Generative Deduplication For Socia Media Data Selection

Xianming Li [1], **Jing Li** [1,2*]
[1] Department of Computing
[2] Research Centre on Data Science & Artificial Intelligence
The Hong Kong Polytechnic University
xianming.li@connect.polyu.hk, jing-amelia.li@polyu.edu.hk

Abstract

Social media data exhibits severe *redundancy* caused by its noisy nature. It leads to increased training time and model bias in its processing. To address this issue, we propose a novel Generative Deduplication framework for social media data selection by removing semantically duplicate data. While related work involves data selection in task-specific training, our model acts as an efficient pre-processing method to universally enhance social media NLP pipelines. Specifically, we train a generative model via self-supervised learning to predict a keyword to capture the semantics of noisy social media text for deduplication. Meanwhile, time-dimensional Gaussian noise is added to improve training complexity and avoid learning trivial features. Extensive experiments suggest that our model can better reduce training samples while improving performance than baselines. The results show our model's potential to broadly advance social media language understanding in effectiveness and efficiency. [1]

1 Introduction

Social media is an abundant resource with vast real-time user-generated content, providing valuable insights into the world and society. It has benefited various applications, such as stance detection (Glandt et al., 2021) and content recommendation (Zeng et al., 2020), taken advantage of cutting-edge NLP practices. However, a common challenge NLP models may face is the severe *redundancy* of social media data (Tao et al., 2013) caused by its noisy nature (Zhang et al., 2023). Here, we define *redundancy* as semantically similar content that leads to information overload and model bias.

The redundant data not only increases the training cost of a model (in time and resources) but also results in the **redundancy bias** adversely affecting its performance (Lee et al., 2022). To illustrate this point, we show some TweetEval examples in Figure 1. The duplicated tweets containing "pumpkin" and "neutral" sentiment labels bias the model to connect "pumpkin" to "neutral wrongly" sentiment, meanwhile rendering unnecessary training costs.

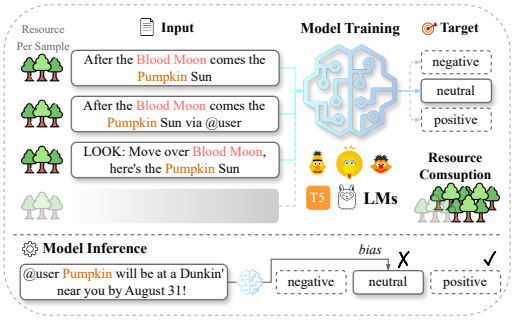


Figure 1: An redundancy example on TweetEval (Barbieri et al., 2020) sentiment analysis. The green trees denote training costs. The duplicated tweets biased the models to connect "pumpkin" to "neutral" sentiment.

To tackle the redundancy problem, we explore the solution from *data selection* (Liu et al., 2019a; Paul et al., 2021; Lee et al., 2022; Xie et al., 2023). The goal is to select a subset of relevant data from a larger dataset to benefit the model training. It has profoundly affected the performance of large language model pretraining (Xie et al., 2023) and its downstream task finetuning (Yu et al., 2023).

In data selection, most work focuses on how to find useful data via augmentation or retrieval for "data addition" (Axelrod et al., 2011; Ruder et al., 2017; Liu et al., 2019a; Xie et al., 2023). In contrast, others center on "**data deduplication**" to remove semantically duplicated data and show its positive effects in pretraining (Lee et al., 2022).

Our work aligns with the "deduplication" line. Previous work has shown its help in social media NLP, such as shingles (Broder, 1997) and simhash (Manku et al., 2007). However, these efforts rely on surface linguistic features rather than semantics.

*Corresponding author
[1]Code is available at: https://github.com/4AI/generative_deduplication.

Findings of the Association for Computational Linguistics: EACL 2024, pages 5765–5776

While textual semantic similarity models (Reimers and Gurevych, 2019; Gao et al., 2021; Li and Li, 2023) can be easily applied to the deduplication, the pairwise comparison would render high complexity. Moreover, our work differs from existing efforts (Xia et al., 2024) to engage data selection in task-specific training. Instead, we aim to provide a pre-processing method for social media data, which can be easily plugged into various NLP pipelines.

To that end, we propose a novel semantic deduplication approach, **Generative Deduplication**, for social media data selection. Specifically, we adopt a generative model as the generative backbone and train it with a self-supervised task to generate a keyword from the input text. Here, we train the generative backbone for only one epoch. Duplicate text undergoes multiple optimizations, enabling more accurate keyword prediction than non-duplicate text with a single optimization. Moreover, we improve the training difficulty with Time-dimensional Gaussian Noise (TGN) to prevent trivial feature learning in one epoch, limiting keyword prediction for non-duplicates. Hence, we can consider samples with correct keyword prediction as duplications and remove them. This way, we allow a computational complexity of $O(n)$, where n is the data size, and avoid pairwise comparison.

To the best of our knowledge, *we are the first to propose Generative Deduplication for social media data selection and study its broad impact on downstream social media language understanding.*

In experiments, the deduplication experiment indicates that our model enables the best data quality with less training time. Then, the results on the TweetEval benchmark show that our selected data allows performance gains with much shorter training time on varying downstream models and tasks. For example, for LLaMA on sentiment analysis, our model reduces the training set (50.9%) and training time (42.9%) yet improves the macro recall from 73.0 to 73.5. Next, the ablation study shows the positive contributions of varying modules. The general short text classification experiments suggest that our method also benefits general scenarios. Lastly, we interpret our model's superiority with more analyses.

