

# Maximizing ChatGPT's Data Augmentation Capabilities: Exploring Optimal Prompts

**Enze Ma**                      **Fangqi Yuan**                      **Xiaolong Huang**                      **Xuan Ding**  
e1ma@ucsd.edu      fayuan@ucsd.edu      xih002@ucsd.edu      xding@ucsd.edu

## 1 What you proposed vs. what you accomplished

- ~~Acquire and pre-process data in our dataset:~~ DONE.
- ~~Build model for the task:~~ DONE.
- ~~Analyze the output of the model and choose the prompt:~~ DONE.
- ~~Work on final reports:~~ DONE.

## 2 Related work

We find some related works analyzing data augmentation's capabilities from different models.

In the article "AugGPT: Leveraging ChatGPT for Text Data Augmentation" (Dai et al., 2023), authors introduce AugGPT, a text data augmentation approach based on ChatGPT and BERT. AugGPT rephrases each sentence in the training samples into multiple conceptually similar but semantically different samples. And this approach outperforms state-of-the-art text data augmentation methods in terms of testing accuracy and distribution of augmented samples. This enlightens us to use the same model, which is to prompt ChatGPT to generate samples that preserve semantic consistency with given labelled data set, and train a BERT-based sentence classifier on the generated data samples to evaluate the model's classification performance.

In the article "Generalizing from a Few Examples: A Survey on Few-Shot Learning" (WANG et al., 2020), authors introduce Few-Shot Learning as a solution to the challenge of performing machine learning with small data sets. It allows for rapid generalization to new tasks with only a few labeled samples by utilizing prior knowledge. It helps us to deal with the small data set training because ChatGPT has the limitation of

generating samples from large data sets. Therefore, we will give ChatGPT a small data set and use Few-Shot Learning to perform machine learning.

In the article "Meta-learning for Few-shot Natural Language Processing: A Survey" (Yin, 2020), authors introduce Meta-learning in Few-Shot NLP, which can train a model on a variety of tasks with rich annotations so that it can solve new tasks using only a few labeled samples. It inspires us to use the "Emotions in Text" data set which is a small data set with 8167 rich labels. The key idea is that we can apply meta-learning on the "Emotions in Text" data set and train the model's initial parameters enough to enable it to quickly adapt to new tasks.

In the article "TransPrompt: Towards an Automatic Transferable Prompting Framework for Few-shot Text Classification" (Wang et al., 2021), authors introduce TransPrompt, a transferable prompting framework for few-shot learning across similar tasks. And they propose De-biasing techniques in TransPrompt to enhance task-agnosticism and reduce bias toward tasks. We may use TransPrompt framework to improve performance compared with other baseline algorithms.

In the article "A Survey of Data Augmentation Approaches for NLP" (Feng et al., 2021), authors introduce the challenges of data augmentation in NLP, such as low-resource domains, new tasks, and the use of large-scale neural networks that require substantial training data. Authors also provide a structured survey of data augmentation for NLP by summarizing the existing literature, which clearly describes the need for further work in this area to address the identified challenges and explore new possibilities. This inspires us to explore the different data augmentation approaches and try to use the model-based approaches men-

tioned in the article such as GPT-2 generation method, syntactic paraphrasing, and BERT cross-encoder to do the research.

### 3 Our dataset

For the datasets we used in our project, the first one is the "go emotions" dataset (Demszky et al., 2020) which is available on "HuggingFace". The GoEmotions dataset is a collection of carefully curated Reddit comments that have been labeled for 28 emotion categories. This dataset includes pre-defined 43,410 train, 5426 validation, and 5427 test splits.

Basic Statistics for GoEmotions Dataset	
Size	47.1MB
Total number of words	2,745,331
Average number of words	13
Total number of sentences	211,225
Total number of documents	1

Table 1: GoEmotions dataset's basic statistics

There are 37 columns in "GoEmotions" dataset which is shown in the following screenshot by showing some examples of GoEmotions dataset:

	text	id	author	subreddit	link_id	parent_id	created_utc	rater_id	example_very_unclear	admiration	...
0	That game hurt.	eeed50	Brody	nfl	13_gj4z	11_ew18eq	1.548381e+09	1	False	0	...
1	...sexuality shouldn't be a grouping category...	eeemcyk	TheGreen888	unpopularopinion	13_u4q37	13_u4q37	1.548894e+09	37	False	0	...
2	You are right, if you don't care then fuck well!	ed2mah1	Lababoi	confessions	13_u8u74	11_ed2m7g7	1.546428e+09	37	False	0	...
3	Man I love reddit.	eeab0d3	MuHobenthaw	facepalm	13_uh4ml	13_uh4ml	1.547895e+09	18	False	0	...
4	[NAME] was realtate near them, he was by the Pa...	ed4dlyr6	American_Fascist713	starewarspeculation	13_eed02	11_eed05q2	1.546669e+09	2	False	0	...

