

# Supervised Graph Contrastive Pretraining for Text Classification\*

Samujwal Ghosh  
IIT Hyderabad  
samujwal86@gmail.com

Subhadeep Maji  
Amazon  
msubhade@amazon.com

Maunendra Sankar Desarkar  
IIT Hyderabad  
maunendra@cse.iith.ac.in

## ABSTRACT

Contrastive pretraining techniques for text classification has been largely studied in an unsupervised setting. However, oftentimes labeled data from related tasks which share label semantics with current task is available. We hypothesize that using this labeled data effectively can lead to better generalization on current task. In this paper, we propose a novel way to effectively utilize labeled data from related tasks with a graph based supervised contrastive learning approach. We formulate a token-graph by extrapolating the supervised information from examples to tokens. Our formulation results in an embedding space where tokens with high/low probability of belonging to same class are near/further-away from one another. We also develop detailed theoretical insights which serve as a motivation for our method. In our experiments with 13 datasets, we show our method outperforms pretraining schemes by 2.5% and also example-level contrastive learning based formulation by 1.8% on average. In addition, we show cross-domain effectiveness of our method in a zero-shot setting by 3.91% on average. Lastly, we also demonstrate our method can be used as a noisy teacher in a knowledge distillation setting to significantly improve performance of transformer based models in low labeled data regime by 4.57% on average.

## CCS CONCEPTS

• **Human-centered computing** → *Social network analysis*; • **Computing methodologies** → **Information extraction**; *Supervised learning*; **Neural networks**; **Learning latent representations**;

## KEYWORDS

Contrastive Learning, Graph Neural Network, Text Classification, Text Representation, Zero-shot Classification, Disaster Response, Sentiment Analysis

## ACM Reference Format:

Samujwal Ghosh, Subhadeep Maji, and Maunendra Sankar Desarkar. 2022. Supervised Graph Contrastive Pretraining for Text Classification. In *Proceedings of ACM SAC Conference (SAC'22)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

\*A condensed version of this paper has been accepted to ACM SAC'22.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
SAC'22, April 25 –April 29, 2022, Brno, Czech Republic  
© 2022 Association for Computing Machinery.  
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Lack of labeled data is probably the most pervasive problem limiting the effectiveness of modern deep learning methods. In text classification, the widespread approach to get around this has been the classic pretrain-finetune paradigm where a model is learnt on a large corpus and then fine-tuned using a domain specific dataset. This pretrain-finetune paradigm is able to perform significantly better than just finetuning the model. Pretraining is generally performed using a large amount of unlabeled data. There are many situations where labeled data from *related* datasets are available and can be utilized for pretraining. Our definition of *related* is fairly broad; we consider two datasets to be related if their label space has the same semantics e.g. tweets related v/s unrelated to a natural disaster, positive v/s negative sentiment in a review, etc.

Due to the recent success of contrastive learning in improving generalization and theoretical guarantees in both Vision [4, 12, 26] and NLP [8, 17, 27] domains, we explore supervised contrastive pretraining in the context of binary text classification. We propose a graph contrastive pretraining technique to effectively utilize labeled data from related datasets when attempting to build a robust classifier for the current dataset. It learns an embedding space for unique tokens in the vocabulary by constructing a token-graph where the collocation of tokens defines the neighborhood.

Existing contrastive learning approaches fall on either unsupervised pretraining [7, 27] or supervised fine-tuning [8, 9, 16]. Unsupervised contrastive pretraining based techniques do not utilize supervised information even when available in related datasets. On the other hand, supervised fine-tuning based approaches use contrastive loss as an additional objective only in addition to cross-entropy loss. We propose supervised contrastive pretraining scheme which utilizes supervised information from related datasets with contrastive formulation. We extrapolate the label information on examples to tokens as their conditional probability  $\Pr(\text{label} | \text{token})$  and use this label information in a supervised contrastive learning formulation to learn a token embedding space. We contrast between the highly confident tokens within a class with similar tokens from other class. Our contrastive formulation results in an embedding space where tokens belonging to different classes are well separated, a desirable characteristic of feature space in a classification setting. We refer our proposed technique as Graph Contrastive Pre-training for Text (GCPT). We also develop theoretical guarantees around effectiveness of GCPT by building upon [24].

We conduct several experiments across 5 different sentiment classification and 8 disaster management datasets in binary classification setting to gain a robust understanding of our method's effectiveness. We find that our method outperforms word embedding techniques trained on unlabeled domain-specific datasets (e.g. disaster related tweets) by 2.5% on average. We also show that

our method is an effective pretraining scheme by combining it with several different baseline models from disaster and sentiment classification domain; in each case it improves the performance of underlying model, on average, by 1.87%. We also compare against example-level contrastive learning formulation and show our token level contrastive formulation is better by 1.8% on average. Our method also achieves robust zero-shot performance wherein no labeled data from current dataset is available (a very realistic scenario in disaster management). Once again it outperforms several baselines by 3.91% on average. Finally, we also improve upon BERT in a knowledge distillation setup wherein our model works as a noisy teacher for a BERT student. In this case, we improve upon BERT’s performance by 4.57% on average.

## 2 RELATED WORK

Most existing works attempting to utilize contrastive learning for text classification can be categorized into two paradigms a) unsupervised pretraining or b) supervised fine-tuning. Our work attempts to combine both approaches and is motivated by recent success of contrastive pretraining for language modelling and sentence representations leading to good generalizability across wide range of applications [8, 9, 17, 27].

**Unsupervised Pretraining:** Pretraining based approaches has been largely explored in an unsupervised setting only. However, oftentimes supervised corpora is available for other *related* datasets, e.g. labeled sentiment analysis dataset for “Book Reviews” when attempting to learn a sentiment analysis classifier for “Kitchen Items”. [6, 7] propose self-supervised contrastive learning methods for language models using data augmentation techniques such as back-translation and sampling spans from same or different documents as positives and negatives respectively. [27] extends the idea to learn sentence level representations by proposing a number of different sentence level augmentation strategies. However, unsupervised or self-supervised contrastive learning based methods have a natural difficulty when it comes to NLP, this is because unlike images changing a few words or even their ordering in a sentence can significantly alter the semantics.

**Supervised Fine-tuning:** On the other hand, [8, 9, 16, 17] takes a different approach by using the labeled information to learn a feature space where examples belonging to different classes are further away than examples from same class. They achieve this by typically adding a contrastive loss component to the standard cross-entropy loss. These methods have shown to be robust to label noise and significantly improve upon large language models such as BERT.

Our work combines above mentioned approaches by utilizing *supervised* information during *pretraining* using contrastive learning. However, there are two key differences in our formulation. First, we attempt to learn a token-level (e.g. word) embedding space while most existing methods are example-level (e.g. sentences). In our experiments, we show our token level formulation is better in comparison to an example level contrastive learning formulation. Second, our method is naturally suited to a cross-domain setting because of our token level pretraining setup. This allows us to use labeled data from related datasets effectively, while existing supervised contrastive learning methods are mostly restricted to

fine-tuning the contrastive loss jointly with cross-entropy loss on the current labeled dataset alone. To establish cross-domain applicability of our method, we show zero-shot text classification performance i.e no labeled data being available for the current dataset.

## 3 APPROACH

At a very high level, our approach is a pretraining technique for learning a token embedding space with the end objective of improving text classification performance on datasets with limited labeled data. We assume a shared label space between the previous and current datasets. This is reasonable assumption as the type of information to collect generally stays consistent across disasters. Using labeled pretraining corpora, from previous datasets, in the form of  $\{(e, y)\}$  where  $e$  is an example, and  $y$  is its class label, our method contrasts between representative candidate tokens from each class. Candidates are found by extrapolating the label information available at example-level to tokens as the conditional probability  $\Pr(\text{label} \mid \text{token})$ . This conditional label information is then used in a contrastive formulation to learn an embedding space where tokens from different classes are well separated. We also present novel theoretical results building upon [24], the theoretical analysis serves as motivation for our algorithm.