In summary, our contributions are as follows:

• We explore the redundancy issue in social media data and disclose its effects on biasing models.

• We propose a novel generative deduplication model to shortlist data and tackle redundancy bias.

• Extensive experiments reveal generative deduplication can help reduce redundancy and broadly improve social media language understanding.

2 Related Work

The proposed Generative Deduplication is in line with data selection. It is an essential technique for selecting helpful data to benefit the training of downstream tasks. Many previous studies center on domain adaptation (Moore and Lewis, 2010; Feng et al., 2022; Xie et al., 2023), where they selected data that aligns with the target distribution from vast data, aiming to improve performance in specific domains. More recently, *deep learning* techniques (Coleman et al., 2020; Mindermann et al., 2022) have been used for better data selection.

Given recent language model advances, some work, such as (Yao et al., 2022; Schoch et al., 2023), explored *retrieval* and *augmentation* to "select and add" relevant data from external resources for task-specific training. However, prior work observed these practices may result in a significant amount of duplicate data, which introduces redundancy (Xie et al., 2023) and adversely affects performance (Hernandez et al., 2022). To mitigate this redundancy issue, **deduplication** (Tirumala et al., 2023) can be applied to shortlist duplicate data to allow more effective and efficient training.

Our work aims to adopt deduplication to address redundancy bias (Tao et al., 2013; Zhang et al., 2023) in social media data. However, existing deduplication methods (Broder, 1997; Manku et al., 2007; Hajishirzi et al., 2010) mainly rely on surface linguistic features, unable to handle semantic-level duplication prevalent in noisy social media data. Meanwhile, directly applying models based on textual semantic similarity (Reimers and Gurevych, 2019; Li and Li, 2023, 2024b) may involve pairwise comparison and result in high deduplication cost. In addition, existing methods engage data selection in end-to-end task-specific training (Xia et al., 2024). In contrast, our work focuses on data selection for pre-processing, which can be seamlessly integrated into various NLP pipelines.

3 Generative Deduplication

This section will elaborate on the proposed generative deduplication with an overall framework in Figure 2. We will first introduce the problem formulation in Section 3.1. Then, we describe the generative training in Section 3.2, followed by the inference as the deduplication stage in Section 3.3.

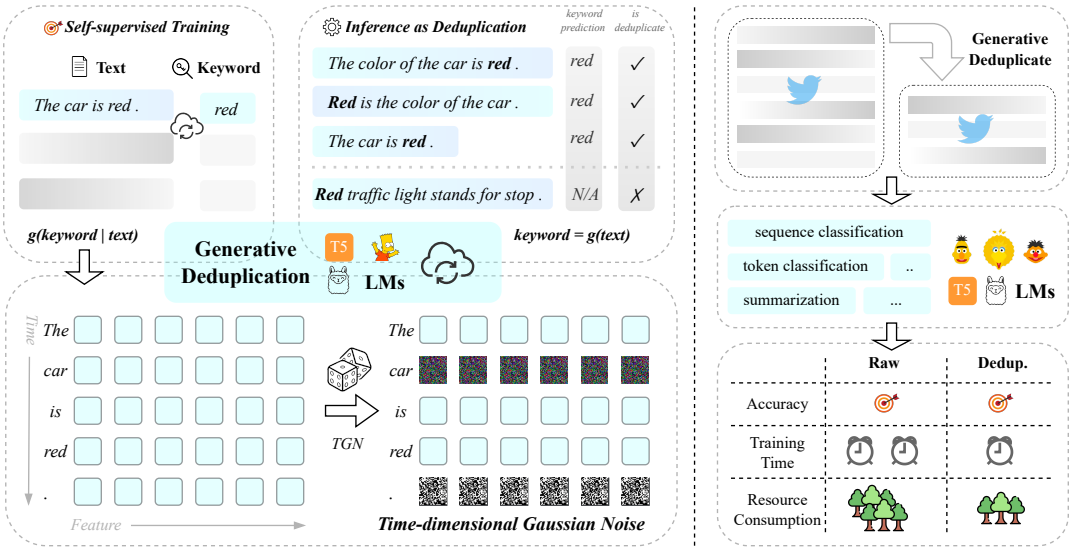


Figure 2: The generic framework of the proposed generative deduplication (GD). It includes two parts. 1) The left part shows the detailed GD process. In the *self-supervised training stage*, the model encodes input text and learns to generate the corresponding keyword. The *Time-dimensional Gaussian Noise (TGN)* will be applied to increase training difficulties and avoid trivial feature learning in the training stage. In the *inference as deduplication stage*, text with correct keyword prediction is identified as a duplicate. 2) The right part depicts the downstream applications. First, the training set is deduplicated using GD. Then, the deduplicated training set is used for training and inference. By doing so, It can reduce training samples and resource consumption while improving accuracy.

3.1 Problem Formulation

Given a corpus $\mathcal{C}$ consisting of n texts, $\{t_1, ..., t_n\}$, generative deduplication aims to identify semantically duplicate texts using generative models. The identified duplicate texts are then removed, forming a smaller corpus $\mathcal{SC}$ comprising m texts, where $m \leq n$. To achieve this, two stages are involved: generative training and inference. We will discuss them in subsequent sections.