The second dataset we used is the "Poem sentiment" dataset (Sheng and Uthus, 2020). This dataset is a collection of poems that have been labeled into 4 categories: negative(0), positive(1), no\_impact(2), and mixed(3). This dataset is split into 892 train, 105 validation, and 104 test sets.

Basic Statistics for Poem Sentiment Dataset	
Size	49.9KB
Total number of words	6,329
Average number of words	7
Total number of sentences	892
Total number of documents	1

Table 2: Poem Sentiment dataset's basic statistics

The third dataset we used is the "sentiment-analysis-tweet" dataset (LYTinn, 2020). The tweet sentiment dataset is a collection of tweets that have

Example data in Poem Sentiment Dataset	
text	label
"and that is why, the lonesome day,"	0
"and very venus of a pipe."	2
"that has a charmingly bourbon air."	1
"the red sword sealed their vows!"	3

Table 3: Poem Sentiment dataset's example

been labeled into 3 categories: negative(0), positive(1), neutral(-1). The dataset has been transformed with data cleaning, random ordering, and keeping the data balance between each category. It has been split into 10000 train and 4910 test sets.

Basic Statistics for Tweet Sentiment Dataset	
Size	1.61MB
Total number of words	192,294
Average number of words	19
Total number of sentences	10,000
Total number of documents	1

Table 4: Tweet Sentiment dataset's basic statistics

Example data in Tweet Sentiment Dataset	
text	label
"CF taking some off here close to 19150"	1
"winking face"	0
"has the worst tvs"	-1

Table 5: Tweet Sentiment dataset's example

The forth dataset we used is "amazon\_polarity" dataset (McAuley et al., 2013). The dataset is a collection of reviews obtained from Amazon, covering a wide range of products. The dataset includes approximately 35 million reviews that have been labeled into 2 categories: negative and positive. Each review includes information about the product, user details, ratings, and a plaintext review. It has been split into 1,800,000 training and 200,000 testing samples.

These datasets provide labeled examples where labels are explicitly associated with textual content, making them an ideal resource for training and evaluating sentiment analysis models. Besides, data augmentation is employed to increase the diversity and quantity of training data, which

Basic Statistics for Amazon Polarity Dataset	
Size	1.61MB
Total number of words	15529934
Average number of words	4
Total number of sentences	3,600,000
Total number of documents	1

Table 6: Amazon Polarity dataset’s basic statistics

Example data in Amazon Polarity Dataset	
text	label
”Stuning even for the non-gamer”	1
”Buyer beware”	0
”A FIVE STAR BOOK”	1

Table 7: Amazon Polarity dataset’s example data

helps improve the generalization and robustness of machine learning models. By using these datasets as a base, we are training ChatGPT by giving good prompts with raw text data, expecting to receive high-quality responses from ChatGPT, in order to perform data augmentation to better capture the data meaning and increase the data variance. With more training samples, we can increase the accuracy of our BERT model by applying the text data augmentation approach based on ChatGPT. The following are some examples of input/output pairs. The input is the prompt that we gave to ChatGPT, and the output is the response generated by ChatGPT that preserves semantic consistency with existing labeled instances. The challenge we meet is that in the ”Poem Sentiment” dataset, the text property has many confusing words that cause the ChatGPT model cannot generate accurate and diverse text data or cannot capture the main emotions of the poems. Hence, it is hard to train ”Poem Sentiment” data in BERT model to get better accuracy in the testing set.