We refer to the current dataset as  $\mathbf{O}$  and the union of related datasets as  $\mathbf{P}$ . The elements in  $\mathbf{O}$  (and  $\mathbf{P}$ ) are pairs  $(x, y)$  where  $x$  is an example (e.g tweet) and  $y \in \mathcal{Y}$  is the class label,  $|\mathcal{Y}| = C$  is the number of classes. We assume label space  $\mathcal{Y}$  to be the same across current and related datasets. Also,  $|\mathbf{O}|$  is much smaller than  $|\mathbf{P}|$ , indicative of our limited labeled data setup.

### 3.1 Token Label Conditional

To extrapolate the label information from examples to tokens, we introduce a token graph  $\mathbf{G} = (V, E)$ . Specifically, set of all unique tokens with minimum frequency ( $\geq 5$ ) from  $\mathbf{p} \in \mathbf{P}$  are vertices ( $V$ ) in the graph. Edges ( $E$ ) are defined by pointwise mutual information (PMI) between two tokens over all  $\mathbf{p} \in \mathbf{P}$ . Given a label  $c$ , let  $\mathbf{P}^c \subset V$  be a set of tokens defined with,

$$\theta_x^j = \frac{n^j(x)}{\sum_{c=1}^C n^c(x)}; \text{ s.t. } \theta_x^c \geq \tau \geq \frac{1}{C}; c = \arg \max_{j \in \{1, \dots, C\}} \theta_x^j \quad (1)$$

where  $n^c(x)$  is the frequency of token  $x$  in examples  $\mathbf{p} \in \mathbf{P}$  such that their label is  $c$ . Intuitively,  $x \in \mathbf{P}^c$  has the highest conditional probability of belonging to label  $c$  (ties broken arbitrarily) amongst all labels. The normalized frequency  $\theta_x^c$  determines the confidence of label-token association, and only tokens with high confidence (i.e  $\geq \tau$ ) have a conditional label. Considering high confidence within a class ensures exclusion of insignificant tokens (e.g. stopwords) as candidates as they occurs uniformly across classes. Note that these tokens may be present in the graph but not considered as candidate. On the other hand, to avoid very less frequent tokens as candidates, we only consider tokens with a minimum frequency when constructing the graph  $\mathbf{G}$ . Using sets  $\mathbf{P}^c$  we define the induced subgraphs for individual labels  $c \in \mathcal{Y}$  as  $\mathbf{G}^c = (\mathbf{P}^c, E^c, W)$ . Note that sets  $\mathbf{P}^c$  are mutually exclusive, and not all tokens  $x \in V$  belong to  $\mathbf{P}^c$  (there can be only one  $c$  for a particular token  $x$ ). To learn a

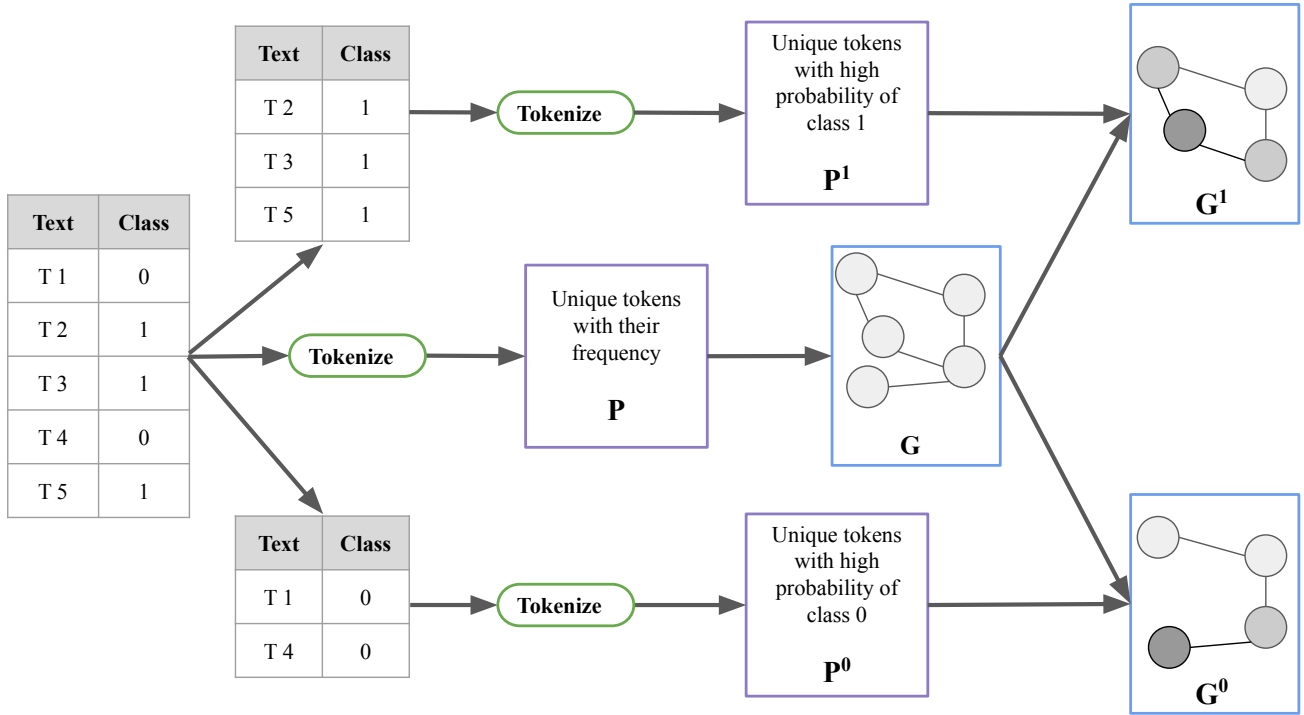


Figure 1: High level illustration of our token-level conditional and graph construction in binary setting.

token (node) representation for  $x \in V$  we apply a 2-layer GCN [15] on  $G$ . Specifically,

$$H^{(l+1)} = \sigma \left( D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (2)$$

with  $H^{(0)} = \text{Glove}(\cdot)$ . Here,  $D$  is the diagonal degree matrix of  $G$  and  $\hat{A} = A + I$ , where  $A$  is the weighted adjacency matrix of  $G$  and  $l$  is the layer with values  $l \in \{0, 1\}$  in this case. GCN based node representations capture co-occurrence patterns between tokens from  $P$  because of the underlying message-passing over neighborhoods. In addition, because  $G$  can be seen as a union of induced subgraphs  $G^c$ , the message-passing with GCN layers over  $G$  smoothens the node representations across these subgraphs and acts as a form of regularization to our contrastive formulation. This regularization is helpful because our extrapolation of label information to tokens is based on observed  $P$  and therefore empirical.