3.2 Generative Training

The generative deduplication involves two important designs: generative self-supervised training to predict keywords and adding time-dimensional Gaussian noise to increase training difficulty.

Generative Self-supervised Training. We employ a novel self-supervised learning task of *keyword generation* for social media texts. We are motivated by the noisy nature of social media data, and the inherent data sparsity can limit the explicit indicators of semantic similarity (Zeng et al., 2018). Keywords as condensed post-level representations can bridge the gap and enable better exploration of semantic similarity for deduplication purposes.

To implement this, we first apply the popular toolkit KeyBERT (Grootendorst, 2020) for keyword extraction. Then, the extracted keyword serves as the target. Specifically, the contextual representations of text t_j in $\mathcal{C}$ is obtained as follows: $\mathbf{H} = \mathrm{g}(t_j)$, where $\mathrm{g}(\cdot)$ represents the generative backbone. For each training sample in $\mathcal{C}$, the objective is to minimize the sum of the negative likelihood of keyword tokens $\{k_1, ..., k_l\}$, where l is the length of the keyword tokens, as follows:

$$\mathcal{L}_g = -\sum_{i=1}^{l} \log \mathrm{g}_\theta(k_i | t_j; k_0, k_1, ..., k_{i-1}). \quad (1)$$

Here, θ is the learnable parameters, t_j is the j-th input text in $\mathcal{C}$, k_0 denotes the pre-defined start token, and $\mathrm{g}(\cdot)$ represents the generative backbone. Notably, the self-supervised training runs only one epoch for a sufficiently large prediction gap of duplicate and non-duplicated data (see Section 3.3).

TGN: Time-dimensional Gaussian Noise. According to the scaling law (Kaplan et al., 2020), large generative models possess exceptional language understanding capabilities due to the large-scale pre-training. However, this can negatively affect deduplication performance because even non-duplicate texts might have accurate keyword predictions in one-epoch training, hindering the separation of duplicate and non-duplicate data.

To address this concern, we propose a novel TGN to add noise and increase training difficulties. It aims to avoid learning trivial features to limit the keyword prediction for non-duplicate data. It first generates binary masks $\mathbf{M}$ using the Bernoulli distribution, where each time step is assigned a value of 0 or 1 based on a given probability p:

$$\mathbf{M} \sim \text{Bernoulli}(p) \tag{2}$$

Then, the time steps with a mask value of 1 are selected, and their corresponding features are entirely replaced by Gaussian noise. This process is shown in Figure 2, and the equation as follows:

$$\begin{aligned} \widehat{\mathbf{H}} &= \mathbf{H} \odot (1 - \mathbf{M}) + \mathbf{M} \odot \mathbf{G} \\ \mathbf{G} &\sim \mathcal{N}(\mu, \sigma^2), \end{aligned} \tag{3}$$

where $\mathbf{G}$ denotes the standard Gaussian distribution, which has the mean μ and standard deviation σ.

3.3 Inference as Deduplication

After the one-epoch self-supervised training with TGN, the model makes inferences to predict keywords for deduplication. The trained generative backbone generates keywords for all texts in $\mathcal{C}$ using beam search during this stage. Specifically, for text t_i in $\mathcal{C}$, its keyword is generated as follows:

$$\widehat{K} = \text{g}(t_i; b), \tag{4}$$

where b is the beam size for beam search. Here, we consider the text duplicates if they can accurately replay the target keyword through the trained generative backbone. Our intuition is that the model is more likely to comprehend semantically duplicate texts than non-duplicates because of the multiple optimizations to the former. Consequently, the duplicate text will result in higher chances for the model to replay the keywords after one-epoch training. Based on this, we compare the generated keyword with the target keyword for deduplication:

$$\text{IsDup}(t) = \begin{cases} 1 & if\ \widehat{K} = K \\ 0 & otherwise \end{cases} \tag{5}$$

K is the target keyword for t, $\widehat{K}$ is its predicted keyword from the generative backbone $\text{g}(\cdot)$, and 1 and 0 indicate "yes" and "no" for deduplication.

4 Experimental Setup

Datasets. For the **deduplication experiment**, we use the MRPC dataset (Dolan and Brockett, 2005), where texts are labeled as either *equivalent* or *non-equivalent* based on semantic duplication. For **social media language understanding**, we evaluate our model on 7 widely used Twitter datasets, including 6 tasks from TweetEval (Barbieri et al., 2020): Emoji Predication, Hate Speech Detection, Offensive Language Identification, Sentiment Analysis, Stance Detection, and Irony Detection, and Humor Detection from SemEval-2021 task 7 (Meaney et al., 2021). The statistics of adopted datasets are listed in Table 1. Due to the limitations of space and computational resources, we use the Emoji dataset in the ablation study and discussion. It is more challenging (with 20 labels) and larger than other datasets, making it ideal for evaluating the performance of the proposed model.

Dataset	Train	Valid.	Test	Labels
Emoji	45,000	5,000	50,000	20
Hate	9,000	1,000	2,970	2
Offensive	11,916	1,324	860	2
Sentiment	45,389	2,000	11,906	3
Stance	2,620	294	1249	3
Irony	2,862	955	784	2
Humor	8,000	1,000	1,000	2

Table 1: Number of labels and instances in training, validation (**valid.**), and test sets for the adopted datasets.

