DiffusPoll: Conditional Text Diffusion Model for Poll Generation

Le Cheng
School of Computer Science
South China Normal University
Guangzhou, China
lecheng@m.scnu.edu.cn

Shuangyin Li [*]
School of Computer Science
South China Normal University
Guangzhou, China
shuangyinli@scnu.edu.cn

Abstract

Online social media platforms often gather user feedback through polls to enhance user engagement. Automatically generating polls from social media and its context can decrease the labor expenses of media workers and enhance workplace productivity. However, on social media platforms, there are internet water armies that manipulate public opinion through sheer numbers and causing the comments to be biased, drowning out minority views. In such circumstances, polls created based on biased comments often have limited types of options and poor coverage. Therefore, it is crucial to diversify the poll options and try to listen to the voices of the minority. To achieve this, we introduce DiffusPoll, a novel paradigm for poll generation based on a non-autoregressive diffusion model that can generate diversified and high-quality samples. Under the new paradigm, we design a task-specific mask strategy tailored to the inherent logic of polls to optimize controlled generation. Furthermore, we also leverage additional attribute tags from comments to enhance the generation quality. Experimental results indicate that DiffusPoll has achieved state-of-the-art performance in both the quality and diversity of poll generation tasks, and is more likely to hit the voices of minority.

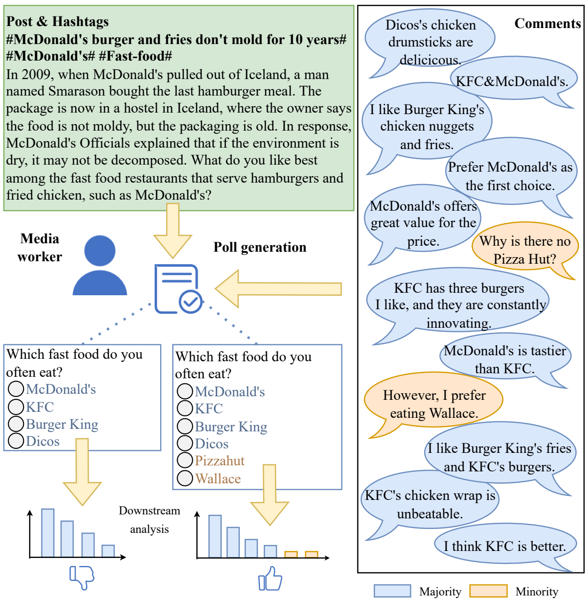

Post & Hashtags
#McDonald's burger and fries don't mold for 10 years#
#McDonald's# #Fast-food#
In 2009, when McDonald's pulled out of Iceland, a man named Smarason bought the last hamburger meal. The package is now in a hostel in Iceland, where the owner says the food is not moldy, but the packaging is old. In response, McDonald's Officials explained that if the environment is dry, it may not be decomposed. What do you like best among the fast food restaurants that serve hamburgers and fried chicken, such as McDonald's?
Media worker
Poll generation
Which fast food do you often eat?
○McDonald's
○KFC
○Burger King
○Dicos
Which fast food do you often eat?
○McDonald's
○KFC
○Burger King
○Dicos
○Pizzahut
○Wallace
Downstream analysis
Comments
Dicos's chicken drumsticks are delicicous.
KFC&McDonald's.
I like Burger King's chicken nuggets and fries.
Prefer McDonald's as the first choice.
McDonald's offers great value for the price.
Why is there no Pizza Hut?
KFC has three burgers I like, and they are constantly innovating.
McDonald's is tastier than KFC.
However, I prefer eating Wallace.
I like Burger King's fries and KFC's burgers.
KFC's chicken wrap is unbeatable.
I think KFC is better.
Majority
Minority

Figure 1: An example of poll generation, where McDonald's, KFC, Burger King, and Dicos with blue color options are more popular than the orange ones that Pizza Hut and Wallace are the voice of the minority. Thus, we need to mine the potential and minority comments to put them as poll options to enhance the poll's coverage.

1 Introduction

Social media allows us to hear the public voices for a better understanding our society and making decisions. There are two traditional methods to gather the public voices. The first way is to collect all the comments from each user, which would be labor-intensive and inefficient when the information increases. The second way is to only collect representative and valuable information, such as the "like" or the "reply". Compared to expressing opinions themselves directly, "voting" is an easy way for users just by clicking, which is the phenomenon of "silent majority" (Lu et al., 2021).

For that, social media polls are a suitable way to gather users' voices by setting pre-defined options for users to vote. After publishing the poll, the media worker could analyse users' voting result and perform user modeling to better. This approach can hugely increase salient user engagement so much that it is now adopted by many social media, such as Twitter and Sina Weibo. However, manually constructing polls can lead to low work efficiency. Thus, how to auto-generate high-quality polls is very interesting and significant, which is called the poll generation and can be shown as Figure 1. It takes a post and comments as input and aims to generate a poll containing a question and options.

Recently, some researchs attempted to employ

[*] Corresponding author.

autoregressive model Transformer (Vaswani et al., 2017) to generate polls (Lu et al., 2021; Li et al., 2023). These models can generate the text in a left-to-right fashion with the fixed context. However, when generating in this fashion, early errors could affect subsequent forecasts, leading to an accumulation of errors that cannot be modified. By adopting a fixed context, the model becomes more focused on local information dependencies. This is effective in enhancing the coverage of poll generation, but its performance in terms of diversity leaves much to be desired. While, in reality, we tend to fall into the dilemma of "invalid speech", because there are a large number of internet water armies who artificially boost comment counts and likes through sheer quantity, resulting in a bad phenomenon that media workers are drawn to the overwhelming majority of comments or posts while ignoring the voices of the minority. So these works could generate polls in an autoregressive manner, they also face the challenge of the internet water army and can not get rid of popularity.