### 3.2 Graph Contrastive Pretraining

Let  $\cup_{c=1}^C G^c$  be the union of the induced subgraphs for individual labels  $c \in Y$ . For node (token)  $n \in G^c$  ensuring  $n$  is closer to its neighbors from  $G^c$  and further away from nodes in  $G^{-c}$  would result in tokens belonging to the same (or different) class to be close (or faraway) from one other in the embedding space. To this effect, we propose a graph contrastive learning method over nodes in  $\cup_{c=1}^C G^c$ . Specifically, for a candidate token  $n \in G^c$ , we sample its positive and negative neighbors as follows:

- **Positive:** The positives are sampled from neighborhood of node  $n$  in the induced subgraph  $G^c$ . We draw  $S$  positive samples with replacement from this neighborhood ( $\text{Nbh}$ ) i.e.  $M^+ \sim_{iid} \text{Nbh}(n)$ , with  $|M^+| = S$ .
- **Negative:** The negatives are sampled uniformly at random from induced subgraphs other than  $G^c$  i.e.  $G^{-c}$ . We draw  $S$  negatives samples with replacement from  $G^{-c}$  i.e.  $M^- \sim_{iid} \mathcal{U}(P^{-c})$ , with  $|M^-| = S$ .  $\mathcal{U}$  represents uniform distribution.

Figure 1 provides an illustration of our sampling process on  $G^c$  and  $G^{-c}$ . Using candidate token  $n$ , the positive samples  $M^+ = m_{1..s}^+$  and negative samples  $M^- = m_{1..s}^-$  we define a contrastive loss formulation with equation (3). The proposed loss is a variant of [13] where positive samples are drawn from graph neighborhoods. Formally, the loss on node  $n$  is defined as  $l_n$  as follows,

$$f(m, n) = \exp \left( \frac{z_m \cdot z_n}{\gamma} \right) \quad (3)$$

$$l_n = -\frac{1}{S} \sum_{m^+ \in M^+} \log \left( \frac{f(m^+, n)}{\sum_{m \in M^+ \cup M^-} f(m, n)} \right)$$

Where  $z_n = H^{(2)}(n)$  is the GCN representation of node  $n$  and  $\gamma$  is a temperature parameter for softmax. Intuitively, the loss tries to bring the representation of nodes from  $G^c$  close to each other in comparison to nodes from  $G^{-c}$ . This formulation results in greater feature separation of different classes in the embedding space. The overall loss is obtained by drawing a random sample of  $N$  such candidate tokens by first sampling a class and then sampling a node

from the set of nodes belonging to that class.

$$\mathcal{L} = \frac{1}{N} \left( \sum_{c \sim \mathcal{U}(1..C)} \sum_{n \sim \mathcal{U}(P^c)} l_n \right) \quad (4)$$

While equation (4) is defined only on nodes in  $\cup_c G^c$ , representation for nodes in  $G \setminus \cup_c G^c$  also get modified during pretraining because GCN layers are trained end-to-end with the contrastive loss  $\mathcal{L}$ . This results in better overall separation in the embedding space for all tokens in the vocabulary.

The success of the unsupervised contrastive learning methods rests on the reliability of positive (and negative) samples belonging to the same (and different) class as the candidate. This is a difficulty with many existing unsupervised contrastive learning methods [24]. Although our contrastive formulation is not unsupervised, it relies on extrapolation of example level supervision to tokens and potentially suffers from the label noise. We address this issue by applying our contrastive formulation to tokens with high confidence label association (i.e.  $\cup_{c=1}^C G^c$ ), resulting in better guarantees on positive (and negative) samples belonging to the same (and different) class as the candidate. In § 3.3, we build upon the theoretical analysis in [24] to develop deeper insights on token level extrapolation of supervision for contrastive losses.

### 3.3 Theoretical Insights

Our theoretical results are for binary classification problems, i.e.  $C = 2$ , also we restrict ourselves to only single positive and negative sample for each candidate  $x$  (i.e.  $S = 1$ ). The extension of these results for multi-class problems with blocks of  $S$  positive and negative samples is left as future work. In this setting, the contrastive loss in equation (3) is,

$$\begin{aligned} l_n &= -\log \left( \frac{\exp(\frac{z_{m^+} \cdot z_n}{\gamma})}{\exp(\frac{z_{m^+} \cdot z_n}{\gamma}) + \exp(\frac{z_{m^-} \cdot z_n}{\gamma})} \right) \\ &= \log \left( 1 + \exp \left( \frac{z_n \cdot (z_{m^-} - z_{m^+})}{\gamma} \right) \right) \\ &:= h(z_n \cdot (z_{m^-} - z_{m^+})) \end{aligned}$$

For ease of exposition, with some abuse of notation we will use  $h(z_n \cdot (z_{m^-} - z_{m^+}))$  and  $h(z_n, z_{m^+}, z_{m^-})$  interchangeably.

Our theoretical analysis is not restricted to the exact sampling scheme discussed in § 3.2 and is more broadly applicable. The only assumption is the label conditional on every token  $x$  is modeled as Bernoulli distribution parameterized by  $\theta_x$ . Note that, we do not make further assumption on estimation of  $\theta_x$  (such as equation (1)).

We now describe a generative process from which our contrastive formulation follows. First sample a token  $x$  from data distribution  $p_{\text{data}}$ , then sample positive and negative classes  $c^+ \sim \text{Ber}(\theta_x)$ ,  $c^- \sim \text{Ber}(1 - \theta_x)$  according to Bernoulli distributions dependent on the conditional probability for token  $x$ . Then sample positive and negative examples for contrastive learning w.r.t token  $x$  from distribution  $D_{c^+}$  and  $D_{c^-}$ . Note that the sampling scheme discussed in § 3.2 is a special case of the above because we don't sample  $c^+$ ,  $c^-$  explicitly. Instead, every token already has the most likely label associated with it (in accordance with equation (1)). The

overall contrastive loss can then be expressed as,

$$L(z) = \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \text{Ber}(\theta_x) \\ c^- \sim \text{Ber}(1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [h(z_x, z_{m^+}, z_{m^-})] \right] \quad (5)$$

In equation (5),  $D_{c^+}, D_{c^-}$  are data distributions of classes  $c^+, c^-$  respectively. For ease of exposition, we will drop the  $\text{Ber}(\cdot)$  notation and instead use  $c^+ \sim \theta_x, c^- \sim (1 - \theta_x)$ . Following [24], we split  $L(z)$  into two components based on  $c^+ = c^-$ , the ‘‘class-collision loss’’ (CCL) and ‘‘ideal contrastive loss’’ (ICL).

$$\begin{aligned} L^{\text{CCL}}(z) &= \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [h(z_x, z_{m^+}, z_{m^-}) \mid c^+ = c^-] \right] \\ L^{\text{ICL}}(z) &= \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [h(z_x, z_{m^+}, z_{m^-}) \mid c^+ \neq c^-] \right] \end{aligned}$$

The ideal behaviour of a learning algorithm minimizing contrastive loss  $L(z)$  should be to minimize both CCL ( $L^{\text{CCL}}(z)$ ) as well as ICL ( $L^{\text{ICL}}(z)$ ) loss components. However, the behaviour of the two loss components is very different from one another. The CCL component comes from the penalty a learning algorithm pays from class specific positive and negative sampling errors while ICL component helps in learning contrastive feature representations  $z$ . The following result allows us to characterize the relation of label-token association parameters;  $\theta_x$  with CCL and ICL components of  $L(z)$ .