### 3.1 Data preprocessing

In each dataset, we checked if there are missing data, such as the texts with no labels, or labels with no text. Specifically, in the ”GoEmotions” dataset, we checked if there exist texts that have multiple emotion labels. In addition, to improve the performance of our text classification BERT model, we relabeled the 28 emotion labels to the 3 categories: positive, negative, and neutral. Besides, due to the ChatGPT word’s limitation, we shrink the huge datasets ”GoEmotions” and ”Amazon Polarity” to

Examples of Input/Output pairs	
Input	Output
”Please rephrase this sentence while retaining the sadness emotion: That game hurt.”	”I was hurt by that game.”
”Please rephrase the sentence: Man I love reddit.”	”I have a strong fondness for Reddit.”
”Please rephrase the sentence from Reddit but keep their emotion: that’s adorable asf”	”That is extremely cute and endearing!”
”Please rephrase the sentence to around 64 words without changing emotion: BUT IT’S HER TURN!”	”However, it is now her opportunity.”

Table 8: Input/output pairs Example

smaller ones to prevent memory overflow.

### 3.2 Data annotation

In our project, we utilized the Emotion dataset as one of our primary datasets, which initially encompassed 28 unique emotion labels assigned to each text entry. However, recognizing the inherent complexity of this dataset and the challenges it posed for effective model training, we made the decision to simplify it. To achieve this, we engaged in a manual re-annotation process, wherein we meticulously identified and labeled the specific emotion conveyed within each text input. Ultimately, we condensed the diverse range of emotions into three primary categories, namely Positive (1), Negative (0), and Neutral (2). The sample text and label are shown in the table 9. This refinement allowed us to streamline the dataset and focus on these core emotions for our model training and subsequent analysis.

Sample Text	Label
Shut up 15 year-old	0
Pretty much every Punjabi dude I’ve met.	1
this definitely fits in r/BoneAppleTea.	2

Table 9: Sample Text and Labels from Emotion dataset

For Twitter Sentiment Dataset, we adjust the label 1, 0, -1 from 2, 0, 1, namely Positive (2), Negative (0), and Neutral (1)

For the remaining datasets utilized in our

project, the labels were found to be adequately structured and appropriately distinct, which means making further re-annotation will be unnecessary. These datasets had been constructed with clarity that allowed for direct utilization of our model, thereby simplifying the pre-processing stages of our project.

## 4 Baselines

In our baseline approach, we initially train a BERT sequence classification model using a dataset consisting of 700 original data samples and 700 synthetic data samples generated by ChatGPT. The prompt used for generating the synthetic data is "Please rephrase the following sentence: 'from paper AugGPT(Dai et al., 2023)'"'. To evaluate the performance of our model, we set aside 400 isolated data samples as our test data.

We continue training the model until it starts to overfit, which indicates that it has reached its maximum performance under the given conditions. At this point, we assess the highest achievable accuracy of the model, which serves as our baseline. In order to ensure the attainment of accurate results, it is imperative to conduct multiple iterations of the experiment then use the average, rather than relying solely on a single run. We consider this accuracy as the limit of the model's performance in this scenario.

To determine the optimal hyperparameters, we conduct a preliminary trial using 1,400 training data samples. From this set, we randomly select 400 samples that are separate from both the training and test data to form our validation dataset. We vary the hyperparameters and select the settings that yield the highest accuracy on the validation data. Once these optimal hyperparameters are identified, we use them consistently for all subsequent experiments without further modification. training/validation/test will be 1400/400/300.

The baseline accuracy is listed on the table 11 for each dataset we used.

## 5 Your approach

- (CONCEPTUAL APPROACH)

We utilize ChatGPT along with various prompts, including the one introduced by AugGPT(Dai et al., 2023), to enhance our training process. Our objective is to generate synthetic data by using our initial training data to create new examples. To achieve

this, we leverage ChatGPT's capabilities to generate new text samples based on the given prompts. For synthetic examples by different prompts, they will be merged with a copy of the original dataset as one possible augmented training set. Next, we employ these augmented datasets to train a Bert sequence classification model for classifying emotions in our test set. The purpose will be to identify the best prompt among the ones we have used.

- (WORKING IMPLEMENTATION)

Yes. We will use the same thought from AugGPT(Dai et al., 2023) and try to implement a similar structure and adjust the prompt during data argumentation by ChatGPT.

- (OTHER PEOPLE'S CODE)

Yes, we rely on the Bert Classification model from Hugging Face, which is a pre-trained model provided by the Hugging Face. It is trained on a large corpus of raw English text with no human annotations. The model learns an inner representation of the language that can extract features and then be used for downstream tasks. Because we are not trying to find the best model, we are actually trying to find the best prompt for sentiment analysis data augmentation. Therefore, we just simply use this model and fine-tune it to its maximum performance to check the data augmentation quality by ChatGPT. Moreover, we use OpenAI's gpt-3.5-turbo model, which is a language model developed by OpenAI. It is an advanced version of the GPT that offers improved performance in terms of language understanding and generation and is specifically designed to generate coherent and contextually relevant text based on given prompts. And it has been proved that it is an effective tool for text data augmentations (Dai et al., 2023). Therefore, we will utilize the get-3.5-turbo model for data augmentation.