Evaluation Metrics. For **deduplication**, we report the F1 score for *equivalent* (duplicate text), marked as F1^{eq}. We also report the deduplication time in seconds. For **social media language understanding**, we follow prior works (Barbieri et al., 2020; Meaney et al., 2021) to report the same evaluation metrics from the original tasks. Specifically, we report macro F1 for Emoji, Hate, Offensive, and Humor, report micro-Recall for Sentiment, report the average of the F1 of *against* and *favor* (marked as $\text{F1}^{a+f}/2$) for Stance, and report the F1 of the *ironic* label (marked as F1^{i}) for Irony.

Baselines and Comparisons. For the **deduplication experiment**, we compare the proposed generative deduplication with two widely used deduplication approaches: shingles (Broder, 1997) and simhash (Manku et al., 2007). We also compare it with pairwise semantic deduplication using pretrained sentence embeddings (Li and Li, 2023, 2024a). For **social media language understanding**, we adopt three popular backbones: RoBERTa (Liu et al., 2019b), BERTweet (Nguyen et al., 2020), and LLaMA (Touvron et al., 2023) and compare with them using deduplicated data.

Model Settings. For generative deduplication, we use the T5-base model as the default generative backbone with a learning rate of $1e^{-4}$, beam size of 1, and a prediction threshold of 0.5. For social media language understanding tasks, we use RoBERTa-base, BERTweet-base, and LLaMA-7B as backbones. For efficient training, we employ the LoRA (Hu et al., 2021) technique for LLaMA-7B fine-tuning with specific parameters ($lora\_r = 16$, $lora\_alpha = 16$, and $lora\_dropout = 0.1$). The batch size is chosen from values $\{16, 32, 64, 128\}$ on the validation data. The initial learning rate is $2e^{-5}$ for BERT/RoBERTa-based models and $2e^{-4}$ for LLaMA-based models.

Dataset Deduplication. Here, we first employ various deduplication approaches, including shingles, pairwise semantic deduplication, and the proposed generative deduplication, to eliminate duplicate training samples from social media language understanding datasets. Table 2 shows the deduplicated training data size and time consumption of each dataset. We can see that the proposed generative deduplication is more effective at removing redundant data than baselines with fewer training samples after deduplication. Also, the generative deduplication is efficient. It achieves competitive deduplication time with shingles and is highly efficient compared to pairwise deduplication. Moreover, we perform random deduplication for a comprehensive evaluation. We randomly reduce the dataset size to match that of the generative deduplication. These deduplicated datasets by different approaches will be used for social media language understanding in Section 5. Note that we only deduplicate the train set for model training, while keeping the validation and test sets for evaluation.

Dataset	**Raw**	**SD**	**PD**	**GD**
Sentiment	45,615	43,951	35,605	22,418
	time →	1,074	29,212	808
Emoji	45,000	40,656	35,170	31,425
	time →	1,043	29,047	901
Offensive	11,916	11,013	10,771	9,595
	time →	78	2,099	240
Hate	9,000	8,810	7,818	8,061
	time →	50	1,251	178
Humor	8,000	7,720	7,848	7,537
	time →	39	964	153
Irony	2,862	2,841	2,712	2,472
	time →	9	135	59
Stance	2,620	2,554	2,547	956
	time →	11	113	57
Total	125,013	117,545	102,471	82,464
	time →	2,304	62,771	2,396

Table 2: Deduplicated training data size and deduplication time (in seconds) of different approaches on different datasets. Raw is the original size. SD means Shingles Deduplication. PD stands for Pairwise Deduplication. GD is the proposed generative deduplication

Model	**F1**^{eq} ↑	**Time** (s) ↓
shingles	32.9 ± 0.0	11.9
simhash	24.9 ± 0.0	**1.4**
pairwise semantic dedup.	51.7 ± 0.0	42.7
Generative Dedup. (T5-small)	52.7 ± 0.3	5.6
Generative Dedup. (T5-base)	**56.3** ± 0.2	18.8

Table 3: Deduplication performance on MRPC dataset. Bold indicates the best results. ↑ means the higher, the better. ↓ stands for the smaller, the higher.

5 Experimental Results

5.1 Deduplication Results

The deduplication results are presented in Table 3. Our proposed generative deduplication outperforms the baselines. Notably, the T5-base generative deduplication achieves a 23.4% improvement in F1^{eq} compared to shingles. Similarly, the T5-small generative deduplication shows a 19.8% F1^{eq} gain and reduces the deduplication time by 6.3 seconds compared to shingles. These improvements are attributed to the ability of generative deduplication to understand and remove semantic duplicates. In contrast, shingles and simhash, which focus on surface linguistic features, struggle with semantic-level duplication. Furthermore, the proposed generative deduplication outperforms pairwise semantic deduplication and is more efficient. For instance, the T5-base generative deduplication is faster, taking 18.8 seconds compared to 43.7 seconds for pairwise semantic deduplication.

5.2 Main Results

We show the main experimental results of social media language understanding tasks in Table 4 and draw the following observations.