**THEOREM 1.** *The contrastive loss  $L(z)$  is upper bounded by  $cL^{\text{CCL}}(z) + L^{\text{ICL}}(z)$ , where  $c = 2\tau(1 - \tau) \in [0.5, 1]$  and  $\tau \geq \frac{1}{2}$ .*

PROOF. For ease of exposition, we will denote  $h(z_x, z_{m^+}, z_{m^-})$  as  $g$ . We first separate  $L^\sharp(z)$  from  $L(z)$  as follows,

$$\begin{aligned}
L(z) &= \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g] \right] \\
&\stackrel{\text{(a)}}{=} \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\mathbb{1}(c^+ = c^-)} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g \mid \mathbb{1}(c^+ = c^-)] \right] \right] \\
&\stackrel{\text{(b)}}{=} \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g | c^+ = c^-] \Pr(\mathbb{1}(c^+ = c^-)) \right. \\
&\quad \left. + \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g | c^+ \neq c^-] \Pr(\mathbb{1}(c^+ \neq c^-)) \right] \\
&\stackrel{\text{(c)}}{=} \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g | c^+ = c^-] (2\theta_x(1-\theta_x)) \right. \\
&\quad \left. + \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g | c^+ \neq c^-] (1-2\theta_x(1-\theta_x)) \right] \\
&\leq \stackrel{\text{(d)}}{\mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g | c^+ = c^-] 2\theta_x(1-\theta_x) \right. \\
&\quad \left. + \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g | c^+ \neq c^-] \right]} \\
&\stackrel{\text{(e)}}{=} \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g \mid c^+ = c^-] 2\theta_x(1-\theta_x) \right] + L^{\text{ICL}}(z)
\end{aligned}$$

Where, step (a) follows from Iterated expectation w.r.t  $\mathbb{1}(c^+ = c^-)$ . Step (b) follows from opening up the expectation w.r.t  $\mathbb{1}(c^+ = c^-)$ . Step (c) follows from property of Bernoulli distribution, i.e  $\Pr(\mathbb{1}(c^+ = c^-)) = 2\theta_x(1-\theta_x)$ . Step (d) follows from non-negativity of  $2\theta_x(1-\theta_x)$ .

Next, we connect true parameters  $\theta_x$  with the estimates from equation (1), which allows us to get an upper bound on the first term of (e). First we note that equation (1) is a MLE of Bernoulli parameter  $\theta_x$ . From a classical result on consistency of MLEs, we note that for any continuous function  $f$  of  $\theta_x$ ;  $f(\hat{\theta}_x)$  is a consistent estimator of  $f(\theta_x)$ . Formally,

$$\lim_{n \rightarrow \infty} P_{\theta_x}(|f(\hat{\theta}_x) - f(\theta_x)| \geq \epsilon) = 0 \quad (6)$$

(see [3] for a proof). Noting that  $f(\theta_x) = 2\theta_x(1-\theta_x)$  is a continuous function of  $\theta_x$ , we apply result (6) to  $f(\theta_x)$  and replace  $f(\theta_x)$  with its estimate  $f(\hat{\theta}_x)$  in step (e) above,

$$\begin{aligned}
L(z) &\leq \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [g \mid c^+ = c^-] 2\hat{\theta}_x(1-\hat{\theta}_x) \right] + L^{\text{ICL}}(z) \\
&\leq 2\tau(1-\tau)L^{\text{CCL}}(z) + L^{\text{ICL}}(z)
\end{aligned}$$

We note that  $2\hat{\theta}_x(1-\hat{\theta}_x)$  is a monotonically decreasing function of  $\hat{\theta}_x$  in domain  $[0.5, 1]$  (we chose  $\hat{\theta}_x \geq 0.5$  for all  $x$  from equation (1)). This implies there exists  $\tau \geq 0.5$  such that  $2\hat{\theta}_x(1-\hat{\theta}_x) \leq 2\tau(1-\tau)$  for all  $x$ , from which the claim follows immediately.  $\square$

As an immediate consequence of Theorem 1, by setting a  $\tau > 0.5$  our sampling algorithm can be conceptualized as minimizing the above upper bound on  $L(z)$  w.r.t  $\tau$  by down weighing the CCL component. Of course, if we set  $\tau = 1$ , we can nullify the entire CCL component. However this comes at the cost of having very few candidates  $x$ , which satisfy  $\theta_x = 1$  in any practical setting. Thus severely limiting the size of the training data available for minimizing the empirical contrastive loss in equation (4).

Theorem 1 although providing an upper bound on  $L(z)$  presents only a partial picture because the upper bound can be loose and minimizing it w.r.t  $\tau$  may not explain the effects on minimization of  $L(z)$ . Therefore, our next step is to prove the existence of a suitable  $\tau$  dependent lower bound on  $L(z)$ .

**THEOREM 2.** *The contrastive loss  $L(z)$  is lower bounded by*

$$(1-2\tau(1-\tau)) \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} [h(z_x \cdot (\mu_{c_x^-} - \mu_{c_x^+})) \mid c^+ \neq c^-]$$

where  $\mu_{c_x^+} = \mathbb{E}_{p \in D_{c^+}} [p]$ ,  $\mu_{c_x^-} = \mathbb{E}_{p \in D_{c^-}} [p]$  are the mean of the two classes  $c^+$ ,  $c^-$ . Where  $2\tau(1-\tau) \in [0.5, 1]$  and  $\tau \geq \frac{1}{2}$ .

PROOF. Using the definition of  $L(z)$  and following steps (a)-(c) in proof of Theorem 1 we arrive at,

$$\begin{aligned}
L(z) &= \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [h(z_x \cdot (m^- - m^+)) \mid c^+ = c^-] (2\theta_x(1-\theta_x)) \right. \\
&\quad \left. + \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [h(z_x \cdot (m^- - m^+)) \mid c^+ \neq c^-] (1-2\theta_x(1-\theta_x)) \right] \\
&\geq \stackrel{\text{(b)}}{\mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ \mathbb{E}_{\substack{m^+ \sim D_{c^+} \\ m^- \sim D_{c^-}}} [h(z_x \cdot (m^- - m^+)) \mid c^+ \neq c^-] (1-2\theta_x(1-\theta_x)) \right]} \\
&\geq \stackrel{\text{(c)}}{\mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} [(1-2\theta_x(1-\theta_x))h(z_x \cdot (\mu_{c_x^-} - \mu_{c_x^+})) \mid c^+ \neq c^-]}
\end{aligned}$$

where step (b) follows from non-negativity of expectation of loss functions and the fact that  $2\theta_x(1-\theta_x) \geq 0$ . Step (c) follows from convexity of log-sum-exp function  $h$  (w.r.t  $z$ ) and Jensen's inequality applied to inner expectation and  $h$ . Once again as in proof of Theorem 1 using consistency of MLE (equation (6)), we replace  $(1-2\theta_x(1-\theta_x))$  with its estimated value in step (c) above.

$$\begin{aligned}
L(z) &\geq \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} \left[ (1-2\hat{\theta}_x(1-\hat{\theta}_x))h(z_x \cdot (\mu_{c_x^-} - \mu_{c_x^+})) \mid c^+ \neq c^- \right] \\
&\geq (1-2\tau(1-\tau)) \mathbb{E}_{\substack{x \sim p_{\text{data}} \\ c^+ \sim \theta_x \\ c^- \sim (1-\theta_x)}} [h(z_x \cdot (\mu_{c_x^-} - \mu_{c_x^+})) \mid c^+ \neq c^-]
\end{aligned}$$

Following the proof of Theorem 1, we note that  $2\hat{\theta}_x(1-\hat{\theta}_x) \leq 2\tau(1-\tau)$  for all  $x$ . This implies  $1-2\hat{\theta}_x(1-\hat{\theta}_x) \geq 1-2\tau(1-\tau)$  for all  $x$ , from which the claim follows immediately.  $\square$

Therefore, attempting to maximize the lower bound in Theorem 2 w.r.t  $\tau$  is equivalent to minimizing  $2\tau(1-\tau)$  w.r.t  $\tau$ . Thus, combining Theorem 1 and Theorem 2 we note that by choosing an appropriate value of  $\tau > 0.5$  our sampling algorithm minimizes an upper bound

on  $L(z)$  (§ Theorem 1) as well as maximizes a lower bound on it (§ Theorem 2).