Thus, there are still two main challenges in the poll generation that need to be fully considered: The first is the coverage on the majority. We need to mine the majority voices to improve the coverage of polls. The most intuitive approach is to mine the popular voices of the majority. Traditional methods with fixed context rely on mining large amounts of data to obtain barely acceptable results. The second is the diversity on the minority. For the example in Figure 1, Pizza Hut and Wallace are the voices of the minority. Minority groups often represent untapped market potential. Understanding and serving these groups can bring new growth opportunities for businesses. Thus, their discontent or complaint in the poll is what we care more about in real life. Existing methods make insufficient use of information to mine the diverse voices.

Therefore, how to leverage the coverage and diversity is very challenging in this task which can be shown in the poll options. To address these challenges, we propose DiffusPoll, a conditional text diffusion framework in a non-autoregressive manner for poll generation. First, we take advantage of the ability of the non-autoregressive diffusion model to capture long-distance contextual dependencies. This model is better to fit the inherent logic of a poll between question and the options. Moreover, the vanilla diffusion model uses a denoise network with self-attention mechanism, which is likely to result in generating the out-of-order samples. So, we design a task-specific mask strategy for this generation tasks to improve the performance on coverage. Second, we design an effective learning architecture to trade off the coverage and diversity on the poll options with a tailored diversity loss function. This architecture includes a well-designed control signal part to enhance diversity from large-scale comments. With that, our model can handle the diversity of opinions when generating the polls.

The main contributions of our work are:

• To the best of our knowledge, this paper is the first to propose the framework for poll generation with a diffusion model. This model can capture long-distance dependencies and global information for generating diverse and high-quality polls.

• An effective learning architecture is proposed to leverage the coverage and diversity with a tailored diversity loss function. The architecture can also control the generation process by the attribute tags to trade-off between coverage and diversity.

• A task-specific mask strategy is designed for poll generation to alleviate the problem that ordinary diffusion models produce out-of-order sentences. It is better to fit the inherent logic of a poll between the question and options with this model.

Finally, experimental results demonstrate the superior performance of DiffusPoll.

2 Related Work

2.1 Poll Generaion

Social media polls offer an easy way to hear the voices of the public and learn from their feelings on important social topics. (Lu et al., 2021) is the first to study poll questions on social media, where their interactions among answer options, source posts, and reader users' comments are first explored. Poll generation is similar to the extractive summary or comment mining task that takes a social media post (i.e., source) and outputs a poll question (i.e., target). For each question, possible answer options (i.e., answers) may also be yielded. To enrich the contexts of source posts, their reply messages (i.e., user comments) are also encoded as external features. Moreover, (Lu et al., 2021) collects a large-scale dataset from a Chinese social media platform named Sina Weibo, which containing over 20K polls and they proposed that treating poll generation as a sequence generation task for the first time. Based on previous research, (Li et al., 2023) adopted the pre-trained language model named T5-

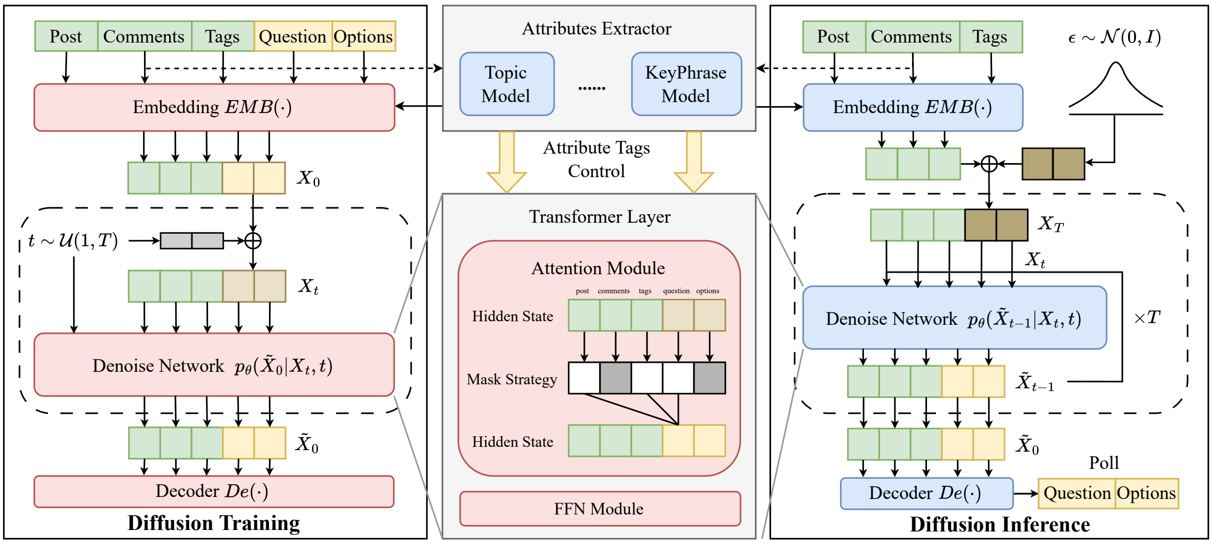


Figure 2: The overview of DiffusPoll. The left part is the training in that the model first extracts and maps the inputs to embeddings then conducts the forward process to corrupt them. Then, the model's denoise network is trained to recover the origin input embeddings. The right part is the inference where the input conditions come with a noisy vector from the Gaussian distribution and then perform the reverse process from T to 0. At last, once we have the restored vector, we can get the poll through a prediction head.

Pegasus (Su, 2021) by introducing three types of prompts for fine-tuning. This work decomposes the original task into three sub-tasks and jointly conducts multi-task learning by specific prompt labels. However, these works solely focused on how to generate polls and did not expand the coverage of polls based on the "silent majority" phenomenon. The most intuitive way to reflect the coverage is based on the number of options. The more options, the wider the coverage. Therefore, improving comment mining involves appropriately diversifying options to further enhance poll coverage.