- (IMPLEMENTATION)

We implemented a model based on AugGPT(Dai et al., 2023) which consists of the Bert classification model and GPT-3.5-turbo. Our implemented model basically follows the

Prompt	Prompt
Prompt 1(Baseline)	Please rephrase the following sentence +text
Prompt 2	Please rephrase the following sentences from +source+ but keep their emotion +text
Prompt 3	Please rephrase the following sentences to around +max_length words without changing emotion +text
Prompt 4	Please rephrase this sentence while retaining the "+label+" emotion +text
Prompt 5	Please translate this sentence to Spanish while retaining the emotion +text+. Please translate this sentence to English:+answer

Table 10: Prompt v.s the text for chatGPT

Prompt	Emotion	Twitter	Poem	Amazon
Prompt 1(Baseline)	0.470	0.580	0.760	0.796
Prompt 2	0.486	0.575	0.761	0.793
Prompt 3	0.488	0.572	0.756	0.773
Prompt 4	<b>0.502</b>	0.613	<b>0.779</b>	0.803
Prompt 5	0.496	<b>0.624</b>	0.766	<b>0.823</b>

Table 11: The Accuracy

theory of AugGPT but we aim to use different prompts other than the original model. For each experiment, we will give GPT-3.5-turbo a different prompt to rephrase each text sentence in the training sets to the conceptual similar one. Then we use the generated samples and the original data for downstream model training. The testing accuracy results given by our models will show the performance for different prompts.

The files are listed as follows:

- hyper.select.ipynb: used to try different hyperparameter setting
- poem.ipynb: run different prompts on Poem dataset
- emotion.ipynb: run different prompts on Emotion Sentiment dataset
- twitter.ipynb: run different prompts on Twitter Sentiment dataset
- amazon.ipynb: run different prompts on Amazon Sentiment dataset

The google drive link is: [Data and Code](#)

#### • (COMPUTE)

We have four different types of computers. These include a Macbook Pro 14, a desktop with an Nvidia 3080ti and AMD 5800x, another desktop with an Nvidia ASUS TUF

3080ti and Intel 12700k, and another desktop with an Nvidia 3080 and AMD 5900x.

It's crucial to note that the Mac does not support "cuda", which results in a significantly longer runtime compared to the other computers when training the dataset and running the model for the different prompts.

#### • (RUNTIME)

For the Emotion sentiment dataset, it will spend almost 20min to get all the augmented text from GPT-3.5-turbo and added back to the original dataset, and takes almost 5min to train a Bert model.

For the Poem sentiment dataset, it will spend almost 5min to get all the augmented text from GPT-3.5-turbo and added back to the original dataset, and takes almost 3min to train a Bert model.

For the Twitter sentiment dataset, it will spend almost 33min to get all the augmented text from GPT-3.5-turbo and added back to the original dataset, and takes almost 10min to train a Bert model

For the Amazon sentiment dataset, it will spend almost 65min to get all the augmented text from GPT-3.5-turbo and added back to the original dataset, and takes almost 20min to train a Bert model.

#### • (RESULTS)



Upon examining the results table above table 10 and table 11, we observe a notable performance improvement. There is approximately a 3 percent enhancement in accuracy for the final two prompts when compared to our baseline model.

Prompt 2 is designed to preserve the emotion of the original text, and imitate the text style in the source text. This approach seeks to maintain the essence of the input text by closely mirroring the speaker's unique style. However, the effectiveness of this prompt may vary depending on how distinctive or explicit the speaker's tone is in the original text. Clear and expressive tones may translate well, while more subtle or ambiguous tones might pose a challenge.

The aim of Prompt 3 is to extend the length of the input sentence, thereby providing a richer context that preserves the original emotional tone of the text. However, the accuracy of this approach proved to be inconsistent. In some instances, it yielded high accuracy, while at other times, the results were less than satisfactory. The variability in performance suggests that although an additional context can enhance emotional clarity, it doesn't always guarantee improved accuracy in emotion detection.

