Generation-driven Contrastive Self-training for Zero-shot Text Classification with Instruction-tuned GPT

Anonymous ACL submission

Abstract

The remarkable performance of large language models (LLMs) in zero-shot language understanding has garnered significant attention. However, employing LLMs for large-scale inference or domain-specific fine-tuning requires immense computational resources due to their substantial model size. To overcome these limitations, we introduce a novel method, namely GENCO, which leverages the strong generative power of LLMs to assist in training a smaller and more adaptable language model. In our method, an LLM plays an important role in the self-training loop of a smaller model in two important ways. Firstly, we utilize an LLM to generate multiple augmented texts for each input instance to enhance its semantic meaning for better understanding. Secondly, we additionally generate high-quality training instances conditioned on predicted labels, ensuring the generated texts are relevant to the labels. In this way, GENCO not only corrects the errors of predicted labels during self-training but also eliminates the need for extensive unlabeled texts. In our experiments, GENCO outperforms previous state-of-the-art methods when only limited ($< 5\%$ of original) in-domain text data is available. Notably, our approach surpasses Alpaca-7B with human instructions, highlighting the significance of self-training.

1 Introduction

Zero-shot text classification poses a challenge in predicting class labels for text instances without requiring labeled instances for supervised training. Effective solutions to this problem is crucial for many real-world applications, as it diminishes the labor-intensive process of manual labeling. With the remarkable advancements of large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022) in recent years, exploiting the generative capabilities of such models to tackle zero-shot text classification problems has emerged as a critical research question.

Recent research in zero-shot text classification primarily falls into two distinct groups. The first approach applies LLM (with billions of parameters) in label prediction with the help of human instructions or prompts (Ouyang et al., 2022; Chiang et al., 2023a). However, even a relatively smaller LLM such as Alpaca-7B (Taori et al., 2023) necessitate considerable computational power and time for large-scale inference and model fine-tuning. Without domain-specific fine-tuning, LLMs struggle to discern between classes characterized by unclear decision boundaries.

The second approach to zero-shot classification involves the self-training of smaller language models, often comparable in size to BERT (Meng et al., 2020; Schick and Schütze, 2020; Gera et al., 2022; Wang et al., 2023). In these methods, the models predict "pseudo labels" for unlabeled instances, and then use these instances alongside their assigned pseudo labels as supervised data for model fine-tuning. This process is iterated for the model to incrementally adapt to the target domain. However, these techniques hinge on accessing a substantial volume of unlabeled texts from the intended domain, sometimes reaching the magnitude of millions as indicated in table 1, a volume that may not always be feasible in many practical contexts. Furthermore, due to the capacity limitation of small language models, the pseudo label predictions are prone to error potentially jeopardizing the efficacy of the self-training loops.

In this paper, we introduce a novel approach called **Gen**eration-driven **Co**ntrastive Self-Training (GENCO). This approach adeptly combines the language understanding ability of LLMs with the adaptability and efficiency of smaller models. Drawing inspiration from PESCO (Wang et al., 2023), we treat zero-shot classification as a sentence alignment task and employ contrastive self-training with smaller models. We provide a theoretical analysis of how self-training can bolster classification generalization. Crucially, we sidestep the dependency on extensive unlabeled texts by capitalizing on the generative strengths of LLMs.

Our approach exploits the LLM generation power in two ways. Firstly, to enhance pseudo label prediction, we employ an LLM to generate multiple variations or extensions of an input text. This augmentation strategy enriches the available information for the classifier, enabling it to make better predictions based on a more comprehensive understanding of the input. Secondly, we employ the LLM to craft new training instances conditioned on the pseudo labels, ensuring the generated content is closely aligned with its assigned pseudo label. This tackles the prevalent issue of mislabeling in self-training. In summary, this paper makes three key contributions:

- We propose a novel approach that enables smaller models to acquire knowledge from LLMs within the self-training loop. Our method is compatible with any new LLMs to effectively train better classifier on target domains. In our experiments, our small model outperforms Alpaca with human instructions.
- We explore the more challenging setting of zero-shot classification where only a limited number of unlabeled texts are available. In this setting, we improve the performance over strong baselines.
- We provide theoretical proof to support the effectiveness of the proposed contrastive loss for self-training.

2 Preliminary: Zero-shot Text Classification as Sentence Alignment

Given a set of N unlabeled documents $X = \{x_1, x_2, \cdots, x_N\}$ and a set of L category descriptions $C = \{c_1, c_2, \cdots, c_L\}$, the goal is to learn a scoring function $g(x, c_i)$ that takes document x and label description c_i as input and produces a similarity score as the measure of how well the document and the label match to each other.

In the zero-shot setting, text classification can be formulated as a sentence alignment problem (Wang et al., 2023), where both the input sentence and the label descriptions are encoded using a pre-trained sentence encoder like SimCSE (Gao et al., 2021). The similarity scores between the sentence and label embeddings are used to predict related labels. The performance can be further improved by converting a short label description into a full sentence via prompts (Wang et al., 2023; Hong et al., 2022). For example, the label “sports" can be converted to “This is an article about sports." Subsequently, we represent the label prompt for a label description c_i as p_i. The scoring function can be implemented as follows:

$$g(x, c_i) = \text{sim}\left(f_\theta(x), f_\theta(p_i)\right) \tag{1}$$

where $f_\theta(\cdot)$ is the sentence encoder parameterized by θ and $\text{sim}(\cdot, \cdot)$ is a similarity function such as dot product or cosine similarity.

Given an input text at inference time, the predicted label is the one with the highest similarity score:

$$\hat{y} = \arg\max_j g\left(x, c_j\right) \tag{2}$$

3 Our Method: GenCo

GenCo is a self-training framework (Meng et al., 2020; Schick et al., 2021; Wang et al., 2023) that harnesses the generative power of LLMs to train a smaller pre-trained sentence encoder in an iterative manner. Each self-training step consists of two parts. First, we apply equation 2 to predict pseudo labels for unlabeled instances. Second, we finetune model on pseudo-labeled data with a proposed contrastive self-training objective. In section 3.2 and 3.3, we will introduce two types of augmentation with LLM to enhance the self-training process.

