Human-in-the-Loop Text Data Augmentation

Raoyuan Zhao
Technical University of Munich
School of Computation, Information and Technology
raoyuan.zhao@tum.de

Yaling Shen
Technical University of Munich
School of Computation, Information and Technology
ge79zuw@mytum.de

Yiran Li
Technical University of Munich
School of Computation, Information and Technology
ge86sib@mytum.de

Abstract

This paper proposes a text augmentation pipeline combining data augmentation methods in different applied levels and human-in-the-loop technology to increase the model performance on downstream tasks (i.e., text classification). As for human-computer interaction, we allow users to choose the cluster of samples they want to augment, in terms of the distribution parameters we provide, which served as instructions. Furthermore, we enable the user to choose the single wrongly classified data sample that he wants to augment. For augmentation aspects, we compare the augmentation performance on token level (word selection, word deletion, synonym replacement, word swap), sentence level (back translation with Russian and German) as well as embedding levels (two-step interpolation). We also compare the performance of different augmenting hyperparameters with fixed augmentation levels to help our human users make better decisions.

1 Introduction

Data Augmentation in NLP refers to the process of generating additional training data from existing text data to improve the performance of NLP models. It is a crucial technique used to overcome the problem of limited training data in NLP, which may further result in the issue of overfitting, poor generalization, and lower accuracy.

Many data augmentation techniques have been proposed to increase the robustness of models, including token-, sentence-, and hidden-level transformation. Token-level modifications, as summarized by Wei and Zou (2019), include Word insertion, deletion, replacement as well as swap. Common methods for sentence-level data augmentation in NLP include paraphrasing, as introduced by Sennrich et al. (2015), which involves rephrasing a sentence while retaining its meaning. This can be done manually or using a paraphrasing tool Edunov et al. (2018). For embedding-level augmentations, the sample dataset is modified by modifying the word embeddings. Commonly used embedding-level augmentations include word embedding interpolation, referring to taking two word embeddings from different sentences and computing the average of their values to generate a new embedding. This new embedding can be used to generate a new sentence that lies between the original sentences in the embedding space. Both TMix Verma et al. (2019), Chen et al. (2020), as well as SSMix proposed by Yoon et al. (2021) which focus on the saliency of tokens, have achieved good performance. Other innovative work in this area which makes use of the variations of interpolation technologies includes puzzlemix (Kim et al. (2020)), mix up with unsupervised learning(Kim et al. (2021)), has also helped us to understand the functionality principle of interpolation better.

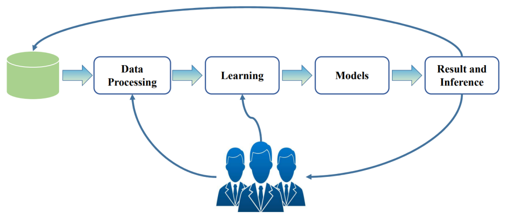

Figure 1: Pipeline of HITL

Human-**i**n-**t**he-**L**oop (HITL) is a process where humans are involved in the decision-making loop of an automated system, providing human feedback and guidance to improve the system's performance. HITL is commonly used in machine learning and artificial intelligence systems where human intervention is required to ensure accuracy and reliability. A common pipeline for HITL can be summarized in Figure 1.

In NLP, HITL has been used for tasks like text classification, syntactic and semantic parsing, sentiment analysis, and so on. A clear survey of HITL in this area can be found in Wu et al. (2022). A classification system for rumors is suggested by Karmakharm et al. (2019). with the idea to use more manual feedback from the journalists to retrain a machine learning model, whereas Liu et al. (2021) has introduced an explainable human-in-the-loop framework for sentiment analysis compensating the lack of the mechanisms to explain when and why the sentiment models give false predictions.

In our work, we combine the above-mentioned two technologies together to provide useful tips for our human users to help them make more reasonable decisions with an unchanged base model architecture. We propose two HITL text augmentation pipelines, one focusing on clusters of text samples, and the other on a single sample, whose workflow is illustrated in Figure 3 and Figure 4, respectively. The experimental results on the downstream text classification task prove that text augmentation can indeed improve the classification performance without changing the model and it is crucial to choose suitable combinations of augmentation methods and hyperparameters, which further prove the importance of involving HITL. Our main contribution is threefold:

- We implement three different types of augmentation methods and allow any combination of these methods to be applied to any data point for data augmentation.
- We propose a HITL cluster augmentation that allows human users to pick the cluster(s) and augmentation method(s) they want to use.
- Experimental results on the text classification task prove the importance and effectiveness of involving humans in the text augmentation.