2.2 Continuous Diffusion Model

Diffusion models are deep generative models utilizing Markov chains of diffusion steps to gradually recover the noise added to data (Sohl-Dickstein et al., 2015). The benefit of diffusion models lies in their ability to generate high-quality and varied data samples. Recently, diffusion models have shown impressive performance on continuous domains such as image and audio generation (Rombach et al., 2022; Kong et al., 2020). Some works on image generation with conditional diffusion explore classifier-guidance (Ho and Salimans, 2021) and classifier-free (Dhariwal and Nichol, 2021) by a classifier or setting guidance scale during training respectively. However, the desired output for these models isn't discrete textual data but consistent vectors representing pixel values. This is not suitable for text generation and needs more exploration.

Therefore, (Savinov et al., 2021) and (Yu et al., 2022) represent the initial two early attempts to model text connectivity through diffusion, employing either latent space or the encoder-decoder architecture. Inspired by these models, some works began to map text to embedding or latent variable and diffused on the embedding or latent space (Li et al., 2022; Kingma et al., 2021; Gong et al., 2023; Han et al., 2023; Dieleman et al., 2022). In addition, the continuous diffusion model is applied to various text generation tasks, such as Text style transfer (Lyu et al., 2023; Horvitz et al., 2023), Text detoxification (Floto et al., 2023), Empathetic response generation (Bi et al., 2023), Poetry generation (Hu et al., 2023), Extractive summarization (Zhang et al., 2023), Question generation (Yuan et al., 2022). They all use the diffusion model to generate a diverse range of sentences. Given the need for various options in poll generation, utilizing the diffusion model is feasible.

3 DiffusPoll

The DiffusPoll's framework is illustrated in Figure 2 and we will introduce our framework in three parts. First, we develop the **Attribute Tags Control** (Section 3.1), which aims to enhance the quality and coverage of polls by mining the potential options from the users' comments. Second, we design a task-specific **Mask Strategy** (Section 3.2) to address the issue of out-of-order sentences generated by vanilla diffusion language models. Third, we employ the **Diffusion Component** (Section 3.3) to generate polls by decoding from a rich semantic latent state and utilize our diversity loss to balance diversity and coverage.

In this paper, we treat the poll generation as sequence-to-sequence text generation tasks. Formally, given a m-length source post with contextual comments $W^U = \{\omega_1^U, ..., \omega_m^U\}$ and c-length attribute tags extracted from comments as conditional control $W^C = \{\omega_1^C, ..., \omega_c^C\}$, we aim to develop a text diffusion model that can generate a n-length poll $W^Y = \{\omega_1^Y, ..., \omega_n^Y\}$, conditioned on the input post, context, and attribute tags.

3.1 Attribute Tags Control

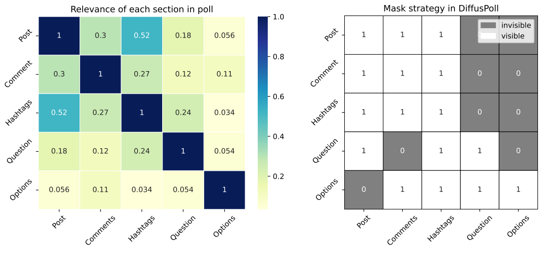
In order to strengthen the relevance of the input context and more effectively mine potential options that occur in the comments, we adopted three types of attributes: hashtags, topics, key-phrases to conduct the conditional generation progress in diffusion training and inference (Section 3.3).

On social media, posts often come with hashtags that start with "#", or are enclosed between "# ... #". These hashtags are typically added manually by the publishers, carrying a certain degree of subjectivity and summarization. Utilizing hashtags in poll generation was influenced by the observation that a significant portion of poll questions and options are closely associated with hashtags, which shown in Figure 3a. (Lu et al., 2021) released the WeiboPolls dataset that contains such hashtags.

Apart from using hashtags, we also referred to (Lu et al., 2021) work on ablation studies where topics were used as features. For over 20,000 documents, we conduct topic modeling and preset the nums of topics and represented each document with the topic words to enhance poll generation.

Unlike previous work, we also employed key-phrases as control attribute tags. We consider that hashtags are more subjectively representative of the original post text, while key-phrases objectively extracted from the contexts are more likely to become poll options. Therefore, we utilize the keyword extraction algorithm and use keywords with high confidence as attribute tags to guide the generation progress.

At last, we put these attribute tags together as a part of tags in Figure 3b and conduct the mask strategy during model training and inference.



(a) (b)

Figure 3: (a) Relevance of each section in the poll generation. (b) Task-specific Mask Strategy in DiffusPoll.

3.2 Mask Strategy

The vanilla diffusion model uses U-Net (Ronneberger et al., 2015) or Transformer (Vaswani et al., 2017) as the denoising network. In text generation tasks, previous work use the encoder Bert or encoder-decoder architecture Bart (Lewis et al., 2020). All of them utilize bidirectional self-attention encoding to extract features, enabling the model to perform cloze tasks for denoising. However, when the source and target are concatenated as the joint embedding for denoising, the model may fail in illogical sequences, resulting in out-of-order sentences during decoding. In particular, there is a logical order between a poll's question and its options and we prefer not to generate sentences randomly. Therefore, we need to modify the mask strategy in Transformer-based denoising network to avoid this issue. Inspired by the work (Bi et al., 2023) that facilitated generation with control-range masking, we design a task-specific mask strategy for poll generation.

To better design our mask, firstly, we calculated the relevance of each section measured by coverage and shown in Figure 3a. We evaluate the co-occurrence through the Rouge-1 metric. Areas with higher scores indicate a greater likelihood of co-occurrence. Secondly, in reality, we can tell the question is usually derived from the post and then users comment based on the post, the question, and their own knowledge. Therefore, there's a high probability that the options will appear in comments. Thirdly, we found that many attribute tags are related to the poll questions or are candidates for the poll options. Therefore, we take all these labels into consideration and not neglect them.