3.1 Contrastive Self-Training Objective

One well-known challenge of self-training is its tendency to exhibit overconfidence in certain labels due to the model inductive bias (Xie et al., 2016). Extensive research has shown that soft labeling (Xie et al., 2016; Meng et al., 2020), label smoothing (Müller et al., 2019), and entropy regularization (Grandvalet and Bengio, 2004) can effectively tackle this issue. Motivated by these, we propose to incorporate soft-labeling and entropy regularization into a contrastive loss.

Given an input text x, the distribution of the predicted label space is:

$$P(\hat{y}_i | x; \theta) = \frac{\exp(\text{sim}(f_\theta(x), f_\theta(p_i)))}{\sum_{c \in C} \exp(\text{sim}(f_\theta(x), f_\theta(p_c)))} \tag{3}$$

Here, $\hat{y}_i$ is the predicted label and p_i is a label prompt for the predicted label. To prevent the

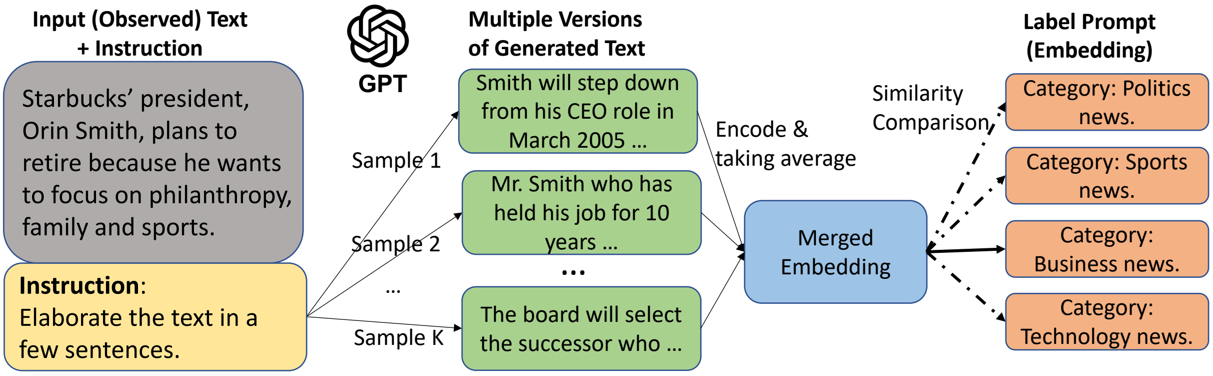
Input (Observed) Text
+ Instruction
Starbucks' president, Orin Smith, plans to retire because he wants to focus on philanthropy, family and sports.
Instruction: Elaborate the text in a few sentences.
GPT
Sample 1
Sample 2
...
Sample K
Multiple Versions of Generated Text
Smith will step down from his CEO role in March 2005 ...
Mr. Smith who has held his job for 10 years ...
...
The board will select the successor who ...
Encode & taking average
Merged Embedding
Similarity Comparison
Label Prompt (Embedding)
Category: Politics news.
Category: Sports news.
Category: Business news.
Category: Technology news.

Figure 1: Enriching textual semantics through LLM Generation: The input text and an instruction are fed into the LLM to generate multiple pieces of elaborated texts, each of which is concatenated to the original input to obtain an augmented text. The embeddings of the augmented texts are then averaged to obtain a merged embedding, which is used for label prediction and contrastive loss in the self-training process.

model from being overconfident, we define the weights of the labels as:

$$Q(\hat{y}_i|x;\theta) = \frac{\exp(\text{sim}(f_\theta(x), f_\theta(p_i))/\tau)}{\sum_{c\in C}\exp(\text{sim}(f_\theta(x), f_\theta(p_c))/\tau)} \quad (4)$$

, where $\tau \leq 1$ is the temperature. A lower temperature implies a sharper distribution and thus greater weights in the predicted label. We drop the notation of θ for convenience.

Combining the above $P(\hat{y}_i|x)$ and $Q(\hat{y}_i|x)$, we propose a text to label ($t2l$) contrastive loss:

$$\mathcal{L}_{t2l} = -\sum_{i=1}^{N}\sum_{j=1}^{L} Q(\hat{y}_j|x_i)\log P(\hat{y}_j|x_i) \quad (5)$$

When $\tau \to 0$, $Q(\hat{y}|x)$ becomes categorical distribution and the loss reduces to a supervised contrastive learning loss (Khosla et al., 2020) with pseudo label $\hat{y}$ as the target:

$$\mathcal{L}_{t2l}^{\tau\to 0} = -\sum_{i=1}^{N}\log P(\hat{y}|x_i) \quad (6)$$

It encourages the model to predict label $\hat{y}$ given x with more confident. On the other hand, when $\tau = 1$, the loss reduces to a minimization of conditional entropy function H:

$$\mathcal{L}_{t2l}^{\tau=1} = H(C \mid X) \quad (7)$$

$$= -\sum_{i=1}^{N}\sum_{j=1}^{L} P(\hat{y}_j|x_i)\log P(\hat{y}_j|x_i) \quad (8)$$

We show a theorem such that minimizing the loss function equation 5 can achieve similar effects Entropy Regularization (Grandvalet and Bengio, 2006, 2004), which is a means to enforce the cluster assumption such that the decision boundary should lie in low-density regions to improve generalization performance (Chapelle and Zien, 2005).

Theorem 1. *Consider a binary classification problem with linearly separable labeled examples. When $0 < \tau < 1$, optimizing equation 5 with gradient descend will enforce the larger margin between classes and achieves max margin classifier under certain constraint.*

We place our formal theorems and proofs in Appendix B. Theorem 2 suggests that self-training is an in-domain fine-tuning that maximizes class separation, which serves as an explanation of why training on pseudo labels can enhance performance even if no extra labeling information is provided. In our experiment, we show that self-training of a smaller model can outperform LLM (Alpaca-7B) prediction, justifying the claim empirically. We set $\tau = 0.1$ (refer to Appendix A.2) to balance supervised classification and low density separation between classes.

While self-training can potentially improve model generalization, the limitations are obvious: 1) pseudo labels are prone to error and may negatively affect model training. 2) self-learning requires a significant load of unlabeled data, which may not always be available. Next, we introduce generation-driven approaches to improve self-training with LLM, such as an instruction-tuned GPT (Alpaca-7B).