2 Related Works

Data augmentation has been effective in improving model performance in the field of computer vision, as well as in natural language processing. However, in comparison to image data, text data generates augmented data in a different way and current augmentation methods can be broadly categorized as follows.

Token Level Augmentation Wei and Zou (2019) proposed Easy Data Augmentation (EDA), which consists of four operations on the word token in a sentence: synonyms replacement, word insertion, word deletion, as well as randomly swap.

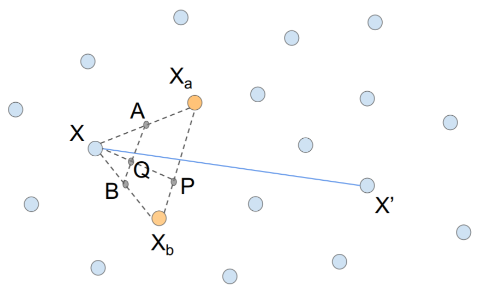

Figure 2: Visualization of Doublemix

Sentence Level Augmentation Sentence-level augmentation methods do not manipulate certain words in a sentence individually but treat the sentence as a whole entity. Sennrich et al. (2015), Edunov et al. (2018) both use back translation as an augmentation method, which has the effect of making the classification model more robust. Min et al. (2020) also suggested that syntactic rewriting can also enhance the robustness of inference models by experimenting with the syntactic rewriting of sentences from the MNLI.

Embedding Level Augmentation The embedding level enhancement is different from the above two methods. It generates new data based on embeddings of text data. One such approach is proposed in DOUBLEMIX by Chen et al. (2022), which executes a two-step interpolation at a specific chosen hidden level of the Bert model. At the first step of the interpolation, DOUBLEMIX mixes up the perturbed data into a synthetic sample, followed by the second step at which the newly synthesized sample will be mixed up with the original data further. A simple Visualization of the idea behind DOUBLEMIX can be seen in Figure 2, the X is the original data sample in this case, X' can be seen as the results of other proposed embedding levels' new data sample, whereas DOUBLEMIX concentrates more on the region nearby the original data sample. By interpolation in the first step, we obtain P, lying on the connecting line of two perturbed new samples, and in the second step, the interpolation along the line connecting X and P will be executed.

Auxiliaries By the visualization of high dimensional embeddings, we use TSNE as proposed by Van der Maaten and Hinton (2008) to downsize the samples and meanwhile keep the similarity between the original and downsized samples. By interaction with humans in our loop, we choose kmeans Hamerly and Elkan (2003) to classify the samples into different clusters and let our user to chose the cluster that he/she wants to augment, based on the variance of the cluster. Whereas for the correction of the prediction result of a single sample, we chose KNN Guo et al. (2003) to allow users to finetune how many neighbors around the misclassified sample he/she would like to augment.

3 HITL Text Augmentation Pipeline

3.1 Cluster Augmentation

The overall workflow of cluster augmentation is illustrated in Figure 3. Before human interaction, we first roughly classify the TREC dataset into six (number of ground truth labels) clusters based on text sentence embeddings via the K-means method and provide cluster information such as the number of samples in each cluster, and cluster variance to our users for reference.

Then we allow our users to pick the cluster(s) they want to augment as well as the augmentation method(s) they want to apply. All three levels of augmentation methods can be chosen here. By interacting with our human users, a specific cluster will be picked and further fed into the data augmentation module to get more samples. Within the module, different augmenting methods can be executed based on the characteristics of the different fed data set.

The performance of augmented data is represented by the numerical experimental results of the downstream text classification task as well as the cluster visualization of the embeddings, based on which, human users can modify their augmentation choices, and therefore form a loop in our text augmentation pipeline.