For prompts 2 and 3, there are instances where they can yield significant enhancements, but most of the time they cause lower accuracy. It appears that these prompts are not reliable. So it may not be a good idea to use such prompts.

The increased accuracy observed for Prompt 4 is attributable to the inclusion of the input text's corresponding label within the data fed into ChatGPT. We identified during our initial investigations that ChatGPT, on certain occasions, could misinterpret the emotional tone of the input text. This misinterpretation would consequently result in incorrect emotional rephrasing of the output text, thereby reducing accuracy.

To address this issue, we incorporated the emotion label associated with the input text into the information given to ChatGPT. This adjustment aimed to foster a more accurate understanding of the emotional context by

ChatGPT, subsequently allowing for more accurate emotional rephrasing.

With this modification, the rephrased output text produced by ChatGPT demonstrated higher congruence with the model's expectations, thereby leading to improved accuracy. This enhancement attests to the effectiveness of providing explicit emotional context, as reflected in the emotion label, to guide the model's rephrasing process.

The performance of Prompt 5 showed a noticeable improvement in model accuracy as well. In Prompt 5, we employed a 'translate-back' mechanism using ChatGPT. Considering language in our input, we instructed ChatGPT to convert the input text into Spanish because Spanish and English are closely related. So we can preserve most of the information and emotional tone. But the slight difference other than the traditional translation back technique we use the advantage of ChatGPT, during the first translation we emphasize the importance of reserving the emotional information. Subsequently, we translated the Spanish text back into English to derive the final output.

Nevertheless, ChatGPT sometimes failed to accurately translate certain input phrases, which were complex or lacked clear meaning in English. For instance, the phrase "fu' to ba' de battle's brunt" posed a challenge for ChatGPT, resulting in the response, "I'm sorry, as an AI language model, I cannot detect the emotion in the phrase ""fu' to ba' de battle's brunt"" as it doesn't have a clear meaning in English. Therefore, I cannot provide a translation while retaining the emotion. Please provide more context or a complete sentence so that I can assist you better." Inclusion of such uninterpretable output text in our dataset inevitably brought down the performance accuracy of our model to some extent. However, despite this limitation, the performance accuracy of Prompt 5 still surpassed that of our baseline model, which underscores the effectiveness of the 'translate-back' mechanism in enhancing data augmentation quality overall.

## 6 Error analysis

In the baseline, for poem dataset, the reason why it predicts wrong is that the poem is hard to understand and it may use a lot of dialect and metaphor which is hard to determine whether its emotional positive or negative. For example, "fu' to ba' de battle's brunt." or "but she spake: "bid hither gunnar, lest the sun sink o'er the bent,"". When we put a poem data into GPT-3.5-turbo, GPT will return "I'm sorry, I can't make a translation as the sentence presented is written in a dialect of English that I can't understand. Please provide a sentence written in standard English so I can help you translate it into Spanish." or keep the original sentence. GPT always failed to detect the emotion in the poem phrase or misunderstands the text because it is hard to recognize a syllable in a dialect. Therefore, ChatGPT failed to meet the basic requirement of predicting a poem's emotion, which involves employing language patterns to condense the chaotic data of personal experiences, emotions, truths, or knowledge, and transforming them into a structured and meaningful expression. Hence, we need to focus on finding the datasets that contain straight text with explicit labels, to make sure that ChatGPT can capture the meaning and emotions of text data.

In the baseline for Emotion Sentiment dataset, there exist many wrong human annotations in the dataset. For example, in "GoEmotions" dataset, the text sentence: "That's the hardest part for some to get over, but hobbies help and getting yourself up and out of the house and spending time with friends" is labeled as negative(0). But we know that the true label should be positive(1). Moreover, for the correct labeled data, ChatGPT may rephrase the neutral sentences or sentences that convey uncertainty or a possibility including "maybe, perhaps" to the negative or positive sample because ChatGPT prefers capturing the emotional words in each sentence rather than feeling or thinking a sentence as a human.

## 7 Contributions of group members

- Enze Ma: found some datasets, related papers, preprocessing data, implementing code, and doing lots of writing.
- Fangqi Yuan: implemented code to build and train models, fine-tuning the model, and run some of the prompts, and do lots of writing.

- Xuan Ding: Come up with good prompts, run models with different prompts, and do lots of writing.
- Xiaolong Huang: Conclude all data into a report and analysis to compare with the baseline model, and do lots of writing.