First, we can see that LLaMA-based models outperform RoBERTa and BERTweet-based models. This can be attributed to its larger model scale and powerful language understanding capability. Second, we can find that the performance using random deduplication (RD) data is poorer than using other deduplication data. It negatively impacts per-

Model	Emoji	Hate	Offensive	Humor	Sentiment	Stance	Irony	Avg.
	Macro F1 ↑				Macro Recall ↑	$F1^{a+f}/2$ ↑	$F1^{i}$ ↑	
RoBERTa								
Raw	30.9 ± 0.2 ♢	46.6 ± 1.8 ♢	79.5 ± 0.7 ♢	95.0 ± 0.6 †	71.3 ± 1.1 ♢	68.0 ± 0.8 ♢	59.7 ± 5.0 ♢	64.4
RD	29.8 ± 0.3	45.7 ± 1.6	79.0 ± 0.9	93.3 ± 0.5	71.5 ± 1.3	65.1 ± 1.2	58.5 ± 2.1	63.3
SD	31.0 ± 0.3	47.3 ± 1.3	79.4 ± 0.6	93.6 ± 0.5	71.5 ± 1.3	67.9 ± 0.9	61.3 ± 3.7	64.6
PD	31.1 ± 0.3	47.7 ± 1.3	79.7 ± 0.7	94.0 ± 0.3	71.5 ± 1.1	68.1 ± 0.8	62.0 ± 2.7	64.9
GD	31.4 ± 0.2	49.5 ± 1.1	80.7 ± 0.7	94.3 ± 0.4	71.8 ± 1.1	68.3 ± 0.6	62.6 ± 1.8	65.5
BERTweet								
Raw	32.3 ± 0.5	54.9 ± 0.9 †	80.5 ± 0.8 †	$\mathbf{95.9} \pm 0.3$ †	72.3 ± 1.2	70.3 ± 0.9 †	78.7 ± 1.4 †	69.3
RD	31.2 ± 0.7	54.5 ± 0.9	80.2 ± 1.0	94.5 ± 0.4	71.4 ± 1.7	65.9 ± 1.3	77.9 ± 1.5	67.9
SD	32.4 ± 0.5	55.0 ± 0.8	80.5 ± 0.8	94.5 ± 0.3	72.1 ± 1.1	69.9 ± 1.1	78.7 ± 1.4	69.0
PD	32.5 ± 0.4	55.3 ± 1.0	80.6 ± 1.0	94.5 ± 0.5	72.2 ± 1.3	69.4 ± 1.0	79.1 ± 1.5	69.1
GD	32.6 ± 0.3	55.7 ± 1.0	80.9 ± 0.6	95.0 ± 0.3	72.2 ± 1.0	69.3 ± 0.8	$\mathbf{80.1} \pm 1.2$	69.4
LLaMA								
Raw	$\mathbf{37.4} \pm 0.6$	58.2 ± 1.3	80.7 ± 1.2	95.3 ± 0.5	73.0 ± 1.2	70.1 ± 0.7	74.5 ± 1.1	69.9
RD	36.2 ± 1.1	57.7 ± 1.8	79.8 ± 1.3	95.1 ± 0.8	71.6 ± 1.6	66.5 ± 0.9	73.2 ± 1.6	68.6
SD	37.1 ± 0.9	58.1 ± 1.3	80.3 ± 1.4	95.2 ± 0.6	73.1 ± 1.3	70.4 ± 0.6	75.6 ± 1.3	70.0
PD	37.3 ± 1.1	58.3 ± 1.4	80.8 ± 1.3	95.2 ± 0.6	73.0 ± 1.4	70.8 ± 0.8	75.9 ± 1.5	70.2
GD	37.3 ± 0.8	$\mathbf{58.6} \pm 1.4$	$\mathbf{81.0} \pm 1.3$	95.3 ± 0.6	$\mathbf{73.5} \pm 0.9$	$\mathbf{71.1} \pm 1.1$	76.4 ± 1.2	**70.5**

Table 4: Results of social media language understanding tasks. ♢: results are from Barbieri et al. (2020). †: results are retrieved from Tan et al. (2023). We follow previous work to report the average result of five runs. "Raw" refers to the use of the original train set. "RD", "SD", "PD", and "GD" are trained on the deduplicated training set of random, shingles, pairwise semantic, and generative deduplication, respectively. The light blue color indicates the best results for each backbone, while the bold marks the best overall results.

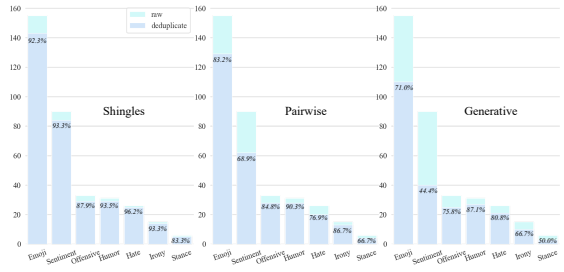


Figure 3: A bar plot shows the training time of RoBERTa-based models for each dataset. The percentage represents the training time on deduplicated data compared to the training time on raw data.

formance. This might be because it eliminates non-duplicate patterns randomly and fails to address redundancy bias. Third, it can be seen that shingles deduplication (SD) and pairwise semantic deduplication (PD) slightly improve the performance. This could be attributed to their ability to remove duplicate data partially and mitigate redundancy bias. Fourth, our proposed generative deduplication (GD) consistently outperforms baselines trained on raw and other deduplicated training sets in average scores. Notably, generative deduplication generally performs better than shingles and pairwise semantic deduplication in various tasks, except for Stance detection using BERTweet. We will explain this exception in Section 5.5. Generative deduplication outperforms shingles because it can identify and remove semantically duplicate data, effectively reducing redundancy bias. Also, generative deduplication is more effective than pairwise semantic deduplication due to its stronger language understanding capabilities enabled by generative models. It can better identify and remove semantically duplicate data, further mitigating redundancy bias.