The T-SNE is used to decrease data dimension and give human users a more intuitive visualization of the clustering result before and after data augmentation, comparing with the numerical distribution evidence, the user can decide to pick which cluster. The final result after the augmentation of a specific cluster will also be visualized to enable the user to have a better understanding of the influence of their choice without the demand for professional knowledge.

3.2 Single Sample Augmentation

To solve the issue that some of the test samples may be wrongly classified even after the augmentation of the whole cluster, we allow human users to continue interacting with our model to choose the single misclassified sample that they want to augment and rectify by further augment the chosen sample as well as its neighbors.

The overall structure of single sample augmentation for prediction correction is shown in Figure 4. We first allow users to pick the single sample they want to augment in the test set, which can be wrongly classified based on the former model. Then we find the k nearest neighbors of the chosen data sample in our training set by computing the distances of their text embeddings (cosine similarity as demonstrated in Equation 1, where e_1 and e_2 are the text embeddings of two chosen samples), while the set for validation and test remains unchanged. Then, we augment the chosen sample k neighbors in the train set and feed the augmented dataset into our model again.

$$\cos(e_1, e_2) = \frac{e_1 \cdot e_2}{||e_1||||e_2||} \tag{1}$$

To enhance the maneuverability of our users, we also provide the opportunity for our users to finetune the hyperparameters, namely the number of neighbors (k) that they would like to concentrate on. However, to guarantee that such an application will not decrease our model performance, based on our experiments on several possible k values, we will suggest our users keep the k value in some certain value interval.

4 Implementation

`DoubleMix` has assembled the above categories of methods previously described, and our work is primarily inspired by them. The main difference and innovation of our work from `DoubleMix` is that we do not augment all training data indiscriminately, but selectively choose augmentation samples in the expectation of finding the augmentation sample and augmentation methods with the greatest gain.

4.1 Dataset and Baseline Model

We use TREC as our data set, which contains 5500 labeled English interrogative sentences in the training set and another 500 for the test set.

The task is to determine what is being asked of the sentence. There are six coarse labels and 50 final labels in the dataset. Given that the dataset is relatively small, we choose the coarse label as the ground truth, which comprises: Abbreviation, Entity, Description and abstract concept, Human being, Location, and Numeric value.