Considering mentioned above, we transform the bidirectional self-attention mask into a task-specific mask strategy, which is then fed into the network along with the vector that needs denoising for restoration. Further modeling the relationship between them with a mask matrix M and integrate it into the self-attention layer in Transformer:

$$
\begin{aligned}
&Q^i, K^i, V^i = h^{i-1}W_q, h^{i-1}W_k, h^{i-1}W_v \ , \\
&S^i = softmax(\frac{Q^i(K^i)^T + M}{\sqrt{d_k}}), \\
&h^i = S^iV^i,
\end{aligned}
\tag{1}
$$

where W_q, W_k, W_v are trainable parameters, h^i is the hidden state of the i-th Transformer layer, d_k is the dimension of the hidden state.

Our specific mask details can be referred to in the Figure 3b, where value 1 and 0 represents the visible relationship between each section. These two values will accordingly transform into 0 and negative infinity during attention computation for reducing computational cost. These logical control relationships can be reflected in the values of the mask matrix. Specifically, for a mask matrix, the value on position (i, j) is 0 if token j is controlled by token i; otherwise, it is negative infinity:

$$
M(i,j) = \begin{cases} 0, & i \Rightarrow j, \\ -\infty, & i \nRightarrow j. \end{cases}
\tag{2}
$$

Such approach ensures that the generated poll will not appear in an out-of-order sequence, and it also makes it easier to uncover potential options that might be present in the comments.

3.3 Diffusion Component

We will introduce our diffusion component in two parts: diffusion training with diversity loss and diffusion inference. During diffusion training, we utilize the training sample (W^U, W^C, W^Y) to optimize a diffusion model with diversity loss, enabling a balance between the poll's diversity and coverage. During inference, given the input context W^U and condition W^C , the trained model generates the final poll $\tilde{W}^Y$ by decoding from the latent space.

Diffusion training with diversity loss. Firstly, we employ the attributes extractor to get the topics or key-phrases as a part of the input for improving control generation. Then the post and comments as input W^U, attribute tags W^C and target poll W^Y are jointly embedded into the word embedding $EMB(W^U||W^C||W^Y)$, marked as E_W. Secondly, little noise from the first step $t = 0$ is added to become X_0 that is similar to $EMB(W^U||W^C||W^Y)$. So we get the distribution of origin X_0:

$$
\begin{aligned}
&E_W = EMB(W^U||W^C||W^Y), \\
&q(X_0|W) = \mathcal{N}(X_0; \sqrt{\alpha_0}E_W, (1-\alpha_0)I),
\end{aligned}
\tag{3}
$$

where the α_0 is a constant close to 1. Then, X_0 will be fed into the diffusion model to execute the forward process and the reverse process sequentially.

(a) Forward process in training. The diffusion model first samples t from the uniform distribution $\mathcal{U}(1, T)$ representing the noise level and calculates the standard variance β_t of the noise part. Secondly, we obtain the target poll part of X_0 by multiplying a binary mask named partial noising (Gong et al., 2023). Then, the target poll part is added to the noise according to the proportion mentioned in Equation 3 to obtain noisy embedding X_t. We have the distribution of noisy embedding X_t:

$$
q(X_t|X_0, t) = \mathcal{N}(X_t; \sqrt{\bar{\alpha}_t}X_0, (1-\bar{\alpha}_t)I).
\tag{4}
$$

(b) Reverse process in training. In this part, the diffusion model aims to recover the origin embedding $EMB(W^X||W^Y)$ from the noisy embedding X_t. We use the encoder with masked self-attention to get prediction $\tilde{X}_0$. In the attention module, we need to distinguish between tokens of different areas mentioned in Section 3.2. In order to train the denoise network f_θ, we minimize the variational lower bound following (Gong et al., 2023):

$$
\begin{aligned}
\mathcal{L}_{vlb} = R(\|X_0\|^2) + \sum_{t=2}^{T} \|Y_0 - \tilde{f}_\theta(X_t, t)\|^2 \\
+ \|EMB(W^Y) - \tilde{f}_\theta(X_1, 1)\|^2,
\end{aligned}
\tag{5}
$$

where Y_0 represents the parts of X_0 that belongs to W^Y, $\tilde{f}_\theta(X_t, t)$, denotes the fractions of recovered $\tilde{X}_0$ corresponding to Y_0, and the $R(\cdot)$ is a mathematically equivalent regularization term to regularize the embedding learning.

(c) Diversity loss. In addition to introducing high diversity through noise, we also design a loss function to measure diversity to enhance the variance of the generated output. Before training in the batch, we first calculate the options in the golden samples and obtain the count of options c_{gold} as the golden label. During training, the word embedding, when added to the noise at step $t = 0$, becomes X_0 in the forward process of the diffusion model. Then, by executing the forward sample at level t, we obtain the noisy vector X_t and feed it to the denoise network for getting the prediction $\tilde{X}_0$ of the input X_0. After rounding operation on this prediction $\tilde{X}_0$, we obtain the logits of prediction and softmax them to count the nums of options c_{pred} in denoising output. Thus, by calculating the count difference and normalizing it with the sigmoid function, we can minimize it as optimization objectives. Loss function can be represented as:

$$\mathcal{L}_{div} = 1/(1 + e^{-(c_{pred} - c_{gold})}) \; . \tag{6}$$

The goal of the diversity loss function is to let the model choose the direction gradient with the most options when denoising as much as possible. When using this loss, we choose a hyperparameter λ to control the performance and diversity.

$$\mathcal{L}_{total} = (1 - \lambda) \cdot \mathcal{L}_{vlb} + \lambda \cdot \mathcal{L}_{div} \; , \tag{7}$$

where $\mathcal{L}_{vlb}$ shown in the section in Equation 5.