Furthermore, we compare the training time using different deduplicated datasets in Figure 3. The generative deduplication can significantly reduce training time. This highlights generative deduplication's superior efficiency.

5.3 Ablation Study

We have demonstrated the overall effectiveness of generative deduplication in the main results. Here, we further test its different settings via the ablation study. The results are presented in Table 5.

T5-base generative deduplication outperforms T5-small and can reduce more training samples. It is attributed to T5-base's more powerful language understanding capabilities (with a larger model size), allowing it to handle noisy social media data more effectively.

The results of generative deduplication with and without TGN indicate that TGN can improve generative deduplication performance. This highlights

Model	Train Size	Macro F1 ↑
Generative Dedup. (T5-base)	$31.4K$	**31.4**
w/o TGN	$30.6K$	30.1
epoch=2	$29.7K$	29.8
Generative Dedup. (T5-small)	$36.2K$	30.0
w/o TGN	$34.6K$	28.7
epoch=2	$34.4K$	28.3

Table 5: Ablation study of generative deduplication on the **Emoji** prediction task. K represents thousands. The train size of the raw **Emoji** is $45K$. K stands for thousand. Bold indicates the best results.

Model	AGNews		Subj		SST-2	
	data ↓	acc ↑	data ↓	acc ↑	data ↓	acc ↑
RoBERTa	120000	94.88	8000	96.60	6920	94.29
w/ RD	47786	94.31	7127	96.65	6276	94.23
w/ SD	104539	**94.91**	7925	97.00	6690	94.45
w/ PD	77789	94.80	7930	97.15	6689	95.00
w/ GD	47786	94.83	7127	**97.30**	6276	**95.05**

Table 6: Results on general short-text classification. data represents the training data size. acc denotes the evaluation metric Accuracy (%). ↓ denotes the smaller the better, while ↑ means the larger the better.

that TGN effectively limits keyword prediction for non-duplicate texts by adding training difficulties and preventing trivial feature learning.

Finally, the results of one- and two-epoch training show that one-epoch outperforms two-epoch training. This is because generative models can replay keyword predictions for non-duplicate texts after multiple training epochs, resulting in the misidentification of non-duplicate texts.

5.4 General Short Text Classification

We have shown that our generative deduplication method effectively challenges social media language understanding tasks. To further demonstrate the generality of our approach, we also evaluate generative deduplication on three general short-text classification tasks, including AGNews (Zhang et al., 2015), Subj (Pang and Lee, 2004), and SST-2 (Socher et al., 2013). The results are presented in Table 6. The proposed generative deduplication can reduce the training data size, while keeping even improving the model performance slightly. Reduced training data size and improved performance demonstrate that the proposed generative deduplication effectively mitigates redundancy bias and can benefit general scenarios.

5.5 Further Discussions

Discussion of Self-supervised Task. In previous experiments, we have proven the effectiveness of the self-supervised keyword prediction task. Here, we examine other self-supervised tasks to provide further insights. The results are presented in Table 7. The table shows that predicting a text's first or last word significantly reduces the training size. However, they yield poorer performance than others, possibly because they remove too many non-duplicate patterns. In contrast, we can observe that the random word prediction task has minimal impact on reducing the training size because it is more challenging for the model to learn. Notably, single keyword (obtained by KeyBERT (Grootendorst, 2020)) and multiple keywords (obtained by ChatGPT (Kim et al., 2023)) prediction outperform the other self-supervised tasks. It is because keywords convey the main idea of a text and enhance semantic learning, thus improving the text understanding and deduplication performance. We choose the KeyBERT-extracted keyword for self-supervised learning by default since it is more efficient and cheaper than ChatGPT-generated keywords and achieves similar performance as ChatGPT.

Discussion of Redundancy Bias. To illustrate the redundancy bias intuitively, we present a plot of prediction confidence in Figure 4.

For the top 4 plots, we can see that the prediction confidence distribution of duplicate texts shifts towards a higher confidence zone than raw texts, suggesting a possible bias in the model. The bias can lead to incorrect predictions for input text. For example, in Figure 1, the model has incorrect prediction biased by the common "Pumpkin" features with duplicate texts. Results in Table 4 can also support this claim. The models trained on the generative deduplication data (less redundancy bias) outperform those without in these 4 tasks, except for LLaMA in Emoji prediction.

In contrast, the 3 bottom plots do not display notable distribution deviations, indicating that redundancy bias is not prominent in these tasks. In such cases, deduplication may negatively impact performance. This explains why the performance using generative deduplication data on BERTweet is lower than the raw training data for Humor and Stance detection tasks in Table 4.