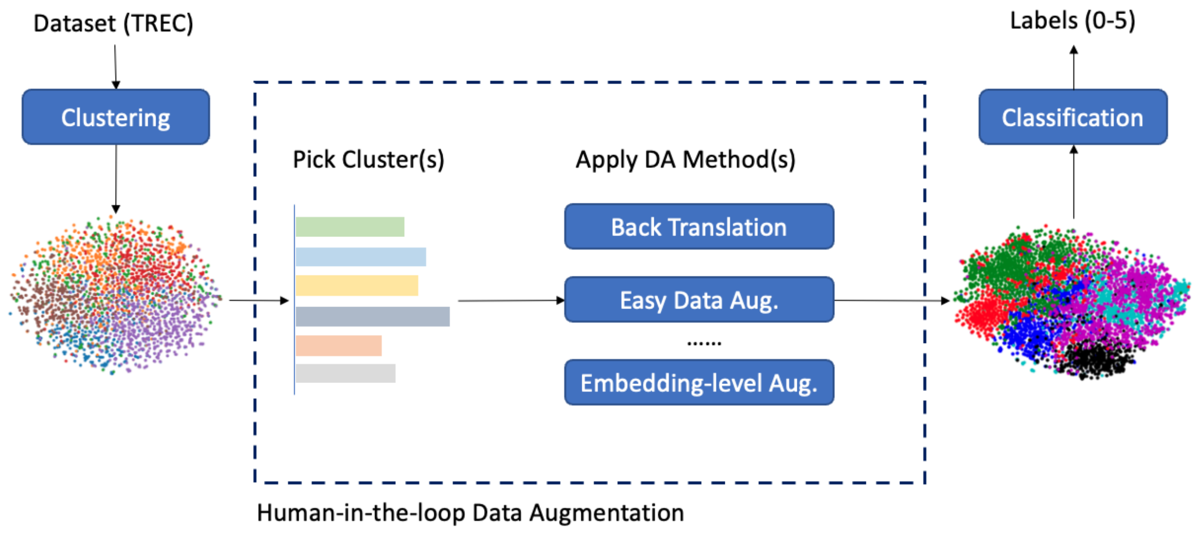

Figure 3: Pipeline of cluster augmentation

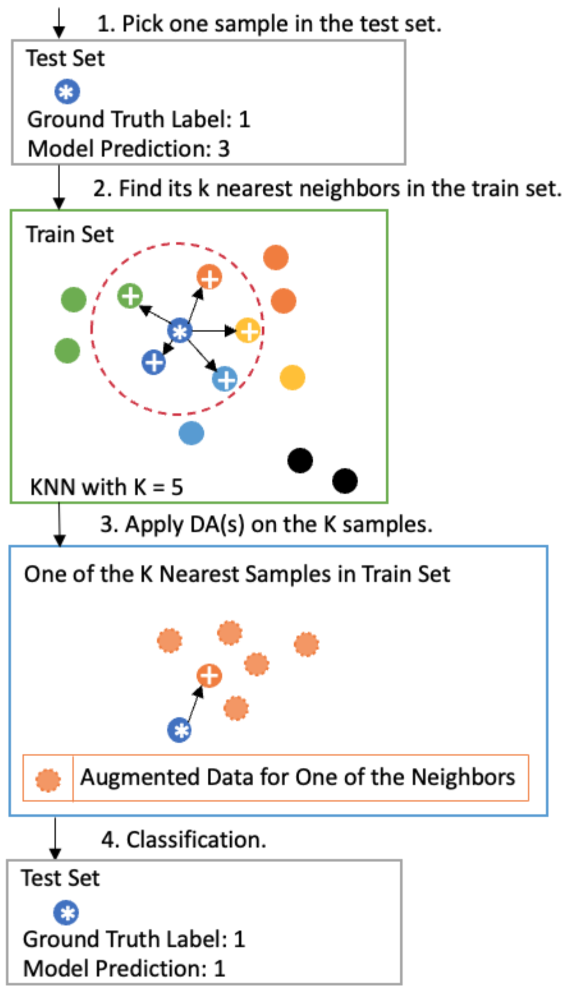
1. Pick one sample in the test set.
Test Set
Ground Truth Label: 1
Model Prediction: 3
2. Find its k nearest neighbors in the train set.
Train Set
KNN with K = 5
3. Apply DA(s) on the K samples.
One of the K Nearest Samples in Train Set
Augmented Data for One of the Neighbors
4. Classification.
Test Set
Ground Truth Label: 1
Model Prediction: 1

Figure 4: Augmentation on single sample

For the baseline model, we choose the popular used model BertForSequenceClassification (bert_uncased), so that our experimental result is able to be compared with the existing results of the related works.

4.2 Augmentation Methods

We implement each of the three types of augmentation mentioned previously.

Token Level Four sub-methods of *Easy Data Augmentation (EDA)* are implemented for token-level text augmentation, including synonyms replacement, word insertion, word deletion, and random swap. Additionally, we allow our users to adjust the number of words they want to change in the sentence accordingly.

Sentence Level For sentence-level data augmentation, we attempt to implement back translation and syntactic data augmentation. As for *back translation*, the translation model from the `fairseq` library is used to translate the original sentences into German or Russian and then translate them back into English to obtain augmentation samples. Compared to EDA, this method ensures that the generated sentences conform to the syntactic rules, but the sentence semantics may sometimes be off. For *syntactic data augmentation*, as the data set is interrogative, interrogative sentences can have many different syntactic structures, which is more difficult to rewrite compared to a uniformly structured declarative sentence, we have only implemented rewriting of special interrogative sentences, which

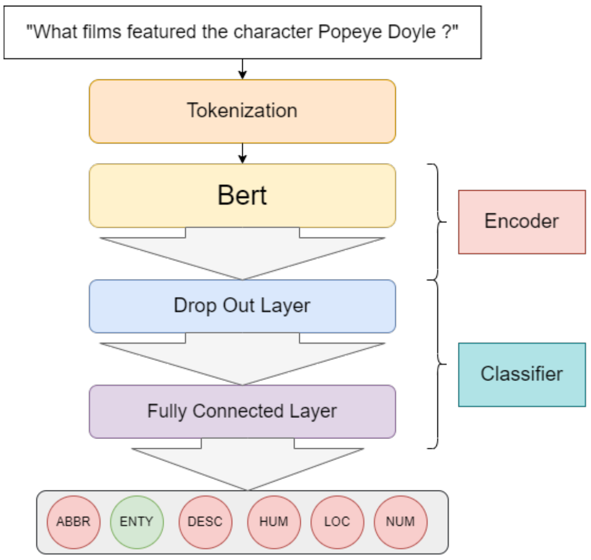

Figure 5: Model Structure

is not possible for every data, and initial tests of the enhancement effect of this enhancement method are not good, so we have not furthered this enhancement method.