Diffusion inference. Since the diffusion model itself is asymmetric in training and inference, the inference part only has the reverse process. Firstly, the input W^U and attribute tags W^C are jointly embedded to $EMB(W^U||W^C)$ as the conditions. Then, the model samples the pure noise from standard normal distribution $\mathcal{U}(0, I)$ and creates a noise vector W^Y. After concatenation of the conditions and noise vector, we get the noisy embedding X_T and feed it to the diffusion model. The denoise network will conduct the recover operation T times at noise level t in T to 0 order. At last, we put the last prediction sample $\tilde{Y}_0$, which denotes the fractions of $\tilde{X}_0$, and use the rounding operation to get the final poll $\tilde{W}^Y$.

4 Experiments

4.1 Dataset and Metrics

Dataset. Our proposed DiffusPoll model is trained and evaluated using the open source WeiboPolls[1] dataset, which is currently the benchmark dataset for poll generation. We use the same data split of (Lu et al., 2021) with 16,201 training, 2,025 validation, and 2,026 testing sentences for fair performance comparisons. WeiboPolls comprises 20,252 pairs of poll questions collected from the Chinese social media platform Sina Weibo. Each sample includes a post's context, hashtags, and associated comments, as well as a user-generated poll containing a question and a series of response options.

Metrics. Firstly, we follow the previous work (Lu et al., 2021) to evaluate the performance with Rouge-1 (R-1), Rouge-L (R-L), Bleu-1 (B-1) and Bleu-3 (B-3). As suggested by previous work, we use **Rouge-1** as the primary evaluation metric. Furthermore, we also incorporate the metric BertScore[2] (BS) with Bert-Chinese-base, which falls between Bleu and Rouge and assesses the semantic similarity between generated sentences and references. Secondly, the diversity of the generated polls will be assessed using distinct unigram (Dist-1), self-Bleu (Self-B) and different 4-grams (Div-4) metric. Dist-1 measures the internal diversity of each generated sentence, where a lower Dist-1 indicates that the generated sentence contains more repeated words. For sentence-level diversity assessment, self-Bleu is used to measure the n-gram overlap between the output set and a source sentence, along with the use of Div-4 to measure the proportion of unique 4-grams in the output set for each source sentence. Lower self-Bleu and higher Div-4 suggest greater diversity in generation. These implementations is based on NLTK[3] and Rouge Chinese[4]. Moreover, we also calculate the length (LEN) of the generated samples and the number (NUM) of options in the poll. NUM represents the number of options in the generated polls, is obtained by calculating the number of option slots. All these metrics are processed and calculated by a single Chinese character.

4.2 Baselines

We compare our methods with two groups:

(a) Transformer-based methods. The first two pre-trained language models (PLM) are to demonstrate the base performance without fine-tuning. (1) Bart-base (Lewis et al., 2020) and (2) GPT-2 (Radford et al., 2019) are the most common generative model. The subsequent two pre-trained models with poll generation task fine-tuning (PLM+FT) to demonstrate enhancement. (3) T5-Pegasus (Su, 2021) is a Chinese generative model based on the mT5 (Xue et al., 2021) and pre-trained in a manner similar to PEGASUS (Zhang et al., 2020) that generates pseudo summaries to fit the downstream fine-tuning tasks. We fine-tuned them on the WeiboPolls. (4) UniPoll (Li et al., 2023) is based on the T5-Pegasus and uses prompt learning for multi-task fine-tuning on the WeiboPolls dataset. By adding prompt labels for three sub-tasks to the training data to achieve data augmentation, there was an improvement over the pre-trained model in this way. (5) GPT-4 (OpenAI, 2023), we follow the work (Li et al., 2023), also compared the performance of the large language model (LLM) in the same scenario.

(b) Diffusion-based methods. (1) Diffuseq (Gong et al., 2023) is introduced as a conditional

[1] https://github.com/polyusmart/Poll-Question-Generation

[2] https://github.com/Tiiiger/bert_score

[3] https://www.nltk.org/_modules/nltk/translate/bleu_score.html

[4] https://pypi.org/project/rouge-chinese

Table 1: Poll generation results on WeiboPolls dataset. The best results are **bold**, while the secondary ones are marked with an underline.

Method		Performance					Diversity			Other	
		R-1↑	R-L↑	B-1↑	B-3↑	**BS** ↑	Dist-1↑	Self-B↓	Div-4↑	NUM↑	LEN
Golden		-	-	-	-	-	-	-	-	3.4205	36.835
PLM	Bart-base	0.3258	0.2457	0.1909	0.1425	0.6263	0.5521	0.7103	0.5004	1.8500	25.289
	GPT-2	0.2617	0.1718	0.1870	0.1085	0.5869	0.4224	0.7741	0.4512	1.5207	27.515
PLM+FT	T5-Pegasus	0.5346	0.4418	0.4285	0.3218	0.7230	0.7372	0.6012	0.5409	3.2230	29.089
	UniPoll	0.5357	**0.4487**	0.4363	**0.3279**	0.7310	0.7322	0.6317	0.5274	3.3253	30.067
LLM	GPT-4	0.3810	0.2908	0.2913	0.1304	0.6899	0.6929	0.5462	0.6300	4.6905	**52.037**
Diffusion	SeqDiffuSeq	0.4181	0.3472	0.3518	0.1960	0.6859	**0.7560**	0.6832	0.5453	3.4259	34.402
	DiffuSeq	0.4224	0.3330	0.3020	0.1524	0.6735	0.7383	0.6745	0.5974	2.9970	25.458
Ours	DiffusPoll$_{base}$	0.5061	0.4159	0.4281	0.2475	0.7282	0.6948	0.6104	0.6293	4.9294	37.464
	DiffuPoll	**0.5501**	0.4478	**0.4912**	0.3121	**0.7464**	0.7523	**0.5063**	**0.6844**	**5.1821**	37.922

diffusion language model for seq2seq tasks, utilizing partial noising, and it achieves a good balance between diversity and performance. This model uses the Bert as a denoise network. (2) SeqDiffuSeq (Yuan et al., 2022) employs the Bart encoder to extract features and the decoder for denoising, achieving results comparable to Diffuseq with less computing resources.

