely weakly supervised text classification: Reconcile seed matching and prompting approaches. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3944–3962. Association for Computational Linguistics.

Jason W. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. 2023. A survey on llm-generated text detection: Necessity, methods, and future directions. *CoRR*, abs/2310.14724.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 4762–4772. International Committee on Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022a. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11653–11669. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2022b. Progen: Progressive zero-shot dataset generation via in-context feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3671–3683. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep long-tailed learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(9):10795–10816.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2023. Pre-trained language models can be fully zero-shot learners. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15590–15606. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8646–8665. Association for Computational Linguistics.