Dicussion of Generative Deduplication Quality. Figure 5a shows the pairwise similarity distribution

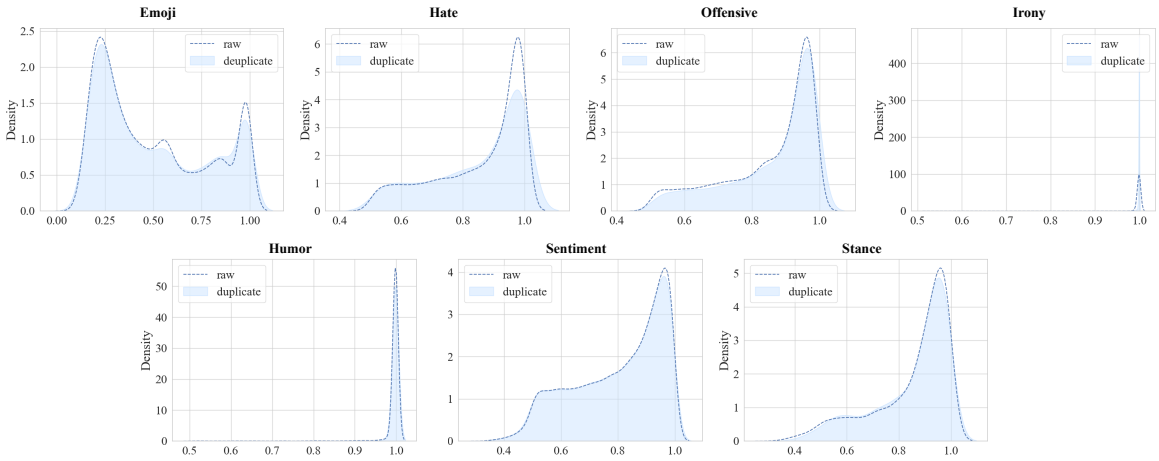
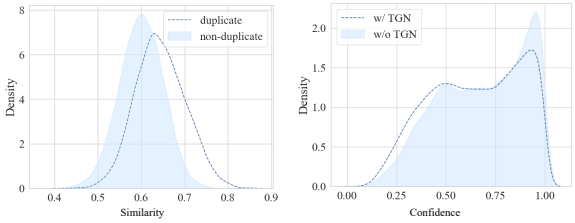


Figure 4: The Kernel Density Estimate (KDE) plot of the prediction confidence of RoBERTa-based models on the training set for each task. x-axis indicates the confidence. The top 4 plots present bias in the high confidence zone. The bottom 3 plots do not show obvious bias.

Model	Train Size	Macro F1 ↑
single keyword (KeyBERT)	$31.4K$	31.4
multiple keywords (ChatGPT)	$37.6K$	**31.8**
first word	$11.4K$	21.6
last word	$15.1K$	24.5
middle word	$26.6K$	28.9
random word	$44.9K$	30.6

Table 7: The results of different generative self-supervised tasks on the **Emoji** prediction task. K represents thousands. The train size of the raw **Emoji** is $45K$. T5-base is the generative backbone for generative deduplication, and RoBERTa-base serves as the downstream backbone model. Bold indicates the best results.



(a) (b)

Figure 5: (a) The similarity distribution for duplicate and non-duplicate texts, which are identified by generative deduplication, for all datasets. (b) The KDE plot of the generation probability (confidence) with and without TGN on **Emoji** generative deduplication.

for duplicate and non-duplicate texts (identified through generative deduplication). It is expected that duplicate texts would display higher similarity than non-duplicate texts. As observed, the similarity distribution of duplicate texts is shifted towards higher similarity values, indicating the good quality of generative deduplication. We also use LLM Claude Sonnet 3.5 to evaluate the quality of the generative deduplication. About 66% of the samples from the Irony dataset were considered semantically similar by Claude Sonnet 3.5. Table 8 shows 10 random samples of generative deduplication.

Effect of TGN. In Section 5.3, we have shown the effectiveness of the proposed TGN mechanism. Here, we further discuss it by comparing it to similar existing mechanisms.

First, TGN is analogous to the mask mechanism used in masked language models such as BERT (Devlin et al., 2019). The key difference is that we employ Gaussian noise instead of a fixed special mask token. TGN is more difficult than the mask mechanism. This design effectively limits the language understanding capabilities, preventing learning trivial features during generative training. As a result, it restricts keyword prediction for non-duplicate texts, thereby reducing misidentification.

Second, TGN also differs from the Dropout (Srivastava et al., 2014) mechanism. Dropout is commonly applied to the feature dimension, not the time dimension, and can potentially improve language understanding capabilities by addressing overfitting. Our experiment on the Emoji prediction demonstrates that replacing the proposed TGN with the Dropout mechanism leads to an inferior performance of 28.9 compared to TGN's 31.4.

Third, Figure 5b shows the TGN mechanism limits its language understanding capabilities, as shown the model without TGN has a higher generation confidence than with.

Text 1	Text 2	Label
Today is awesome	Today is awesome!	1
I have such a loving family	Dead supportive family I've got.	0
At least I woke up feeling a lot better today..	Yeah this is so good just woke up	1
I absolutely LOVE moving house	Dayum, I really got the house to myself while my brother still has school all week	0
Well, weekend is over!|Now it's #TwitterTime again :D|Have a nice monday!|	Nice weekend off but back at work tonight.	1
working on my birthday #yay #sucks	Love the fact I'm sick on my birthday	1
Be Blessed friends. Merry Christmas to all!	Merry Christmas @user	1
Great start to the day	Great way to start of the day	1
Love these cold winter mornings best feeling ever-rrrrrr!	I love cold winter days cause I never know when my car decides not to start	0
It's 8:46 and I'm ready for bed.	I am now heading for bed orz	1

Table 8: Ten random samples of generative deduplication from the Irony dataset. The label indicates the duplication judgment by Claude Sonnet 3.5. A label of 1 means that text 1 and text 2 are duplicates, 0 is non-duplicate.