3.2 Semantic Enrichment using LLM

In this section, we propose a way to enrich the semantic information of an input text with multiple

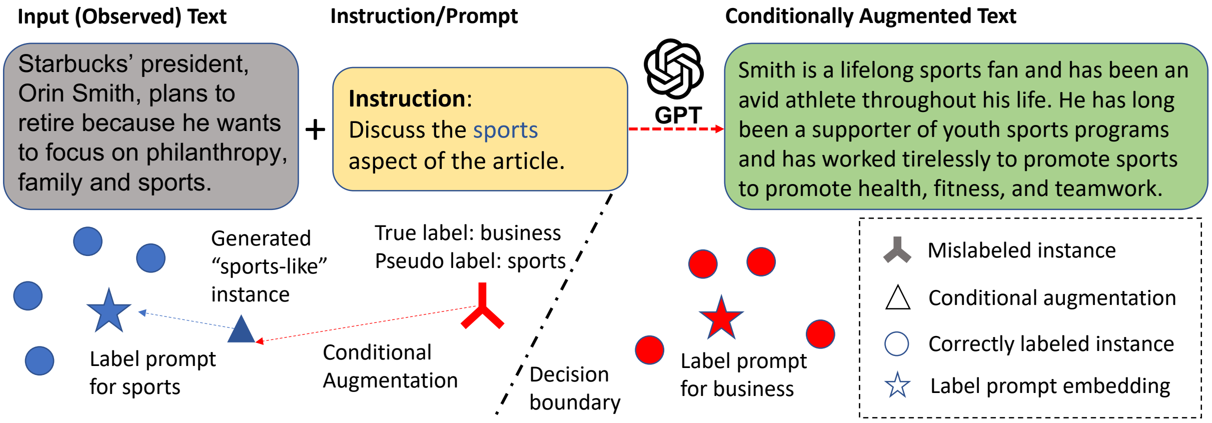
Input (Observed) Text
Starbucks' president, Orin Smith, plans to retire because he wants to focus on philanthropy, family and sports.
+
Instruction/Prompt
Instruction: Discuss the sports aspect of the article.
GPT
Conditionally Augmented Text
Smith is a lifelong sports fan and has been an avid athlete throughout his life. He has long been a supporter of youth sports programs and has worked tirelessly to promote sports to promote health, fitness, and teamwork.
Generated "sports-like" instance
True label: business
Pseudo label: sports
Label prompt for sports
Conditional Augmentation
Decision boundary
Label prompt for business
Mislabeled instance
Conditional augmentation
Correctly labeled instance
Label prompt embedding

Figure 2: Conditional text augmentation to address mislabeling in self-training: When a pseudo label is incorrect, it can mislead the training process and decrease classification performance. We generate augmented text conditioned on the pseudo label, aiming to make the generated text closer to the majority members in the category of the pseudo label. This approach aims to improve the quality of the generated instances for self-training.

LLM-generated pieces of text. When the input text is relatively short, such as consisting of only one or a few sentences, the information may not be sufficient for alignment-based method to match relevant labels.

A remedy is to query an LLM to elaborate the input and generate multiple pieces of extended texts. As shown in figure 1, the instruction, "Elaborate the text with a few sentences," steers the LLM towards creating relevant expansions and continuations for the input text x. These augmented texts, denoted as x^{aug}, serve for two purposes: 1) improving the quality of pseudo label, and 2) forming the positive pair in contrastive learning, as detailed below:

Enhancing pseudo label quality. We enhance pseudo label prediction by enriching the input embedding of equation 2 by:

$$\frac{1}{K}\sum_{i=1}^{K} f_\theta(x \oplus x_i^{\text{aug}}), \tag{9}$$

where $\oplus$ is the concatenation operator for text and x_i^{aug} is the i-th sample from $P_g(\cdot|x)$. The mean of the embeddings summarize the information induced by LLM.

Constructing positive training pairs. We propose a contrastive loss between input text and generated text as another training objective. Let I be a training batch and $A(i)$ be the set of augmented texts with the same pseudo-label as input x_i. Our objective encourages proximity between x and x^{aug} (sampled from $A(i)$) in the embedding space:

$$\mathcal{L}_{t2g} = \sum_{i\in I} \frac{-1}{|A(i)|} \sum_{x^{\text{aug}}\in A(i)} \log \frac{\exp(sim(f_\theta(x_i), f_\theta(x^{\text{aug}})))}{\sum_{j\in I}\exp(sim(f_\theta(x_i), f_\theta(x_j)))}. \tag{10}$$

3.3 Crafting Training Pairs with LLM

Self-training can introduce bias into a classifier due to mislabeling instances. To address this issue, we propose to generate high quality pseudo-labeled data pairs, as shown in figure 2. Consider an instance where an article about the retirement of Starbucks' president, whose true label is "business", is mistakenly labeled as "sports". Training the model with this incorrect label blurs the distinction between the business and sports categories.

To mitigate this issue, we employ the LLM to conditionally augment the input text based on the sports category. This is achieved by framing instructions like, "Discuss the sports aspects of the article". Consequently, the produced text mirrors typical articles within the sports category. By optimizing this newly generated text, instead of the original mislabeled instance, we correct its placement relative to the decision boundary separating "sports" and "business". Essentially, by creating texts based on pseudo labels, we synthesize training pairs that enhance the separation of class labels in the embedding space, thereby addressing the challenges of mislabeling inherent to self-training.