## 8 Conclusion

In our project, we find prompt with certain sentiment assist or back translation with emotion to GPT-3.5-turbo can help us improve our synthetic data quality and improve the performance of the BERT classification model. Based on different prompts that we have given to ChatGPT, as shown in table 11, the test accuracies will increase 3% on average when using prompt 4 or prompt 5 in some cases compared to the baseline prompt.

In summary, our experiment involving five distinct prompts has yielded valuable insights about the behavior and performance of our model.

Prompt 4 and prompt 5 involved the inclusion of the text's emotion label and the application of a 'translate-back' mechanism, which showed a significant improvement in test accuracy. These strategies effectively guided ChatGPT in correctly interpreting and rephrasing the text. Prompt 3 aimed at enriching the context by extending the sentence length, demonstrated mixed results. While in some cases it improved accuracy, in others it did not, indicating that additional context does not always lead to increased accuracy. Prompt 2 designed to mimic the original speaker's tone, highlighted the importance of maintaining the unique style and voice of the text. However, the effectiveness of this strategy was dependent on the explicitness of the author's tone in the original text.

These findings underline the complexity of emotion detection in text. They suggest that a combination of strategies, including providing explicit emotional context, employing language translation mechanics, and mimicking speaker tonality, may be necessary to enhance model accuracy. Despite certain limitations and variable results, the use of these refined prompts still led to an overall improvement in performance compared to our baseline model. This study thus contributes valuable knowledge for the further development and refinement of AI models in the field of emotion detection.

There are a few things hard to accomplish in the project. First of all, generating augmented text for each original text through GPT-3.5-turbo is time-consuming and usage-limited. The GPT-3.5-turbo’s usage is based on the total tokens (3000 tokens/min) used. So, for huge datasets, we have to choose a partial dataset for training our model and the performance of the model may achieve lower than the one which trains with the whole dataset. Secondly, GPT-3.5-turbo’s training data primarily consists of internet text, which may not cover all domains or specific topics. This can limit the model’s ability to generate diverse and accurate augmentations for certain specialized or domain-specific contexts. Hence, in the Poem Sentiment dataset, GPT-3.5-turbo cannot fully understand the sentiment and metaphor in poems without the user giving proper orders and common semantics to modify the rephrasing process. Thus, ChatGPT cannot generate accurate responses with the Poem sentiment dataset.

For future work, we are utilizing the API of ChatGPT 3.5-turbo at present. However, when we have access to GPT-4, we anticipate that our model’s accuracy might further improve. We assume the pattern should still hold if OPENAI does no change the inside logic for GPT. Given that GPT-4 is a more advanced iteration, we expect it to have a superior understanding and interpretation of the emotional tone in text inputs. Therefore, leveraging GPT-4 in our work could potentially yield more accurate emotion detection and rephrasing results, thus enhancing the effectiveness of our model. Also, there are sometimes we find that the performance will depend on the words we used like even changing one word will make GPT give a completely different style of rephrasing. If we have more time, we may dig into this, but now we hope our findings give an intuition about how should we use ChatGPT for data augmentation.

## References

Dai, H., Liu, Z., Liao, W., Huang, X., Cao, Y., Wu, Z., Zhao, L., Xu, S., Liu, W., Liu, N., Li, S., Zhu, D., Cai, H., Sun, L., Li, Q., Shen, D., Liu, T., and Li, X. (2023). Auggpt: Leveraging chatgpt for text data augmentation. <https://arxiv.org/abs/2302.13007>.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., and Ravi, S. (2020). GoEmotions: A Dataset of Fine-Grained Emotions. In 58th Annual Meeting of the Association for Computational Linguistics (ACL).

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for nlp. <https://arxiv.org/abs/2105.03075>.

LYTinn (2020). sentiment-analysis-tweet.

McAuley, Julian, and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text.

Sheng, E. and Uthus, D. (2020). Investigating societal biases in a poetry composition system.

Wang, C., Wang, J., Qiu, M., Huang, J., and Gao, M. (2021). Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. <https://aclanthology.org/2021.emnlp-main.221/>.

WANG, Y., YAO, Q., KWOK, J. T., and NI, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. <https://arxiv.org/abs/1904.05046>.

Yin, W. (2020). Meta-learning for few-shot natural language processing: A survey. <https://arxiv.org/abs/2007.09604>.