Once again, its important to stress that *minimization* of  $2\tau(1-\tau)$  in context of the analysis does not mean choosing its minima w.r.t  $\tau$  (which would be  $\tau = 1$ ). Instead, the analysis simply indicates that by choosing a  $\tau > 0.5$  our sampling mechanism results in a trade-off between CCL and ICL components of the contrastive loss. The correct choice of  $\tau$  is still application dependent. Choosing a  $\tau \approx 0.5$  results in more training data at the risk of more class-collision. On the other hand, choosing  $\tau \rightarrow 1$  incurs little CCL loss at the cost of smaller training datasets for minimizing ICL and possibly hurts performance due to poor representations learnt. Thus the analysis illustrates, by choosing an appropriate threshold on label conditional probability ( $\tau$ ) our method (in § 3.1 and § 3.2) trades-off between components of contrastive loss.

## 4 EXPERIMENTAL SETUP

We perform experiments on two collections of total 13 datasets from disaster and sentiment domains. For each collection, we consider one dataset as the current and rest as related datasets and apply our pretraining technique on related datasets only. Training set of the current dataset is used only to train classifier models and not for contrastive pretraining. A 2-layer BiLSTM model is used as classifier for majority of our experiments as BiLSTM model is shown to have near state-of-the-art performance across the two domains on which we report our experiments. However, our pretraining method is generic and not tied to a particular choice of classification model. To illustrate this in our experiments we show that our pretraining mechanism improves the performance of a number of SOTA models in different application domains.

To ease the explanation of various scenarios, we follow the notations as per Table 1. We employ 2 phase setup for our experiments.

Notation	Meaning
$\mathbf{P}$	Union of labeled related datasets
$\mathbf{O}_{\text{TR}}$	Train set of current dataset
$\mathbf{O}_{\text{TE}}$	Test set of current dataset
$f_{\text{CE}}(\cdot)$	Training with cross-entropy loss
$h_{\text{CL}}^{\text{S}}(\cdot)$	Pretraining with contrastive loss over sentences
$h_{\text{CL}}^{\text{T}}(\cdot)$	Pretraining with contrastive loss over tokens

Table 1: Notations

In the first phase, we apply a GCN with contrastive loss over the token-graph to learn vectors for each token ( $h_{\text{CL}}^{\text{T}}(\cdot)$ ). Then, in the second phase, we train the classifier with Cross-Entropy (CE) loss ( $f_{\text{CE}}(\cdot)$ ) where example tokens are represented by the pretrained token embeddings obtained in the first phase. We report Weighted F1 scores averaged over 3 runs on  $\mathbf{O}_{\text{TE}}$  with different seeds for all our experiments. We present and analyze results, aggregated across all datasets in disaster and sentiment collections, of various experiments in context of several research questions in § 5. Next, we discuss the datasets used in our experiments.

## 4.1 Datasets

Our review sentiment collection has 5 datasets containing reviews and the disaster collection has 8 datasets containing tweets. The sentiment datasets are from [2] with binary annotation signifying positive or negative sentiment from 5 domains (books, dvd, electronics, kitchen, video) each containing 3000 reviews. Following [16], we use 400 random examples as test and split the remaining examples into 80% : 20% ratio as train (4480) and validation (1120) sets. The disaster datasets are collected from 3 sources. We use the 6 datasets “2013 Alberta Flood” (AF), “2013 Boston Bombing” (BB), “Oklahoma Tornado Season 2013” (OT), “Queensland Floods 2013” (QF) and “Sandy Hurricane 2012” (SH) and “2013 West Texas Explosion” (WE) from CrisisLexT6 [21]. Additionally, also use “2015 Nepal Earthquake” (NE) dataset from [1] and Kaggle’s “Natural Language Processing with Disaster Tweets” (KL) competition data<sup>2</sup> [11]. Only the “train.csv” portion of the KL dataset is used for our experiments. All disaster datasets are tagged with binary relevance information as either *relevant* (Class 1) or *not relevant* (Class 0). We divide each dataset into train (60%), validation (10%) and test (30%) sets. Table 2 shows more details about each dataset. NE and KL datasets have special significance to our experiments. As the NE dataset falls outside of CrisisLexT6, it works as a heterogeneous dataset to verify the effectiveness of our technique. Additionally, the KL dataset contains examples from multiple disasters including storm, wildfire, flood, etc unlike the rest. We believe, incorporation of these two datasets works as an additional evidence on the effectiveness of our approach in different scenarios.

Dataset	1	0	Train	Val	Test
Alberta Flood (AF)	5189	4842	6019	1002	3008
Boston Bombing (BB)	5648	4364	6007	1001	3004
Kaggle (KL)	3356	4336	4622	816	2164
Nepal Earthquake (NE)	5527	6141	7000	1166	3502
Oklahoma Tornado (OT)	4827	5165	5994	998	2997
Queensland Floods (QF)	5414	4619	6019	1003	3011
Sandy Hurricane (SH)	6138	3870	6005	1001	3002
West Texas Explosion (WE)	5246	4760	6001	1001	3004

Table 2: Details of 8 disaster datasets used in our experiments. Classes 1/0 indicates tweets relevant/irrelevant to the disaster.

## 5 RESULTS

We now discuss the experimental results with reference to several important research questions to understand the usefulness of the proposed pretraining approach that we refer to as Graph Contrastive pretraining for Text (GCPT). We run our experiments on each dataset and report the scores averaged over all datasets in a collection. Refer to Appendix §A for more detailed dataset-wise results.

**Q1) Does GCPT outperform unsupervised pretrained embedding baselines?**

<sup>2</sup><https://www.kaggle.com/c/nlp-getting-started/overview>

GCPT uses supervised data and propagates that information to individual tokens by graph contrastive learning. In this experiment, we want to see if our technique is useful over unsupervised pretraining, as done in GloVe, Word2Vec, etc. To that effect, we perform text classification on current data using a classifier with CE ( $f_{CE}(O_{TR})$ ) initialized with different word embeddings to see which initialization leads to better performance. Specifically,

- (1) *Word embeddings pretrained on large corpus*: We use GloVe [22] embedding pretrained on large general purpose corpora. In addition, we also use a Word2Vec model pretrained on a large corpus of disaster-related data with 9.4 million tokens [20] denoted as CNLP. As CNLP is trained on disaster data only, we do not perform experiments on sentiment datasets using CNLP.
- (2) *Word embeddings trained on available corpus*: We train a Word2Vec [19] model on all data available from related datasets. We denote this embedding set as W2V.

Table 3 present our findings for these experiments on both disaster and sentiment datasets. GCPT outperforms all unsupervised pretrained embeddings due to its capability to capture and transfer supervised information effectively. We are able to achieve an absolute improvement of 3.3% for disaster and 1.7% for sentiment datasets. We also observe that GloVe outperforms both CNLP and W2V embeddings for majority of the datasets. For this reason, we use GloVe for the following experiments.

Embedding	Disaster	Sentiment
GloVe	76.52	56.11
W2V	76.28	55.23
CNLP	76.73	-
GCPT	<b>80.03</b>	<b>57.81</b>

**Table 3: Performance comparison of GCPT with unsupervised pretrained embeddings aggregated over disaster and sentiment datasets.**

## Q2) Is GCPT suitable for different types of neural networks?

We verify the applicability of GCPT with different types of neural networks by replacing the BiLSTM classifier with other neural architectures such as MLP and CNN. For MLP, we represent an input example as the average of the embeddings of its tokens, and denote the method as MLP-BoW. We also use DenseCNN [14] and XML-CNN [18] following [23] for disaster and DPCNN [10] for sentiment datasets. The results are presented in Table 4. Our experiments show that GCPT performs well for almost all types of neural networks except MLP-BoW for disaster data, highlighting the benefit brought in by the refinement in the embedding space through the contrastive learning. In case of MLP-BoW, averaging over contrasted tokens within an example makes it difficult to capture proper context resulting in lower performance.