Let x^{cond} be the conditionally augmented text,

Algorithm 1: Self-training with GPT assisted in the loop

```
Require: Unlabeled texts $X$, label descriptions $C$, instruction-tuned GPT model $g(\cdot)$.
Initialization: Classifier $f_\theta(\cdot)$ initialized with pre-trained sentence encoder. Empty dictionary
    $\mathrm{GenDict}$ to cache conditional generated text.
Input augmentation: For each observed text, generate $K$ samples of augmented text from $P_g(\cdot|x)$.
for $t: 1 \rightarrow T$ self-training iterations do
    Use $f_\theta(\cdot)$ to generate pseudo-labels $\hat{y}$ (eq.2) and soft-target $Q$ (eq.4) for texts with input
      augmentation in Section.3.2. Sample a balanced subset of pseudo-labeled training pairs of
      size $S_t$ according to prediction confidence;
    for each training sample $(x, \hat{y})$ do
        if key $(x, \hat{y}) \in \mathrm{GenDict}$ then
            Fetch generated texts from $\mathrm{GenDict}$          ▷ Use cached generated text;
        else
            Generate $M$ samples from $P_g(\cdot|x, \hat{y})$     ▷ Conditional augmentation in Section 3.3;
            Add generated texts to $\mathrm{GenDict}$             ▷ Cached generated text;
    Use sampled training pairs and the conditionally generated text to update the parameters $\theta$ of
      $f_\theta(\cdot)$ with the objective function $\mathcal{L} = \mathcal{L}_{g2l} + \mathcal{L}_{t2g}$ from equation 10 and 11.
```

the modified equation 5 is:

$$\mathcal{L}_{g2l} = -\sum_{i=1}^{N}\sum_{j=1}^{L} Q(\hat{y}_j|x_i^{\text{cond}}) \log P(\hat{y}_j|x_i^{\text{cond}}) \quad (11)$$

3.4 Algorithm for Self-training

We apply self-training with equation 10 and 11 in an iterative way as shown in Algorithm 1 with LLM assisting in the loop. During training, we found that a balanced sampling that keeps the same number (S_t for iteration t) of training for each category is important for the stability of self-training. Additionally, we use a dictionary $\mathrm{GenDict}$ to cache the conditional generated text to avoid repeated generation for better efficiency.

4 Experiments

4.1 Datasets and Experimental Settings

We conduct experiments on 4 benchmark text classification datasets: AG News, DBpedia, Yahoo Answers and Amazon, with the statistics shown in table 1. In the experiments, we initialize our sentence encoder with supervised SimCSE Roberta-base model (110M parameters) (Gao et al., 2021). For the generative model, we use the Alpaca-7B (Taori et al., 2023) as our choice of LLM, which is a GPT model fine-tuned with human instructions (Touvron et al., 2023). The label prompts and the instruction template are illustrated in table 3 in Appendix. Please refer to section A in Appendix for implementation details.

4.2 Baseline Methods

Alpaca-7B is a LLM baseline for zero-shot classification. We solicit the LLM for zero-shot classification with the instruction "Classify the text by outputting a single category from [label categories]".

iPET (Schick and Schütze, 2020) formulates zero-shot text classification as a cloze test, where a pretrained BERT (Devlin et al., 2018) model is used to predict the output label(s) by completing a prompt such as "This article is about _", which is concatenated right after an input document. An iterative self-training algorithm is used in iPET to improve the model for better generalization.

LOTClass (Meng et al., 2020) applies the BERT model to extract keywords related to the label names from unlabeled texts and then create pseudo labels based on the extracted keywords. LOTClass also applies a self-training algorithm to further improve the classification performance.

PESCO (Wang et al., 2023) formulates zero-shot classification as sentence alignment and uses contrastive self-training to improve the model performance. As an augmentation, it selects salient sentences from documents to create additional positive training pairs.

4.3 Experimental Results

In table 2, we present a comparison of the test accuracy of our model with other baselines on four benchmark classification datasets. Specifi-

Dataset	Classification Type	#Classes	#Train	#Test	Avg Length
AG News	News Topic	4	120,000	7,600	38
DBPedia	Wikipedia Topic	14	560,000	70,000	50
Yahoo Answers	Question Answering	10	1,400,000	60,000	70
Amazon	Product Review Sentiment	2	3,600,000	400,000	78

Table 1: Statistics of datasets for multi-class text classification.

ID	Self-train	Methods	AG News	DBpedia	Yahoo Answers	Amazon
1	–	Supervised	94.2	99.3	77.3	97.1
2	No	SimCSE (Sentence-enc)	74.5	73.8	55.6	88.8
3	No	Alpaca-7B (LLM)	77.4	60.6	52.1	86.6
4	Yes	iPET	86.0	85.2	68.2	95.2
5	Yes	LOTClass	86.4	91.1	–	91.6
6	–	Supervised-downsample*	93.8	98.7	76.5	97.0
7	Yes	PESCO*	85.0	96.6	65.8	92.4
8	Yes	GenCo *	**89.2**	**98.3**	**68.7**	**95.4**
9	Yes	GenCo * - CA	87.5	97.6	65.1	94.3
10	Yes	GenCo * - IA	86.2	97.1	63.5	93.6
11	Yes	SimCSE + Self-training (Eq 5)	83.2	94.3	62.7	91.5

Table 2: Comparison of classification methods on benchmark datasets. The test accuracy of best performing zero-shot method is highlighted in bold phase. Row 7-11 (with *) use a down-sampled dataset with 4k (3.4%), 11.2k (2%), 15k (<1%), 20k (<1%) unlabeled training instances respectively. Rows 9-11 are ablation tests with input augmentation (IA) or conditional augmentation (CA) removed.

cally, rows 1-5 are experiments using the entire (unlabeled) training set and rows 6-11 use a down-sampled dataset with 4k (3.4%), 11.2k (2%), 15k (<1%), 20k (<1%) unlabeled training instances from the original datasets respectively.

Comparison with Alpaca-7B: While Alpaca-7B (row 3) has demonstrated strong instruction following ability to solve problems without any training, it exhibits lower performance compared to GenCo (row 8) and other self-training methods on classification task. The reason could be attributed to the domain adaptation effect of self-training. Classification tasks involve comparing instances, such as an article being more likely to belong to the “sports" category when compared to articles in the “business" category. In our analysis in section 3.1, self-training enforces the separation between classes to improve the generalization ability. This can be further supported when the number of classes increases in DBpedia and Yahoo Answers dataset, the performance of Alpaca gets worse. Furthermore, Alpaca-7B takes 9 minutes per 10k instances on one A6000 gpu while GenCo takes 10 seconds, which is roughly x50 speed up.