4.3 Implementation Details

DiffusPoll use Bert (Devlin et al., 2019) as the denoise network. For the WeiboPolls is not a huge corpus, we choose a medium-sized Bert With fewer parameters, about 120M, and also use the data augmentation method back-translation with translation API. For the poll generation, Bert-medium has 8 heads, a hidden size of 512, and 8 layers of stack blocks, which works better than other versions. For topic modeling analysis, we employ the LDA toolkit[5], it is simple and has fast inference speed. For key attribute tags, we utilize the TextRank algorithm to extract key-phrases. For diffusion model, we adopt the square-root noise schedule (Li et al., 2022) and set T=1,000 diffusion steps in the training and inference process. The maximum input length after concatenation is 256. The vocabulary of the model is more than 50,000 in T5-Pegasus and uses Jieba[6] as a pre-tokenizer. Drawing on the experience from previous work (Gong et al., 2023; Strudel et al., 2022), we choose the word embeddings to be in the size of 128 with random initialization.

For training settings, we use AdamW optimizer and set the learning rate as $1e^{-4}$. The batch size and dropout value are set as 512 and 0.1, respectively. For all experiments, we set 80,000 iterations and sample near $1e^{9}$ samples. We also adjust the micro-batch to 128 according to the specific device. The aforementioned work mentioned that the larger the micro-batch size, the better the performance. The λ in Equation 7 is set 0.01.

For all comparable methods, we use their official codes with the same settings or follow the original papers. For the decode strategy in baselines, we adopt the beam search strategy as described in their original code, and set the beam numbers as 4. In the pre-trained language models, Bart-base utilizes Bart-base-chinese[7], while GPT-2 utilizes GPT2-dialogbot-base-chinese[8]. Among them, due to the issue of close-source, the GPT-4 model adopts API[9] prompting to get polls.

For inference settings, we use batch size as 50 and set the MBR candidate S as 10. All experiments are deployed on NVIDIA A100 80G GPUs, and we use 2 GPUs for training and a single GPU for sampling.

4.4 Experimental Results

Performance comparisons. Poll generation results on the WeiboPolls are shown in Table 1. We have two key observations from the results. First, DiffusPoll outperforms others in the Rouge-1 and BertScore of these two main metrics, which means the model not only speaks fluently but also produces a good variety of samples. Second, We found that all diffusion models have salient advantages in terms of diversity and generated length, while the

[5] https://www.cs.columbia.edu/~blei/lda-c/

[6] https://github.com/fxsjy/jieba

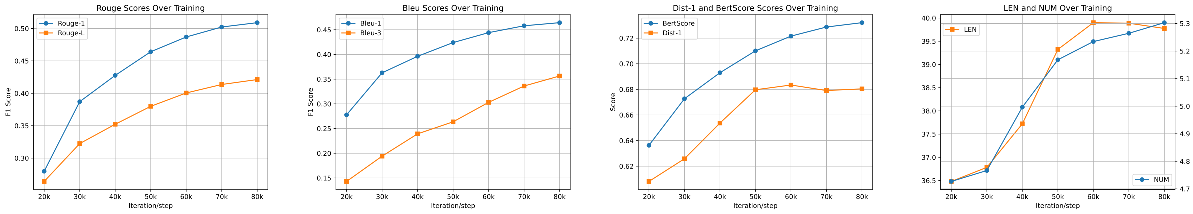
[7] https://huggingface.co/fnlp/bart-base-chinese

[8] https://huggingface.co/shibing624/gpt2-dialogbot-base-chinese

[9] https://openai.com/product

Table 2: The ablation experiments for various modules and enhancement rates of each module highlighted in *italics*.

Method	Performance					Diversity			Others	
	R-1↑	R-L↑	B-1↑	B-3↑	**BS**↑	Dist-1↑	Self-B↓	Div-4↑	NUM↑	LEN
Golden	–	–	–	–	–	–	–	–	3.4205	36.835
DiffusPoll$_{base}$	0.5061	0.4159	0.4281	0.2475	0.7282	0.6948	0.6104	0.6293	4.9294	37.464
+Mask	0.5411 *6.92%*↑	0.4416 *6.18%*↑	0.4933 *13.85%*↑	0.2958 *17.66%*↑	0.7455 *2.21%*↑	0.7489 *7.79%*↑	0.6002 *1.67%*↑	0.6554 *4.15%*↑	5.5913 *13.43%*↑	39.995 *6.76%*↑
+Div	0.5190 *2.47%*↑	0.4219 *1.44%*↑	0.4384 *1.18%*↑	0.2647 *5.29%*↑	0.7388 *1.29%*↑	0.7002 *0.78%*↑	0.5080 *16.77%*↑	0.6790 *7.90%*↑	5.3179 *7.88%*↑	36.734 *1.95%*↓
+Topic	0.5189 *2.54%*↑	0.4272 *2.72%*↑	0.4369 *0.83%*↑	0.2663 *5.93%*↑	0.7393 *1.36%*↑	0.6785 *2.35%*↓	0.5291 *13.30%*↑	0.6500 *3.29%*↑	5.1170 *12.62%*↑	34.469 *3.11%*↓
+Key	0.5281 *3.10%*↑	0.4242 *2.00%*↑	0.4416 *1.92%*↑	0.2460 *5.25%*↑	0.7398 *1.43%*↑	0.7056 *1.55%*↑	0.5059 *17.12%*↑	0.6820 *8.37%*↑	5.2611 *6.73%*↑	36.378 *2.90%*↓



(a) Rouge

(b) Bleu

(c) BertScore and Dist-1

(d) NUM and LEN

Figure 4: Different metrics over training.

autoregressive models perform relatively poorly. We attribute this to the fact that the diffusion model generates samples in a non-autoregressive way, allowing for closer semantic interactions between texts in more views. In contrast, autoregressive models use sequential training, and the sentences generated later can only see the results generated earlier. This manner is effective in generating fluent sentences, but would be too close to the training data, limiting the diversity of mining.