Embedding Level For embedding level data augmentation, we have implemented a method where, for selected samples apply a method for calculating distances, as for instance the cosine similarity previously presented. Then the top K closest samples to the sentence vector are found in the training set, and the new vectors generated by averaging these K vectors in pairs are used as the new data for augmentation.

4.3 Train and Test

For the token level and sentence level methods, the new data generated are natural text, so they can simply be added to the original dataset along with the labels of the original samples during training. The new data generated by the embedding level is a word vector, which cannot be mapped back to sentences, and the BERT model cannot modify the type of input, so it cannot simply be incorporated directly into the original training set for training. To solve this problem, we built our own model mimicking the structure of the model, which consists of an encoder part and a classifier part containing a dropout layer and a fully connected layer. The structure can be seen in Figure 5.

The model was first trained with the original dataset, then its encoder was extract which was used to encode the whole training set to obtain the corresponding sentence embeddings for subsequent nearest sample search, and then train only the classifer part separately. To use both embedding level augmentation and token and sentence level augmentation, the encoder should be extracted after the training set has been trained with the full model and new sentences have been added.

In training, the epoch is 3, the learning rate is 0.01 and the batch size is 16.

For the first two methods, the model used for testing and training are identical, whereas for the embedding level, the encoder and classifier are spliced together for testing.

4.4 Data Selection

With the capability to generate new samples, we need to consider which data to select as the original sample for augmentation.

We considered two approaches: one is to select data points for augmentation; the other is The cluster augmentation, which is proposed in view of the fact that if the user wants to augment a large amount of data, it would be too cumbersome to specify so many indexes of the data points manually. So we allow users to cluster the data first and then choose one cluster as the sample. There are also many options for data clustering, and the T-SNE used by pipeline, described previously, is a prevalent clustering method.

5 Results and Analysis

We have evaluated our work based on different clusters as well as different data sizes on the downstream text classification task. Furthermore, we have also compared the results using different data augmentation levels with different combinations of hyperparameters, as shown in Table 1.

We can see that although the baseline Bert has already performed really well, we can still further improve the accuracy and F1 score after data augmentation. Concretely speaking, for all augmented original sets, the best augmentation result among three different levels has managed to increase the accuracy by at least 1 percent.

Comparing the result from single cluster augmentation and whole dataset augmentation, it is worth noting that, the method that performs best is different among the different sizes of datasets. Although embedding level augmentation is able to achieve an excellent improvement when it is used to augment the whole training dataset, it may not suitable for the cases when only a specific sub-

Method		TREC (cluster 4)			TREC (cluster 2)			TREC		
		Acc.	F1	Support	Acc.	F1	Support	Acc.	F1	Support
Baseline: BERT		96.60	95.64	5452	96.60	95.64	5452	96.60	95.64	5452
Easy Data Augmentation (rate= 0.1)	+ Synonym Replacement	97.40	96.08	6557	97.40	96.96	6123	97.00	95.16	10904
	+ Random Swap	**97.80**	**97.29**	6557	**97.60**	**97.92**	6123	**97.40**	96.94	10904
	+ Random Insertion	97.40	96.25	6557	97.20	96.74	6123	97.00	95.04	10904
	+ Random Deletion	**97.80**	97.21	6557	**97.60**	97.12	6123	96.40	95.32	10904
	+ Mix All	97.20	96.80	**9872**	97.00	96.46	**8136**	**97.40**	**97.03**	**27260**
	+ Mix Weighted	97.60	96.97	6557	97.40	96.02	6123	**97.40**	95.47	10904
Back Translation	+ German	97.40	96.81	6557	96.60	96.27	6123	96.80	95.82	10904
	+ Russian	**97.60**	97.12	6557	**97.60**	**97.14**	6123	**97.60**	**97.14**	10904
	+ Mix All	97.40	**97.78**	**7662**	97.40	96.92	**6794**	96.60	96.25	**16356**
	+ Mix Weighted	97.40	96.83	6557	96.60	97.03	6123	97.20	95.93	10904
Embedding Augmentation	+ Nearest Neighbor (NN) = 5	**97.20**	96.81	5572	**97.40**	**96.84**	5572	97.00	96.45	5572
	+ NN = 10	**97.20**	96.77	5992	97.20	96.65	5992	**97.20**	**96.64**	5992
	+ NN = 15	96.60	**96.88**	6712	96.80	96.23	6712	94.40	90.42	6712
	+ NN = 20	92.80	92.33	**7732**	94.80	95.16	**7732**	91.40	76.05	**7732**