Comparison with SOTA Methods: Both iPET (row 4) and LOTClass (row 5) use self-training algorithm for zero-shot classification, but GenCo outperforms the previous self-training methods even with significantly fewer instances ($< 5\%$ of original size). The iPET model improves pseudo label prediction with an ensembling about 15 models to reduce prediction variance. In comparison, our approach improves pseudo label prediction by ensembling augmented text embedding during self-training, leading to improved performance and a more memory efficient alternative. While LOTClass uses a BERT model to extract keywords for each category as an augmentation, it is less expressive than using an LLM to generate coherent human language as augmentation. PESCO (row 7) is the most recent SOTA with contrastive self-training and introduced an augmentation technique by learning on salient sentences. However, the method still requires a large amount of data to be effective. In scenarios where only a limited number of unlabeled texts are available, PESCO still underperforms our model.

Effectiveness of Contrastive Self-training: Row 2 represents the sentence encoder baseline with SimCSE, whereas row 11 represents SimCSE +

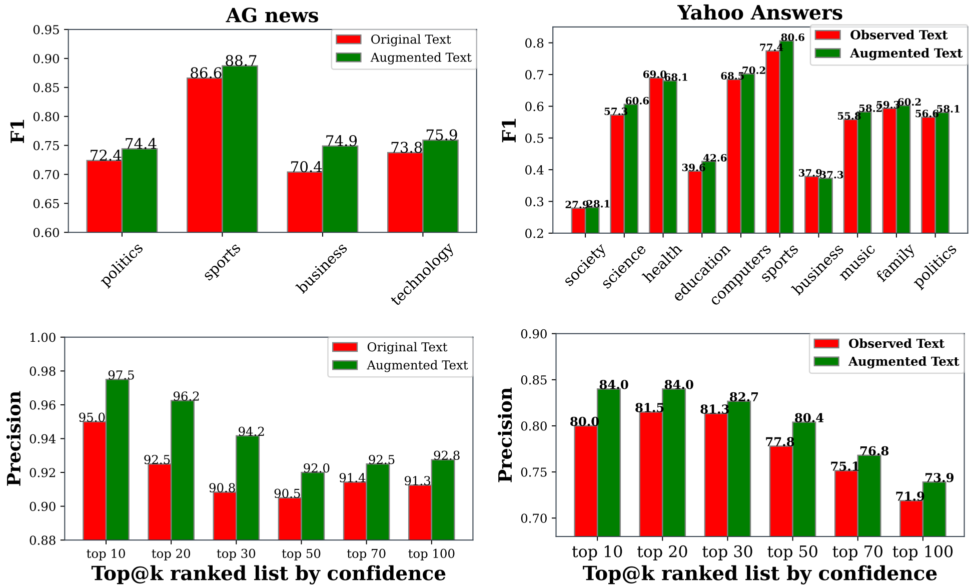

Figure 3: Per class F1 (upper) and ranking-based precision (lower) for classification performance with input augmentation.

contrastive self-training algorithm as per equation 5. The result shows that incorporating contrastive self-training leads to significant gains. Compare row 3 (Alpaca-7B) with row 11. Despite being a larger model in scale, Alpaca-7B still outperforms the self-training approach across all benchmark datasets, underscoring the effectiveness of class separation with self-training for classification task.

4.4 Analysis of LLM Augmentation

In this section, we denote the input augmentation in section 3.2 as IA and the conditional augmentation based on pseudo label in section 3.3 as CA. Rows 9 and 10 in table 2 shows ablation tests with CA and IA removed. Overall, our LLM data augmentation, with and without conditioning on pseudo label, both lead to improved performance, due to their ability to provide more accuracy pseudo label and high quality synthetic training pairs.

Effectiveness of IA: In this evaluation, we investigate the effectiveness of input augmentation for first round pseudo-labeling *without training*. We evaluate the performance of our model on two datasets, namely AG News and Yahoo Answers, using two evaluation metrics: per class F1 metric and ranking-based precision metric according to prediction confidence. The per class F1 metric provides an insight into how well the model performs on each individual class by balancing precision and recall. In the upper part of figure 3, our findings indicate that LLM augmented data leads to improved performance across all categories for AG News and in eight out of ten classes for Yahoo Answers.

In the lower part of figure 3, we employ a ranking-based precision metric to assess the quality of the most confident cases. Our results demonstrate that using augmented data yields better precision for the most confident cases. Notably, our study on the Yahoo Answers dataset indicates that the predictions are better calibrated with the use of augmented data, implying that highly confident samples exhibit better precision. Conversely, such a trend was not observed in unaugmented data, where the top 30 had higher accuracy than the top 10. Better calibration justifies the sampling from the most confident pools for self-training, making it a more reliable method for improving model performance.

Effectiveness of CA: To study the quality of conditional generation based on class labels, we first present examples of generated texts from an sample in AG News dataset, shown in table 6 in Appendix. Each example is a cherry-picked sample out of five random samples. The generated text expands on a specific aspect regarding the label while retaining the original meaning of the observed text.

In the left of figure 4, we show a heatmap of the probability when a conditionally generated text (vertical) aligns with the corresponding label class (horizontal). The highest probability occurs along the diagonal, indicating that the conditionally augmented text based on pseudo label has a closer meaning to the corresponding label class. In the right of figure 4, we plot the distribution of the gen-