6 Conclusion

In this paper, we have introduced a novel preprocessing method called generative deduplication for social media data selection. It tackles the semantic redundancy bias in noisy social media data. Extensive experiments have suggested that generative deduplication can significantly reduce the training cost of a model (in time and resources) while improving social media language understanding.

Ethics Statement

In our empirical study, we use publicly available social media understanding datasets that have been widely used in previous studies. These datasets typically do not have direct societal consequences. Our model introduces a novel paradigm, generative deduplication, for social media data selection. The proposed generative deduplication reduces the number of training samples, resulting in decreased computational resources, which is beneficial for the environment.

Limitations

We have initially tested the proposed generative deduplication method on widely used social media data, specifically TweetEval. In the future, we plan to extend our evaluation to additional social media datasets, such as Reddit TIFU (Kim et al., 2018) and GoEmotions (Demszky et al., 2020). Furthermore, generative deduplication is a general technique that can be applied to different contexts beyond social media. We will explore its applicability in broader scenarios.

Acknowledgements

This work is supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, the NSFC Young Scientists Fund (Project No. 62006203), China (Project No. PolyU/25200821), the Innovation and Technology Fund (Project No. PRP/047/22FX), PolyU Internal Fund from RC-DSAI (Project No. 1-CE1E), and PolyU Embodied Artificial Intelligence Lab (No. N-ZGNN).

Other than that, we sincerely thank the reviewers and ACs for their valuable input, which has greatly improved our work.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Francesco Barbieri, Jose 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*, pages 1644–1650, Online. Association for Computational Linguistics.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compres-*

sion and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171), pages 21–29. IEEE.

Cody Coleman, Christopher Yeh, et al. 2020. Selection via proxy: Efficient data selection for deep learning. In *Proc. of the ICLR 2020*.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yukun Feng, Patrick Xia, Benjamin Van Durme, and João Sedoc. 2022. Automatic document selection for efficient encoder pretraining. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9522–9530, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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*, pages 6894–6910. Association for Computational Linguistics.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in COVID-19 tweets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1596–1611, Online. Association for Computational Linguistics.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Hannaneh Hajishirzi, Wen-tau Yih, and Aleksander Kolcz. 2010. Adaptive near-duplicate detection via similarity learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 419–426.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. 2022. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jared Kaplan, Sam McCandlish, et al. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2018. Abstractive summarization of reddit posts with multi-level memory networks.

Yuheun Kim, Lu Guo, Bei Yu, and Yingya Li. 2023. Can ChatGPT understand causal language in science claims? In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 379–389, Toronto, Canada. Association for Computational Linguistics.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Xianming Li and Jing Li. 2024a. Aoe: Angle-optimized embeddings for semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839.

Xianming Li and Jing Li. 2024b. BeLLM: Backward dependency enhanced large language model for sentence embeddings. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 792–804, Mexico City, Mexico. Association for Computational Linguistics.

Miaofeng Liu, Yan Song, Hongbin Zou, and Tong Zhang. 2019a. Reinforced training data selection for domain adaptation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1957–1968, Florence, Italy. 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. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. 2007. Detecting near-duplicates for web crawling. In *Proceedings of the 16th international conference on World Wide Web*, pages 141–150.

J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

Sören Mindermann, Jan M Brauner, et al. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, pages 20596–20607.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proc. of the EMNLP 2019*, pages 3980–3990. Association for Computational Linguistics.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2017. Data selection strategies for multi-domain sentiment analysis. *arXiv preprint arXiv:1702.02426*.

Stephanie Schoch, Ritwick Mishra, and Yangfeng Ji. 2023. Data selection for fine-tuning large language models using transferred shapley values. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 266–275, Toronto, Canada. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, pages 1929–1958.

Hanzhuo Tan, Chunpu Xu, Jing Li, Yuqun Zhang, Zeyang Fang, Zeyu Chen, and Baohua Lai. 2023. Hicl: Hashtag-driven in-context learning for social media natural language understanding. *arXiv preprint arXiv:2308.09985*.

Ke Tao, Fabian Abel, Claudia Hauff, Geert-Jan Houben, and Ujwal Gadiraju. 2013. Groundhog day: near-duplicate detection on twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1273–1284.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2023. D4: Improving LLM pretraining via document de-duplication and diversification. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023. Data selection for language models via importance resampling. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xingcheng Yao, Yanan Zheng, Xiaocong Yang, and Zhilin Yang. 2022. Nlp from scratch without large-scale pretraining: A simple and efficient framework. In *International Conference on Machine Learning*, pages 25438–25451. PMLR.

Yue Yu, Rongzhi Zhang, Ran Xu, Jieyu Zhang, Jiaming Shen, and Chao Zhang. 2023. Cold-start data selection for better few-shot language model fine-tuning: A prompt-based uncertainty propagation approach. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2499–2521, Toronto, Canada. Association for Computational Linguistics.

Jichuan Zeng, Jing Li, Yan Song, Cuiyun Gao, Michael R. Lyu, and Irwin King. 2018. Topic memory networks for short text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3120–3131, Brussels, Belgium. Association for Computational Linguistics.

Xingshan Zeng, Jing Li, Lu Wang, Zhiming Mao, and Kam-Fai Wong. 2020. Dynamic online conversation recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3331–3341, Online. Association for Computational Linguistics.

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

Yuji Zhang, Jing Li, and Wenjie Li. 2023. VIBE: Topic-driven temporal adaptation for Twitter classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3340–3354, Singapore. Association for Computational Linguistics.