## Q3) Is token-level contrastive formulation effective?

This experiment verifies the effectiveness of GCPT over other straw-man approaches of incorporating the supervised information by replacing the GCPT pretraining phase (i.e. replacing  $h_{CL}^T(P)$ ). We consider two approaches that utilize supervised information present in related data:

Model	Disaster		Sentiment	
	CNLP	GCPT	GloVe	GCPT
MLP-BoW	53.74	50.95	52.94	<b>55.48</b>
CNN	77.30	<b>78.17</b>	54.89	<b>56.85</b>
DenseCNN	78.81	<b>79.99</b>	-	-
XML-CNN	71.74	<b>73.90</b>	-	-
DPCNN	-	-	55.36	<b>57.38</b>

**Table 4: GloVe vs GCPT performance with various neural networks.**

- (1) *Example-level Contrastive Learning (ECL)*: In this intuitive approach, we employ contrastive learning over *examples* instead of tokens. We pretrain a model with contrastive loss over examples from related datasets (i.e.  $h_{CL}^S(P)$  instead of  $h_{CL}^T(P)$ ) and finally re-train the classifier using the current dataset to fine-tune ( $f_{CE}(O_{TR})$ ). We consider each example in the training set as a candidate item for pretraining. For each candidate, we randomly select 5 examples from the same class as positive for that instance and 5 examples from other classes as negative examples to contrast. Trained model weights and token embeddings are transferred from pretraining to fine-tuning dataset.
- (2) *Related Dataset Fine-tuning (RDF)*: Here, we treat all labeled data (both related and current dataset) as training data. We directly fine-tune a model using related datasets with CE loss instead of any fine-tuning (i.e.  $f_{CE}(P)$  instead of  $h_{CL}^T(P)$ ), followed by fine-tuning the model again with the current dataset ( $f_{CE}(O_{TR})$ ). We compare with this baseline by introducing  $h_{CL}^T(P)$  to the front of the pipeline and replacing the embeddings in subsequent phases.

Table 5 presents the results for ECL (example-level) compared with GCPT (token-level). Our findings show that GCPT outperforms ECL by 1.91% and 1.69% for disaster and sentiment datasets on average. However, ECL performs better for the KL dataset. As KL contains data from multiple disasters, token-level contrast might reduce separability among various disasters, confusing the model. From a practical standpoint, it is generally unexpected that data will be coming from multiple domains during an ongoing disaster.

Recall that we achieved 80.03% & 57.81% (Ref. Table 3) for disaster and sentiment datasets when only current dataset data is utilized. Utilizing all available data to train the model (RDF setting), improves performance by 5.9% & 0.46% on average for disaster and sentiment datasets as shown in Table 6. However, when we incorporate GCPT into the RDF setting, by replacing the token embeddings, we achieve even higher performance improvement of 7.81% & 2.76% for disaster and sentiment datasets respectively. We also perform the same experiment with other models (MLP-BoW, CNN, DenseCNN and XML-CNN) and found incorporating GCPT helps achieve better performance in 6 out of 7 scenarios as shown in Table 7.

## Q4) What if no labeled data is available for the current dataset?

In domains like disaster, it is expected that no labeled data is available for an ongoing disaster. Data labelling during an ongoing disaster is very costly. In such a scenario, we can deploy models

Dataset	ECL	GCPT	Dataset	ECL	GCPT
AF	78.90	<b>80.23</b>	books	56.45	<b>58.48</b>
BB	66.64	<b>71.35</b>	dvd	52.72	<b>53.62</b>
KL	<b>56.45</b>	55.44	electronics	61.77	<b>64.66</b>
NE	52.46	<b>53.76</b>	kitchen	64.98	<b>65.33</b>
OT	65.14	<b>66.24</b>	video	58.63	<b>60.92</b>
QF	69.98	<b>71.66</b>	Avg	58.91	<b>60.60</b>
SH	61.64	<b>63.86</b>			
WE	70.19	<b>74.14</b>			
Avg	65.17	<b>67.08</b>			

Table 5: Performance comparison of ECL (Ref. 1) & GCPT for disaster (left) sentiment (right) datasets.

Embedding	Disaster	Sentiment
GloVe	<u>82.62</u>	56.47
W2V	82.01	55.79
CNLP	82.60	-
RDF (GCPT)	<b>84.32</b>	<b>58.43</b>

Table 6: Performance comparison of GCPT with unsupervised pretrained embeddings in RDF scenario aggregated over disaster and sentiment datasets.

Model	Disaster		Sentiment	
	GloVe	GCPT	GloVe	GCPT
MLP-BoW	65.98	<b>70.02</b>	<b>50.46</b>	50.22
CNN	81.94	<b>83.25</b>	51.84	<b>53.37</b>
DenseCNN	83.59	<b>84.00</b>	-	-
XML-CNN	83.27	<b>84.02</b>	-	-
DPCNN	-	-	58.36	<b>62.60</b>

Table 7: GloVe and GCPT performance with various neural models in RDF scenario.

trained on labeled data from related disasters only (i.e.  $f_C(\mathbf{P})$ ). We experiment with such a scenario by learning token representations and refer to this scenario as Zero-Shot setting.

Table 8 reports the performance among different pretraining techniques. We observe that GCPT outperforms other baselines for all datasets except NE. This is interesting, as NE contains tweets from earthquake which is different from rest of the disaster types. However, We observe CNLP performs better than GCPT as it is trained on a large corpora of disaster data which includes earthquake disaster. Table 9 presents performance comparison of W2V and GCPT for different models. We use W2V for this experiment, as it performs best (Ref. Table 8) in Zero-Shot scenario. We find a similar observation that GCPT outperforms W2V in 6 out of 7 scenarios. We are able to improve performance 5.35% on average just by replacing the token embeddings in zero-shot scenario. This

is a very important in case of a new disaster as usually no labeled data is available in such a scenario. Our experiment show that we can utilize labeled data from related disasters effectively in such a scenario, saving crucial time.

Embedding	Disaster	Sentiment
GloVe	62.76	52.22
W2V	<u>67.38</u>	<u>53.16</u>
CNLP	64.13	-
GCPT	<b>72.73</b>	<b>55.64</b>

Table 8: Performance comparison of GCPT with unsupervised pretrained embeddings in Zero-Shot scenario aggregated over disaster and sentiment datasets.

Model	Disaster		Sentiment	
	W2V	GCPT	W2V	GCPT
MLP-BoW	51.10	<b>57.15</b>	<b>50.89</b>	50.46
CNN	61.27	<b>67.20</b>	51.44	<b>54.45</b>
DenseCNN	61.41	<b>64.34</b>	-	-
XML-CNN	59.53	<b>62.81</b>	-	-
DPCNN	-	-	54.33	<b>56.45</b>

Table 9: W2V and GCPT performance with various neural models in Zero-Shot scenario.

#### Q5) Is GCPT effective to improve performance of transformer models?

Transformer based language models require large amount of end-task specific data to perform well. Due to this, we wanted to experiment with transformer-based models by applying GCPT in knowledge distillation setting. We follow a similar setting as [25] to fine-tune BERT [5] using Hard Knowledge Distillation. To analyze the effect of availability of additional data, we divide the current dataset train set into two equal splits and treat one as labeled and the other as unlabeled. As a baseline, a BERT model with a MLP classifier head (Bert) is trained using the labeled split. We train a 2-layer BiLSTM model with GCPT embeddings with the labeled split as teacher and predict on the unlabeled split of the data. Finally, we train another BERT with a MLP classifier (BKD) using labeled part with original labels and unlabeled part with labels predicted by the teacher model. We observe in Table 10 that additional data with GCPT predicted labels boost the model performance significantly for all 13 datasets.