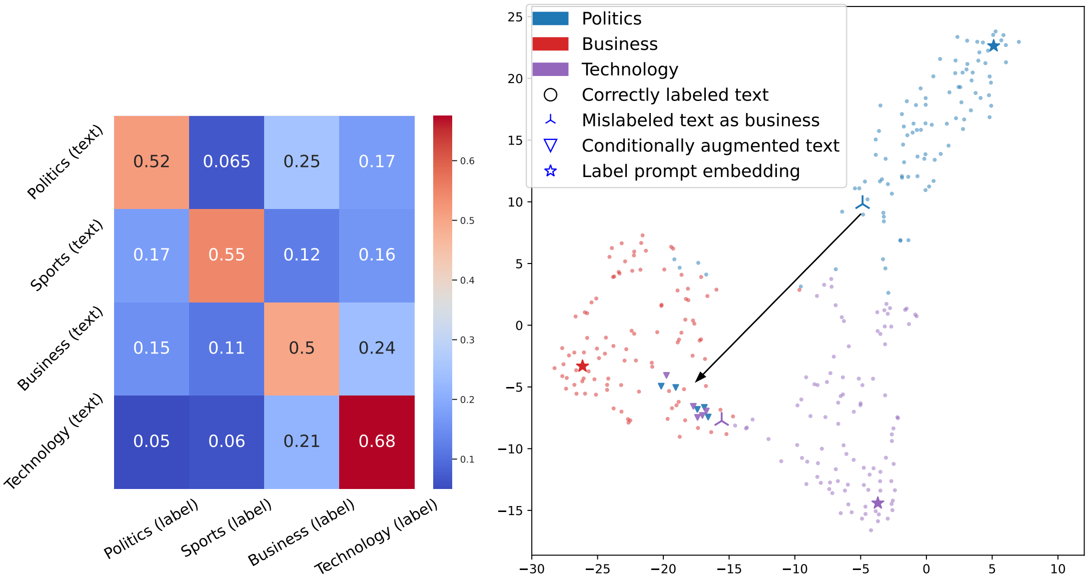

Figure 4: The left figure shows a heatmap of the probability when a conditionally generated text based on pseudo label aligns with each of the label prompts. The right figure shows the distribution of the generated text plotted using T-SNE (sports category is out of scope).

erated text plotted using T-SNE. The embeddings were obtained by our sentence encoder trained on the 100-th (out of 1000) iteration. We selected two instances that were misclassified as business and located close to the decision boundary. The augmented text, conditioned on the business category, was found to be closer to the label prompt embedding of the business category. This demonstrates the effectiveness of our method to generate less confusing training pairs away from the decision boundary and closer to the pseudo label centroid.

5 Related Work

Knowledge Distillation from GPT: To leverage the language modeling power of large model, previous works distills LLM (Honovich et al., 2022; Chiang et al., 2023b), generate text and label pairs (Yoo et al., 2021; Ye et al., 2022; Meng et al., 2022) to train a classifier for downstream tasks. However, generating training data from scratch can lead to low-quality data with unrelated or ambiguous examples analyzed in (Gao et al., 2022). Our generation is grounded in the context of the corpus with enrichment in semantic and diversity, providing a practical alternative to generation-based methods for zero-shot text classification and knowledge distillation.

Zeroshot Text Classification: Zeroshot text classification predicts class labels without labeled instances (Cho et al., 2023; Fei et al., 2022) and can be formulated as sentence alignment (Gao et al., 2021; Hong et al., 2022; Shi et al., 2022; Wang et al., 2023; Zhang et al., 2023) between document and labels. Sentence encoders are typically trained with contrastive learning, which optimizes representations by pulling inputs with similar semantics closer in the embedding space and pushing inputs with different semantics further apart. Our model applies LLM to generate training pairs for contrastive learning to train robust classification with limited instances available.

6 Conclusion

In conclusion, our proposed approach, GenCo, effectively addresses the difficulties and limitations of using LLMs directly for zero-shot text classification. By leveraging the generative power of an LLM in a self-training loop of a smaller, sentence encoder classifier with contrastive learning, GenCo outperform state-of-the-art methods on four benchmark datasets. Our approach is particularly effective when limited in-domain text data are available. The success of our approach highlights the potential benefits of incorporating the generative power of LLM into iterative self-training processes for smaller zero-shot classifiers. We hope that our work will inspire further research in this direction, ultimately leading to more efficient and effective NLP models.

7 Limitations

The main goal of our paper is to promote the usage of LLMs (Alpaca-7B in our case) to assist in training of a smaller model (Roberta-SimCSE) on zero-shot classification tasks. We are aware that there are rooms more experiments with self-training algorithms, such as how the temperature of our loss function can affect the training stability. Currently, we mainly use that as a theoretical motivation of leveraging decision boundaries between classes, but tuning the temperature will be additional work to do.

Another part is data efficiency. We have shown that using GPT generated data can alleviate the data hungry issue for deep learning models. However, when there is abundant of data, generating training instances with LLM can be expensive with less gains. Also, due to compute and buget limitations, we didn't use larger LLMs for our experiments, as an estimited cost will be around 150$ per dataset with the GPT-3.5 at time of writing.

Finally, we realize that more tricks and engineering designs are employed in our experiments and please refer to our code for reference.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Olivier Chapelle and Alexander Zien. 2005. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics*, pages 57–64. PMLR.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023a. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023b. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Hyunsoo Cho, Youna Kim, and Sang-goo Lee. 2023. Celda: Leveraging black-box language model as enhanced classifier without labels. *arXiv preprint arXiv:2306.02693*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yu Fei, Ping Nie, Zhao Meng, Roger Wattenhofer, and Mrinmaya Sachan. 2022. Beyond prompting: Making pre-trained language models better zero-shot learners by clustering representations. *arXiv preprint arXiv:2210.16637*.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2022. Zerogen+: Self-guided high-quality data generation in efficient zero-shot learning. *arXiv preprint arXiv:2205.12679*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. *arXiv preprint arXiv:2210.17541*.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17.

Yves Grandvalet and Yoshua Bengio. 2006. Entropy regularization.

Jimin Hong, Jungsoo Park, Daeyoung Kim, Seongjae Choi, Bokyung Son, and Jaewook Kang. 2022. Tess: Zero-shot classification via textual similarity comparison with prompting using sentence encoder. *arXiv preprint arXiv:2212.10391*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, pages 18661–18673. Curran Associates, Inc.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *arXiv preprint arXiv:2202.04538*.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. *arXiv preprint arXiv:2010.07245*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. Nearest neighbor zero-shot inference. *arXiv preprint arXiv:2205.13792*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yau-Shian Wang, Ta-Chung Chi, Ruohong Zhang, and Yiming Yang. 2023. Pesco: Prompt-enhanced self contrastive learning for zero-shot text classification.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Ruohong Zhang, Yau-Shian Wang, Yiming Yang, Donghan Yu, Tom Vu, and Likun Lei. 2023. Long-tailed extreme multi-label text classification by the retrieval of generated pseudo label descriptions. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1092–1106, Dubrovnik, Croatia. Association for Computational Linguistics.

A Experiments

A.1 Implementation Details

The label prompts are shown in the upper part of table 3. The label prompts are similar to the ones used in in PESCO (Wang et al., 2023). We solicit LLM for text augmentation with the instruction template in the lower part of table 3, which is the same ones used for Alpaca fine-tuning.