Besides, we make a quantitative comparison shown in Table 3 by sampling over 300 instances with 3-5 options and 10-20 comments each, totaling over 4,700 comments. We manually annotated over 1,000 potential minority comments and calculated the recall rate of the minority and majority labels hit by the options generated in the polls. At last, we give some metrics in different steps (Figure 4).

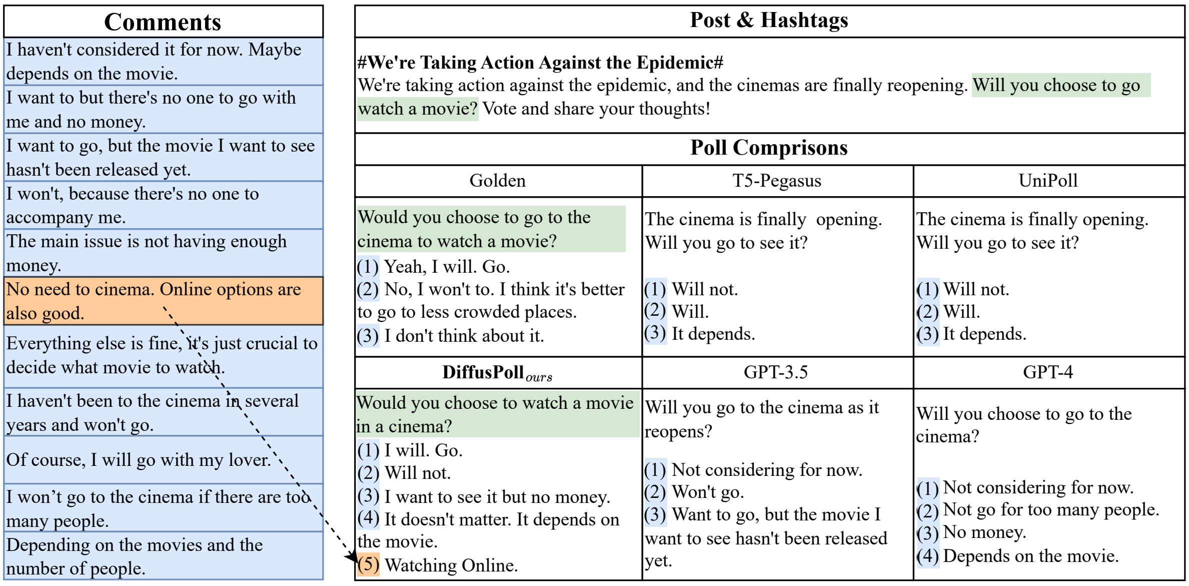
Ablation studies. We conduct a two-part ablations to investigate the effectiveness of the method proposed in the section. One verifies the role of modifying the module of model training, and the other is on the attribute extractor. The results of ablation experiments are shown in Table 2.

(a) Mask strategy and diversity loss ablations. Firstly, we utilized our mask strategy and compared it to the models without it. Among them, we can see that mask has the largest improvement both in diversity and performance. This demonstrates the effectiveness of our mask strategy. What's more, when using the mask strategy, the length and number of samples generated by the model increased significantly. Secondly, we test our diversity loss on the same settings. We can see that the impact of the diversity loss is relatively small, not as substantial as the improvement brought by the mask. Moreover, all metrics of performance except diversity decreased slightly. We attribute this to the diversity loss being particularly sensitive to rounding operations, which may use a more powerful decoder to match it better.

Table 3: Quantitative indicators measure the model's effectiveness in hitting both the minority and majority.

Method	**Minority**↑	Majority↑	Perplexity↓
T5-Pegasus	1.96%	54.41%	101.728
UniPoll	2.94%	58.60%	98.357
GPT-4	4.90%	44.33%	**68.770**
DiffusPoll	**6.85%**	**60.17%**	102.447

(b) Attribute tags ablations. To evaluate the impact of attribute tags, we choose topic and keyphrase as ablation experiments, which shown in Table 2. Under the comprehensive evaluations of diversity and performance, we found that the influence of the topic is greater than that of the keyphrases. It can be seen that by adding attribute



Comments
I haven't considered it for now. Maybe depends on the movie.
I want to but there's no one to go with me and no money.
I want to go, but the movie I want to see hasn't been released yet.
I won't, because there's no one to accompany me.
The main issue is not having enough money.
No need to cinema. Online options are also good.
Everything else is fine, it's just crucial to decide what movie to watch.
I haven't been to the cinema in several years and won't go.
Of course, I will go with my lover.
I won't go to the cinema if there are too many people.
Depending on the movies and the number of people.

Post & Hashtags
#We're Taking Action Against the Epidemic# We're taking action against the epidemic, and the cinemas are finally reopening. Will you choose to go watch a movie? Vote and share your thoughts!