## 6 CONCLUSION & FUTURE WORK

In this paper, we propose a supervised contrastive pretraining technique to effectively utilize labeled data from related datasets. Our proposed supervised graph contrastive pretraining brings tokens with similar context from same class closer. By contrasting non-contextual tokens, the learned embeddings can capture the supervised information in meaningful way that generalize well to



Dataset	Bert	BKD
AF	87.14	<b>93.74</b>
BB	87.01	<b>91.11</b>
KL	72.44	<b>77.36</b>
NE	70.62	<b>75.79</b>
OT	88.37	<b>91.72</b>
QF	90.85	<b>94.68</b>
SH	85.23	<b>88.35</b>
WE	91.82	<b>96.70</b>
Avg	84.18	<b>88.68</b>

**Table 10: Performance comparison of Bert & BKD.**

follow-up datasets. Our experiments over 8 disaster and 5 sentiment datasets show that this separability helps achieve better performance compared to utilizing the supervised information in other ways. We also show that Transformer based models can be taught (using knowledge distillation) to utilize this information. In future, we plan to utilize sub-word level token information for the graph construction which we believe will reduce the vocabulary mismatch and further improve performance.

## REFERENCES

- [1] Firoj Alam, Shafiq Joty, and Muhammad Imran. 2018. Domain Adaptation with Adversarial Training and Graph Embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1077–1087.
- [2] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, 440–447. <https://aclanthology.org/P07-1056>
- [3] George Casella and Roger Berger. 2001. *Statistical Inference*. Duxbury Resource Center.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [6] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. CERT: Contrastive Self-supervised Learning for Language Understanding. [arXiv:cs.CL/2005.12766](https://arxiv.org/abs/2005.12766)
- [7] John Giorgi, Osvad Nitski, Bo Wang, and Gary Bader. 2021. De-CLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. [arXiv:cs.CL/2006.03659](https://arxiv.org/abs/2006.03659)
- [8] Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. <https://openreview.net/forum?id=cu7IU0uhjH>
- [9] Qishi Huang, Tom Ko, H. Lilian Tang, Xubo Liu, and Bo Wu. 2021. Token-Level Supervised Contrastive Learning for Punctuation Restoration. *Interspeech 2021* (Aug 2021). <https://doi.org/10.21437/interspeech.2021-661>
- [10] Rie Johnson and Tong Zhang. 2017. Deep Pyramid Convolutional Neural Networks for Text Categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 562–570. <https://doi.org/10.18653/v1/P17-1052>
- [11] Kaggle. 2019. Natural Language Processing with Disaster Tweets. <https://www.kaggle.com/c/nlp-getting-started/overview>
- [12] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* (2020).
- [13] 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*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 18661–18673. <https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf>
- [14] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1746–1751. <https://doi.org/10.3115/v1/D14-1181>
- [15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR '17)*. <https://openreview.net/forum?id=SJU4ayYgl>
- [16] Tian Li, Xiang Chen, Shanghang Zhang, Zhen Dong, and Kurt Keutzer. 2021. Cross-Domain Sentiment Classification with Contrastive Learning and Mutual Information Maximization. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 8203–8207. <https://doi.org/10.1109/ICASSP39728.2021.9414930>
- [17] Danqi Liao. 2021. Sentence Embeddings using Supervised Contrastive Learning. [arXiv:cs.CL/2106.04791](https://arxiv.org/abs/2106.04791)
- [18] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-Label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 115–124. <https://doi.org/10.1145/3077136.3080834>
- [19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc., 3111–3119. <https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>
- [20] T. D. Nguyen, Kamla Al-Mannai, Shafiq R. Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. 2017. Robust Classification of Crisis-Related Data on Social Networks Using Convolutional Neural Networks. In *ICWSM*.
- [21] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2014. CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*, Eytan Adar, Paul Resnick, Munmun De Choudhury, Bernie Hogan, and Alice H. Oh (Eds.). The AAAI Press. <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8091>
- [22] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- [23] Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. 2020. Cross-Lingual Disaster-related Multi-label Tweet Classification with Manifold Mixup. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, Online, 292–298. <https://doi.org/10.18653/v1/2020.acl-srw-39>
- [24] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*. PMLR, 5628–5637.
- [25] Tian Shi, Liuqing Li, Ping Wang, and Chandan K. Reddy. 2021. A Simple and Effective Self-Supervised Contrastive Learning Framework for Aspect Detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 13815–13824. <https://ojs.aaai.org/index.php/AAAI/article/view/17628>
- [26] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. 2021. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3024–3033.
- [27] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: Contrastive Learning for Sentence Representation. [arXiv:cs.CL/2012.15466](https://arxiv.org/abs/2012.15466)

## A TECHNICAL APPENDIX

### A.1 Hyperparameter Configuration

We have two sets of hyperparameters to tune, during pre-training and fine-tuning. We set minimum token frequency as 5 and minimum class conditional probability (i.e.  $\tau$ ) as 0.9. Maximum number of epochs during pre-training was set to 80 with learning rate 0.005. These values were set based on classification performance in the development set. We used the above pre-training hyper-parameters throughout all experiments. Fine-tuning hyperparameters were also tuned based on the performance on the validation set with patience 4. Learning rate was chosen from values ( $10^{-2}$ ,  $10^{-3}$ ,  $10^{-4}$ ) with maximum epoch 10 for disaster and 30 for sentiment datasets. 300 dimensional input vectors were used for all experiments. We set hidden dimension as 300 whenever required for any experiments. A system with a 'Nvidia Tesla P100' GPU with 64GB RAM and 56 core CPU was used for all experiments. For BERT related experiments, maximum number of epochs were set to 5 with sequence length 128.

### A.2 Vocabulary Overlaps

We mention the token-wise overlap percentage between related and current data for each dataset after pre-processing with minimum token frequency of 5.

Dataset	Overlap %	Dataset	Overlap %
AF	81.63	books	80.94
BB	89.42	dvd	87.44
KL	65.46	electronics	83.41
NE	73.17	kitchen	84.36
OT	90.02	video	88.43
QF	81.97		
SH	89.37		
WE	87.28		

Table 11: Details of vocabulary overlap between related and current data for disaster (left) and sentiment (right) datasets.

### A.3 Dataset-wise Results

Here, we report all dataset-wise results.

Dataset	GloVe	CNLP	W2V	GCPT
AF	75.92	77.36	<u>79.74</u>	<b>83.87</b>
BB	<u>79.96</u>	78.72	78.39	<b>80.84</b>
KL	54.84	55.51	<u>57.53</u>	<b>61.30</b>
NE	<u>64.71</u>	64.16	62.93	<b>65.84</b>
OT	<u>80.56</u>	80.09	77.42	<b>82.49</b>
QF	81.54	<u>84.62</u>	84.28	<b>89.61</b>
SH	<u>85.29</u>	85.10	84.75	<b>85.61</b>
WE	<u>89.38</u>	88.31	85.24	<b>90.75</b>

Table 12: Performance comparison of GCPT with unsupervised pre-trained embeddings for disaster datasets.