For the generation parameters, we used $temperature$=0.8, top_p=0.95, and sample K=5 augmented texts for each instance with $min\_length = 64$ and $max\_length = 128$. For the self-training of sentence encoder model, we used $batch\_size$=$3 * |C|$ ($|C|$ is the number of categories), lr=1e-5, the max length is 128 for AG News and DBPedia and 192 for Yahoo Answers and Amazon. All the experiments are performed on NVIDIA RTX A6000 gpus. Please refer to our code for details.

Label Prompt
(1)Category: [label].
(2)It is about [label].
Instruction-based (Conditional) Augmentation
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Elaborate the text in a few sentences. (Discuss the [pseudo label] aspects of the article.) ### Input: [text] ### Response:

Table 3: The designed prompts for enhanced label description and conditional augmentation based on pseudo label.

A.2 Selection of Temperature in Eq 5

As shown in table 4, we include the results with over 5 runs on each dataset. We found $\tau = 0.1$ to be a reasonble choice with slightly better performance, but we acknowledge that the difference is rather small, sometimes fall within std. The choice of τ may serve more of a theoretical motivation rather than practically concerns (as acknowledged in limitation). The theoretical framework unifies previous soft labeling approaches in (Meng et al., 2020; Wang et al., 2023) and is easier for the proof of theorem.

A.3 Inference Time Augmentation

While GenCo doesn't require LLMs during inference, in our ablation test in table 5, we study the

	Agnews	DBpedia	Yahoo Answers	Amazon
τ=1.0	82.75 ± 0.06	93.77 ± 0.07	62.66 ± 0.06	91.39 ± 0.06
τ=0.5	83.04 ± 0.05	94.19 ± 0.05	62.70 ± 0.10	91.44 ± 0.06
τ=0.1	**83.18** ± **0.05**	94.29 ± 0.05	62.74 ± 0.08	**91.48** ± **0.05**
τ=0.05	83.03 ± 0.05	**94.34** ± **0.03**	**62.77** ± **0.10**	91.42 ± 0.04
τ=0.01	83.02 ± 0.05	94.33 ± 0.03	62.76 ± 0.11	91.42 ± 0.04

Table 4: For the choice of temperature τ in equation 5, we include the results with over 5 runs on each dataset. We found $\tau = 0.1$ to be a reasonble choice with slightly better performance, but we acknowledge that the difference is rather small, sometimes fall within std.

impact of inference time augmentation (assuming GPT is available at test time) and self-training on the performance metric. To test inference time augmentation, we performed experiments on a downsampling of both training and testing instances.

Our results show that inference time augmentation (rows with "IA") leads to a performance gain of 1-2%, with a more substantial improvement observed for AG News and Yahoo Answers. This may be attributed to the fact that AG News has an average text length of only 38 words, and the Yahoo Answers dataset includes many answers with only one phrase. Inference time augmentation effectively enhances the quality of shorter text inputs.

A.4 Qualitative Examples for Conditionally Generated Examples on Pseudo-label

In table 6, we show generated examples of a sample text from the Agnews dataset. We generate 5 examples conditioned on each of the 4 labels, and cherry-pick one for each label in the table presentation. The example shows that the topic of a generated text is related to the label which is conditioned on, while pertains the original meaning. This opens a path to leverage the language understanding ability of LLM for data augmentation, especially during self-training.

ID	Self-train	Methods	AG News	DBpedia	Yahoo Answers	Amazon
		# unlabeled train	4k (3.4%)	11.2k (2%)	15k ($<$ 1%)	20k ($<$ 1%)
		# test	7.6k	28k	20k	20k
1	No	Sentence-enc	75.6	73.4	55.5	89.6
2	No	Sentence-enc + Inf-Aug	78.2	74.7	57.4	90.2
3	Yes	Self-train	83.3	96.3	62.5	91.1
4	Yes	Self-train + Inf-Aug	83.9	96.8	64.3	91.3
5	Yes	GenCo	89.2	98.4	68.6	95.3
6	Yes	GenCo + Inf-Aug	89.7	98.5	70.2	95.4

Table 5: Evaluation of inference time augmentation. "Inf-Aug" represents input augmentation added during inference.

Observed Text	Starbucks Corp’s president and chief executive, Orin Smith, said Tuesday he plans to retire early next year because he wants to slow down and focus on philanthropy, family and sports.
Politics	The announcement of Orin Smith’s retirement is likely to have political implications for Starbucks Corporation... His replacement will have big shoes to fill, and the decision could have long-term effects on the company’s policies and direction.
Sports	...Sports have always been a major part of Smith’s life, as he was a college athlete and later went on to become the CEO of Starbucks. It is clear that sports have had a major influence on his life and he wants to make time for them in his retirement.
Business	Starbucks Corp has seen great success under the leadership of Orin Smith, with the company’s stock price more than tripling since he became CEO in 2005. This success has allowed him to retire early and . . .
Technology	Orin Smith’s plan to retire early next year is an example of how technology has changed the way we work and live. By utilizing technology, Smith is able to take advantage of the increasingly popular trend of “work-life balance" ...

Table 6: Examples of generated text conditioned on pseudo labels in the left column.

B Proof of Theorems

Theorem 2. *Consider a binary classification problem with linearly separable labeled examples, when* $0 < \tau < 1$*, optimizing* $\mathcal{L}_{t2l} = -\sum_{i=1}^{N}\sum_{j=1}^{L} Q(\hat{y}_j|x_i)\log P(\hat{y}_j|x_i)$ *with gradient descend will enforce the larger margin between classes.*