Poll Comprisons		
Golden	T5-Pegasus	UniPoll
Would you choose to go to the cinema to watch a movie? (1) Yeah, I will. Go. (2) No, I won't to. I think it's better to go to less crowded places. (3) I don't think about it.	The cinema is finally opening. Will you go to see it? (1) Will not. (2) Will. (3) It depends.	The cinema is finally opening. Will you go to see it? (1) Will not. (2) Will. (3) It depends.
DiffusPoll$_{ours}$	GPT-3.5	GPT-4
Would you choose to watch a movie in a cinema? (1) I will. Go. (2) Will not. (3) I want to see it but no money. (4) It doesn't matter. It depends on the movie. (5) Watching Online.	Will you go to the cinema as it reopens? (1) Not considering for now. (2) Won't go. (3) Want to go, but the movie I want to see hasn't been released yet.	Will you choose to go to the cinema? (1) Not considering for now. (2) Not go for too many people. (3) No money. (4) Depends on the movie.

Figure 5: Case studies among polls generated from different models. The blue parts are popular comments, while the orange ones are from the minority. It shows that DiffusPoll uncovers the opinions of the minority. The green part represents the poll's loyalty to the input. Among them, the poll question generated by our model are closer to those topics in the post and shows more loyalty. Besides, we compared popular large models GPT who proposes the universal poll question is less loyalty to the post.

tags, the options number of samples generated by the model is increased, but the length generated is shorter, which indicates that the generated options may be more concise and representative.

5 Case Studies

Here we list a case where different models generated problems based on the same scenario shown in Figure 5. The green part represents the fidelity of the generated sample to the golden input, the orange part represents comments or options from the minority, and the blue represents the majority. Firstly, comparing the first two Transformer-based models, Unipoll and T5-Pegasus, we can find that the polls they generate are shorter and have fewer and more concise options. Even the results of Unipoll are the same as those of T5, indicating that the ability of Unipoll comes from the pre-training of T5 itself. Besides, we follow the work (Li et al., 2023), also compared the performance of the large language model on the same scenario. We choose the ChatGPT in GPT-3.5 and GPT-4 models to generate the poll by the prompt. The results clearly show that these large models generate more poll options, but the generated poll questions are pretty different from golden samples, which means the GPT will give the poll less loyalty and relevance. Looking carefully at these option candidate sets, we can find that the options generated by these GPT models are closer to the keyword from comments. We believe that this phenomenon comes from GPT's training data containing keyword extraction and summary tasks. However, when the comment section is occupied by the Internet water army, and most comments become dominant, the option of GPT to generate a poll runs the risk of being controlled by the Internet water army. Moreover, we found that our DiffusPoll generates poll not only effectively maintained loyalty but also made breakthroughs in the diversity of options. An interesting phenomenon is that our model also learned the option "Others...", which shows that the model's generated poll generation is scalable and more suitable for real scenarios.

6 Conclusion

In this paper, we use the diffusion model to tackle the poll generation for generating high-quality and diverse samples. We introduce the DiffusPoll, a conditional diffusion model, where the conditional parts use the task-specific mask strategy and attribute tags to improve performance and diversity. Experimental results show that DiffusPoll matches the performance of the Transformer model while offering greater diversity, which makes it promising for application in real social media to enhance user engagement and mine voices of the minority.

Limitations

We focus on developing automated methods to generate high-quality social media polls. The opportunity to embed these polls within our social platform's business operations and utilize them for downstream tasks, like feedback loops, deserves additional investigation. Furthermore, evaluating this approach poses a significant challenge. Currently, we employ automated assessments with quantitative analysis. In fact, for successful implementation and deployment, the incorporation of human annotations and real user feedback is indispensable.

Acknowledgements

This work was supported by Natural Science Foundation of Guangdong (2023A1515012073) and National Natural Science Foundation of China (No. 62006083).

References

Guanqun Bi, Lei Shen, Yanan Cao, Meng Chen, Yuqiang Xie, Zheng Lin, and Xiaodong He. 2023. DiffusEmp: A diffusion model-based framework with multi-grained control for empathetic response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2812–2831, Toronto, Canada. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. 2022. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*.

Griffin Floto, Mohammad Mahdi Abdollah Pour, Parsa Farinneya, Zhenwei Tang, Ali Pesaranghader, Manasa Bharadwaj, and Scott Sanner. 2023. DiffuDetox: A mixed diffusion model for text detoxification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7566–7574, Toronto, Canada. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations, ICLR*.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. 2023. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11575–11596, Toronto, Canada. Association for Computational Linguistics.

Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Zachary Horvitz, Ajay Patel, Chris Callison-Burch, Zhou Yu, and Kathleen McKeown. 2023. Paraguide: Guided diffusion paraphrasers for plug-and-play textual style transfer.

Zhiyuan Hu, Chumin Liu, Yue Feng, and Bryan Hooi. 2023. Poetrydiffusion: Towards joint semantic and metrical manipulation in poetry generation. *arXiv preprint arXiv:2306.08456*.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*.

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

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343.

Yixia Li, Rong Xiang, Yanlin Song, and Jing Li. 2023. Unipoll: A unified social media poll generation framework via multi-objective optimization. *arXiv preprint arXiv:2306.06851*.

Zexin Lu, Keyang Ding, Yuji Zhang, Jing Li, Baolin Peng, and Lemao Liu. 2021. Engage the public: Poll question generation for social media posts. 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 29–40, Online. Association for Computational Linguistics.

Yiwei Lyu, Tiange Luo, Jiacheng Shi, Todd Hollon, and Honglak Lee. 2023. Fine-grained text style transfer with diffusion-based language models. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 65–74, Toronto, Canada. Association for Computational Linguistics.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. 2021. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. 2022. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*.

Jianlin Su. 2021. T5 pegasus - zhuiyiai. Technical report.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Peiyu Yu, Sirui Xie, Xiaojian Ma, Baoxiong Jia, Bo Pang, Ruiqi Gao, Yixin Zhu, Song-Chun Zhu, and Ying Nian Wu. 2022. Latent diffusion energy-based model for interpretable text modeling. In *Proceedings of International Conference on Machine Learning (ICML)*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. DiffuSum: Generation enhanced extractive summarization with diffusion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13089–13100, Toronto, Canada. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.