Dataset	GloVe	W2V	GCPT
books	<u>51.60</u>	50.17	<b>52.88</b>
dvd	<u>55.82</u>	55.11	<b>56.43</b>
electronics	<u>58.24</u>	56.36	<b>60.79</b>
kitchen	<u>60.72</u>	59.63	<b>63.44</b>
video	54.19	<u>54.89</u>	<b>55.51</b>

Table 13: Performance comparison of GCPT with unsupervised pre-trained embeddings for sentiment datasets.

Dataset	MLP-BoW	CNN	DenseCNN	XML-CNN
AF	56.38 / 55.22	79.17 / 81.42	80.05 / 82.78	68.73 / 69.92
BB	53.30 / 50.80	78.92 / 79.62	80.53 / 83.30	71.30 / 72.81
KL	44.71 / 42.28	54.02 / 57.83	54.70 / 55.61	53.42 / 54.99
NE	48.24 / 44.23	63.37 / 64.71	63.76 / 64.47	62.65 / 63.62
OT	58.84 / 52.83	80.88 / 78.87	84.28 / 85.06	78.94 / 79.08
QF	58.08 / 56.99	88.77 / 87.59	88.77 / 89.46	81.66 / 85.36
SH	57.21 / 58.27	87.45 / 87.75	86.99 / 87.52	79.59 / 82.02
WE	53.21 / 47.04	85.84 / 87.62	91.81 / 91.75	77.64 / 83.46

Table 14: CNLP vs GCPT performance with various neural models for disaster datasets.

Dataset	MLP-BoW	CNN	DenseCNN	XML-CNN
AF	76.84 / 77.20	87.44 / 88.10	88.51 / 91.87	87.96 / 88.43
BB	66.09 / 69.11	86.03 / 87.60	86.84 / 86.81	84.62 / 86.06
KL	55.69 / 57.47	60.76 / 62.67	60.74 / 59.11	61.74 / 61.82
NE	50.75 / 51.44	64.86 / 65.22	67.23 / 65.48	67.77 / 69.07
OT	62.48 / 66.94	83.24 / 85.42	90.69 / 91.36	88.56 / 89.16
QF	75.53 / 79.50	92.03 / 94.96	92.49 / 93.46	94.02 / 94.86
SH	68.08 / 77.02	88.70 / 89.29	89.75 / 90.34	89.01 / 89.51
WE	72.45 / 81.50	92.48 / 92.81	92.48 / 93.61	92.48 / 93.31

Table 15: GloVe and GCPT performance with various neural models for disaster datasets in RDF scenario.

Dataset	MLP-BoW	CNN	DenseCNN	XML-CNN
AF	57.69 / 59.66	74.66 / 76.41	74.33 / 76.15	71.72 / 74.54
BB	58.71 / 67.76	62.30 / 68.82	61.25 / 64.31	59.74 / 60.37
KL	42.35 / 55.26	54.79 / 54.97	56.08 / 55.86	55.37 / 51.71
NE	37.14 / 32.52	51.06 / 52.53	51.54 / 52.77	47.94 / 50.17
OT	44.51 / 62.57	61.08 / 65.39	63.22 / 65.80	57.57 / 64.90
QF	58.21 / 50.95	66.11 / 81.25	66.96 / 71.26	68.85 / 77.63
SH	57.12 / 70.82	57.88 / 64.53	54.84 / 58.33	55.37 / 57.79
WE	53.07 / 57.68	62.32 / 73.72	63.11 / 70.24	59.75 / 66.08

Table 16: W2V and GCPT performance with various neural models for disaster datasets in Zero-Shot scenario.

Dataset	MLP-BoW	CNN	DPCNN
books	52.55 / 55.82	50.96 / 55.01	54.47 / 56.21
dvd	55.31 / 57.60	52.43 / 54.71	51.19 / 53.75
electronics	50.37 / 57.39	60.44 / 61.89	60.25 / 60.72
kitchen	55.13 / 55.09	59.08 / 57.01	58.40 / 61.02
video	51.37 / 51.54	51.58 / 55.66	52.49 / 55.21

Table 17: GloVe vs GCPT performance with various neural models for sentiment datasets.

Dataset	GloVe	CNLP	W2V	GCPT
AF	<u>87.47</u>	87.38	86.85	<b>89.31</b>
BB	85.31	<u>85.60</u>	83.24	<b>87.18</b>
KL	<u>62.32</u>	62.07	60.53	<b>63.44</b>
NE	<u>67.99</u>	67.61	65.81	<b>68.55</b>
OT	<u>87.31</u>	87.13	85.51	<b>88.86</b>
QF	89.52	89.69	<u>90.49</u>	<b>93.32</b>
SH	88.23	<u>88.62</u>	88.38	<b>89.38</b>
WE	<u>92.88</u>	92.72	92.70	<b>94.59</b>

Table 18: Performance comparison of GCPT with unsupervised pre-trained embeddings for disaster datasets in RDF scenario.

Dataset	GloVe	W2V	GCPT
books	<u>52.69</u>	51.73	<b>58.16</b>
dvd	<u>56.22</u>	56.16	<b>57.74</b>
electronics	54.96	<u>55.74</u>	<b>59.21</b>
kitchen	<u>57.24</u>	56.99	<b>58.80</b>
video	<u>56.27</u>	55.87	<b>58.25</b>

Table 19: Performance comparison of GCPT with unsupervised pre-trained embeddings for sentiment datasets in RDF scenario.

Dataset	MLP-BoW	CNN	DPCNN
books	45.92 / 45.26	49.75 / 51.93	55.87 / 59.16
dvd	49.22 / 48.68	50.44 / 50.57	60.50 / 61.46
electronics	52.77 / 52.89	51.46 / 55.73	55.99 / 62.56
kitchen	52.49 / 52.43	55.43 / 56.75	60.25 / 65.58
video	51.92 / 51.87	52.14 / 51.89	59.19 / 64.24

Table 20: GloVe and GCPT performance with various neural models for sentiment datasets in RDF scenario.

Dataset	Glove	CNLP	W2V	GCPT
AF	65.96	<u>66.64</u>	66.42	<b>72.82</b>
BB	64.89	65.85	<u>73.52</u>	<b>76.60</b>
KL	52.24	53.74	<u>55.69</u>	<b>57.19</b>
NE	52.43	<b>53.76</b>	52.96	<u>53.28</u>
OT	68.34	64.50	<u>70.70</u>	<b>80.91</b>
QF	67.81	73.00	<u>76.66</u>	<b>81.84</b>
SH	61.15	63.70	<u>66.56</u>	<b>73.05</b>
WE	69.26	71.86	<u>76.54</u>	<b>85.49</b>

Table 21: Performance comparison of GCPT with unsupervised pre-trained embeddings for disaster datasets in Zero-Shot scenario.

Dataset	GloVe	W2V	GCPT
books	<u>51.37</u>	50.94	<b>54.53</b>
dvd	51.75	<u>53.10</u>	<b>54.43</b>
electronics	53.27	<u>54.96</u>	<b>58.52</b>
kitchen	51.22	<u>52.59</u>	<b>54.25</b>
video	53.49	<u>54.23</u>	<b>56.51</b>

Table 22: Performance comparison of GCPT with unsupervised pre-trained embeddings for sentiment datasets in Zero-Shot scenario.

Dataset	MLP-BoW	CNN	DPCNN
books	48.17 / 50.43	48.90 / 54.26	55.58 / 57.89
dvd	50.50 / 48.52	51.46 / 53.01	52.43 / 55.15
electronics	52.29 / 55.18	52.50 / 56.52	53.15 / 57.68
kitchen	51.95 / 52.43	50.61 / 50.82	53.72 / 53.81
video	51.58 / 45.77	53.77 / 57.65	56.81 / 57.72

Table 23: W2V and GCPT performance with various neural models for sentiment datasets in Zero-Shot scenario.