Proof. We use dot product $\langle\cdot,\cdot\rangle$ as implementation of similarity function. Let the embedding of instance i be $\boldsymbol{x}_i = f_\theta(x_i)$ and the embedding of label prompt j be $\boldsymbol{e}_c = f_\theta(p_c), c \in \{1, 2\}$ for binary classification. Then,

$$P(\hat{y}_1|x_i;\theta) = \frac{\exp(\langle \boldsymbol{x}_i, \boldsymbol{e}_1\rangle)}{\exp(\langle \boldsymbol{x}_i, \boldsymbol{e}_1\rangle) + \exp(\langle \boldsymbol{x}_i, \boldsymbol{e}_2\rangle)} = \frac{1}{1 + \exp(-\langle \boldsymbol{x}_i, \boldsymbol{e}_1 - \boldsymbol{e}_2\rangle)} \tag{12}$$

$$P(\hat{y}_2|x_i;\theta) = 1 - P(\hat{y}_1|x_i;\theta) \tag{13}$$

Notation-wise, define $d_i = \langle \boldsymbol{x}_i, \boldsymbol{e}_1 - \boldsymbol{e}_2\rangle$, then

$$P(\hat{y}_1|x_i;\theta) = \frac{1}{1 + e^{-d_i}} \tag{14}$$

$$P(\hat{y}_2|x_i;\theta) = 1 - \frac{1}{1 + e^{-d_i}} \tag{15}$$

$$\tag{16}$$

In binary classification, the margin is simply

$$\text{margin} = \begin{cases} d_i & x_i \text{ is class 1} \\ -d_i & x_i \text{ is class 2} \end{cases}$$

For soft-label distribution Q,

$$Q(\hat{y}_1|x_i;\theta) = \frac{1}{1 + e^{-d_i/\tau}} \tag{17}$$

$$Q(\hat{y}_2|x_i;\theta) = 1 - \frac{1}{1 + e^{-d_i/\tau}} \tag{18}$$

$$\tag{19}$$

Then $\mathcal{L}_{t2l}$ is derived as

$$\mathcal{L}_{t2l} = \sum_{i=1}^{N} \log(1 + e^{-d_i}) + \frac{d_i e^{-d_i/\tau}}{1 + e^{-d_i/\tau}} \tag{20}$$

Calculate the derivative of $\mathcal{L}_{t2l}$ w.r.t d_i,

$$\frac{\partial \mathcal{L}_{t2l}}{\partial d_i} = \frac{-d_i e^{-d_i/\tau}}{\tau(e^{-d_i/\tau} + 1)^2} + \frac{e^{-d_i/\tau} - e^{-d_i}}{(e^{-d_i/\tau} + 1)(e^{-d_i} + 1)} \tag{21}$$

For the first part of equation 21, the sign depends on $-d_i$. For the second part, the sign depends on $e^{-d_i/\tau} - e^{-d_i}$. When $0 < \tau < 1$,

$$\begin{cases} e^{-d_i/\tau} - e^{-d_i} < 0 & \text{when } d_i > 0 \\ e^{-d_i/\tau} - e^{-d_i} > 0 & \text{when } d_i < 0 \end{cases}$$

Therefore,

$$\begin{cases} \frac{\partial \mathcal{L}_{t2l}}{\partial d_i} < 0 & \text{when } d_i > 0 \\ \frac{\partial \mathcal{L}_{t2l}}{\partial d_i} > 0 & \text{when } d_i < 0 \end{cases} \tag{22}$$

One step of gradient descend optimizes d by $d'_i = d_i - \eta\frac{\partial \mathcal{L}_{t2l}}{\partial d_i}$. From equation 22, we get the conclusion that $|d'_i| > |d_i|$. In other words, the margin becomes larger after optimization, which finishes the proof. □

Theorem 3. *Under the setting in Theorem 2, let m_i be the margin of instance i and consider the constraint $m_i \leq B$ for all i, the classifier converges to a max margin classifier, as the bound B goes to infinity.*

Proof. Using the definition from Theorem 2,

$$\mathcal{L}_{t2l} = \sum_{i=1}^{N} \log(1 + e^{-d_i}) + \frac{d_i e^{-d_i/\tau}}{1 + e^{-d_i/\tau}} \tag{23}$$

The margin m_i for instance i can be written as $m_i = \begin{cases} d_i & x_i \text{ is class 1} \\ -d_i & x_i \text{ is class 2} \end{cases}$.

The equation 23 can be written as

$$\mathcal{L}_{t2l} = \sum_{y_i=0} \log(1 + e^{-m_i}) + \frac{m_i e^{-m_i/\tau}}{1 + e^{-m_i/\tau}} + \sum_{y_j=1} \log(1 + e^{m_j}) - \frac{m_j e^{m_j/\tau}}{1 + e^{m_j/\tau}} \tag{24}$$

Let $m^* = \min(m_i)$ be the minimal margin, let N_1 and N_2 be the number of instances in class 1 and class 2 respectively which reaches the minimal margin. From the gradient analysis in equation 22, the examples with $m_i > m^*$ has loss lower bounded by that with minimal margin. Then

$$\begin{aligned} \mathcal{L}_{t2l} = N_1(\log(1 + e^{-m^*}) + \frac{m^* e^{-m^*/\tau}}{1 + e^{-m^*/\tau}}) + N_2(\log(1 + e^{m^*}) - \frac{m^* e^{m^*/\tau}}{1 + e^{m^*/\tau}}) \\ + O(\log(1 + e^{-m^*}) + \frac{m^* e^{-m^*/\tau}}{1 + e^{-m^*/\tau}}) + O(\log(1 + e^{m^*}) - \frac{m^* e^{m^*/\tau}}{1 + e^{m^*/\tau}}) \end{aligned} \tag{25}$$

When B approaches ∞, for N_1 part in equation 25,

$$\log(1 + e^{-m^*}) + \frac{m^* e^{-m^*/\tau}}{1 + e^{-m^*/\tau}} \sim e^{-m^*} + m^* e^{-m^*/\tau} \tag{26}$$

When $m \to B$, $\lim_{m \to B} e^{-m^*} \to 0$, and $\lim_{m \to B} m^* e^{-m^*/\tau} = \lim_{m \to B} \frac{1}{1/\tau e^{m^*/\tau}} = 0$ by L'Hopital's rule.

For N_2 part in equation 25,

$$\log(1 + e^{m^*}) - \frac{m^* e^{m^*/\tau}}{1 + e^{m^*/\tau}} \sim \log(1 + e^{m^*}) - m^* \tag{27}$$

When $m \to B$, $\lim_{m \to B} \log(1 + e^{m^*}) - m^* = \lim_{m \to B} \log(1 + \frac{1}{e^{m^*}}) = 0$.

Therefore, the loss is minimized when the minimal margin is maximized and thus the classifier converges to a max margin classifier when B goes to infinity. □