Table 1: Experimental Results

	Precision	Recall	F1 Score
Before	97.70	90.43	93.92
After	**97.73**	**91.49**	**94.51**

Table 2: Experimental results before and after single sample augmentation for samples whose ground truth label is 1.

region of the dataset is chosen and need to be augmented. In such a special case, namely, augment for a specifically chosen region, the simple and straightforward EDA and back translation perform better. Additionally, as the number of the nearest neighbor get larger, the accuracy of our model firstly increase, and afterward decrease again. Considering the average result of three different clusters and data size, the embedding level augmentation achieves its best performance when the number of the nearest neighbor is around 5 to 10.

A case study using the proposed single sample augmentation pipeline to correct the former model prediction has been shown in Table 2. Prediction performance on samples with ground truth label 1 has been improved by augmenting the 5 nearest samples in the train set as shown in Figure 4.

6 Conclusions and Future Works

All in all, we have implemented a pipeline that involved the technology data augmentation and human-in-the-loop model modification together and provide some detailed observations at both aspects.

For data augmentation, we have experimented with mainstream data augmentation methods in different levels, and different size of datasets. We find that, the best performance augmentation methods can be different along different dataset situations. It is worth trying different experimental settings when data augmentation methods is applied. In our case, when dataset is quite small, token and sentence level augmentation are prefered, whereas for larger dataset, embedding level outperform with higher accuracy improvement.

To the human-in-the-loop aspect, we provide with opportunity letting human users involve in the modification of our model in different modules and give them suggestions to help them make their decisions. In special case of combining knn and augmentation, keeping the number of k within a region between 5-10 would be a smart choice.

Further studies can be dived into, for example, combining the EDA, Sentence as well as embedding levels augmentation together. Namely, instead of using a single method to augment the data set, a sequential application of different levels methods may lead to the advantages assembling and thus the data set size issue can be ignored.

Acknowledgements

We would like to thank the social computing group at TUM for organising and providing us with this wonderful practical course and MSc. Edorado Mosca and Dr. Daryna Dementievay for their professional and kind guidance.

References

Hui Chen, Wei Han, Diyi Yang, and Soujanya Poria. 2022. Doublemix: Simple interpolation-based data augmentation for text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4622–4632.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. 2003. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer.

Greg Hamerly and Charles Elkan. 2003. Learning the k in k-means. *Advances in neural information processing systems*, 16.

Twin Karmakharm, Nikolaos Aletras, and Kalina Bontcheva. 2019. Journalist-in-the-loop: Continuous learning as a service for rumour analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 115–120.

Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. 2020. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR.

Yekyung Kim, Seohyeong Jeong, and Kyunghyun Cho. 2021. Linda: Unsupervised learning to interpolate in natural language processing. *arXiv preprint arXiv:2112.13969*.

Zhe Liu, Yufan Guo, and Jalal Mahmud. 2021. When and why does a model fail? a human-in-the-loop error detection framework for sentiment analysis. *arXiv preprint arXiv:2106.00954*.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. *arXiv preprint arXiv:2004.11999*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*.

Soyoung Yoon, Gyuwan Kim, and Kyumin Park. 2021. Ssmix: Saliency-based span mixup for text classification. *arXiv preprint arXiv:2106.08062*.