SUBSTITUTING DATA ANNOTATION WITH BALANCED UPDATES AND COLLECTIVE LOSS IN MULTI-LABEL TEXT CLASSIFICATION [*]

Muberra Ozmen
McGill University
Montreal, Canada
muberra.ozmen@mail.mcgill.ca

Joseph Cotnareanu
McGill University
Montreal, Canada
joseph.cotnareanu@mail.mcgill.ca

Mark Coates
McGill University
Montreal, Canada
mark.coates@mcgill.ca

ABSTRACT

Multi-label text classification (MLTC) is the task of assigning multiple labels to a given text, and has a wide range of application domains. Most existing approaches require an enormous amount of annotated data to learn a classifier and/or a set of well-defined constraints on the label space structure, such as hierarchical relations which may be complicated to provide as the number of labels increases. In this paper, we study the MLTC problem in annotation-free and scarce-annotation settings in which the magnitude of available supervision signals is linear to the number of labels. Our method follows three steps, (1) mapping input text into a set of preliminary label likelihoods by natural language inference using a pre-trained language model, (2) calculating a signed label dependency graph by label descriptions, and (3) updating the preliminary label likelihoods with message passing along the label dependency graph, driven with a collective loss function that injects the information of expected label frequency and average multi-label cardinality of predictions. The experiments show that the proposed framework achieves effective performance under low supervision settings with almost imperceptible computational and memory overheads added to the usage of pre-trained language model outperforming its initial performance by 70% in terms of example-based F1 score.

1 INTRODUCTION

Multi-label text classification (MLTC) is the task of selecting the correct subset of labels for each text sample in a corpus. MLTC has numerous applications, such as tagging articles with the most relevant labels or recommending related search engine queries (Tsoumakas & Katakis, 2007; Varma, 2018). The majority of the literature (Liu et al., 2017; Nam et al., 2017; You et al., 2019; Ozmen et al., 2022) addresses the MLTC problem in a supervised setting, relying upon an abundance of annotated data. Despite their impressive classification performance on benchmark research datasets, most of these methods remain inapplicable in real-world applications due to the high cost of annotation.

More recently, there has been an increasing focus on the single-label text classification problem with less (Gururangan et al., 2019) or no (Meng et al., 2020) annotated data. The adaptation of methods to the multi-label scenario, however, is not straightforward and often results in significant performance deterioration. One exceptional study by Shen et al. (2021) considers hierarchical multi-label text classification without annotated data, but the algorithm requires a strict label taxonomy. Such extensive, and restrictive, prior information on the label space structure is not generally available.

In this work, we study generalized multi-label text classification, focusing on the limited annotated data setting while avoiding assumptions about the availability of strong structural information. We use a pre-trained language model to obtain preliminary label predictions using a natural language inference (NLI) framework. Pre-trained language models are trained on large-scale corpora which makes them better at recognizing patterns and relationships in natural

[*] *Citation*: Ozmen, M. Cotnareanu, J. and Coates, M. Substituting Data Annotation with Balanced Updates and Collective Loss in Multi-label Text Classification. *Proc. Conf. Lifelong Learning Agents (CoLLAs)*, 2023.
The implementation is available at https://github.com/muberraozmen/BNCL

language and allows them to handle rare words and phrases that may not appear frequently in a specific training dataset. We develop a framework that incorporates label dependencies and easily obtained supervision signals to adapt the predictions made by the pre-trained language model to the contextual properties of the specific data under study. Our experiments show that the proposed framework is efficient and effective in terms of improving the prediction performance. In summary, our key contributions are:

1. We develop a framework for multi-label text classification in two limited supervision settings:
   - (1) label descriptions and (2) expected label observation probabilities and average subset cardinality or,
   - (1) label descriptions and (2) a small set of annotated data.
2. We use multiple external linguistic knowledge bases: (1) a pre-trained language model that provides preliminary label likelihoods; (2) a set of pre-trained word embeddings to calculate signed label dependency graph.
3. We propose a model that updates preliminary likelihoods by modelling label dependencies based on balance theory and by effectively using weak supervision signals through aggregated predictions.

2 Related Work

We identify three relevant lines of research: (1) zero-shot multi-label text classification; (2) weakly supervised single-label text classification; and (3) weakly supervised hierarchical multi-label text classification.

Zero-Shot Multi-Label Text Classification. Zero-shot learning refers to a model's ability to recognize new labels that are not seen during training. This is typically achieved by learning semantic relationships between labels and input text through external linguistic knowledge bases (Yin et al., 2019). Many zero-shot learning methodologies have been developed for single-label text classification (Yin et al., 2019; Zhang et al., 2019; 2022; Ding et al., 2022). The zero-shot multi-label text classification problem remains much less explored. Most existing work specializes in biomedical text classification, namely Automatic ICD (i.e., International Classification of Diseases) coding (Rios & Kavuluru, 2018; Song et al., 2020). Rios & Kavuluru (2018) use label descriptions to generate a feature vector for each label and employ a two layer graph convolutional network (GCN) (Kipf & Welling, 2017) to encode the hierarchical label structure. Song et al. (2020) propose a framework that exploits the hierarchical structure and label descriptions to construct relevant keyword sets through an adversarial generative model. Although the setting is similar to ours in terms of problem definition, these methods rely heavily on substantial annotated data being available during training.

Weakly Supervised Single-Label Text Classification. This problem assumes that there is no access to annotated data, but the full label set, with label names, and descriptions or keywords, is available. Usually, methods employ an iterative approach, building a vocabulary of keywords for each label and evaluating the overlap between the label vocabularies and input text content. Meng et al. (2020) use a pre-trained language model, BERT (Devlin et al., 2019), to generate a list of alternative words for each label. By comparing the text, the list of candidate replacements, it is determined which words in the text are potentially class-indicative. The method is effective, but computationally very expensive, since it requires running the pre-trained language model on every word in the labelled corpus. In addition, adaptation to a multi-label scenario requires training a binary classifier for each label. Mekala & Shang (2020) argue that forming the keyword vocabulary for labels independent from the context of the input text makes it impossible for the model to differentiate between different usages of the same word. By using BERT (Devlin et al., 2019) to build context vectors, they propose a method that can associate different meanings with different labels. Zhang et al. (2021) observe that treating keywords of labels independently ignores the information embedded in keyword correlations. By building a keyword graph, the method they propose can take into account correlations using a graph neural network classifier. Existing techniques for weakly supervised text classification do not consider the multi-label classification task, and are challenging to extend to this setting because they do not account for label dependencies. The reliance on keywords limits the scope of their applicability.

Weakly Supervised Hierarchical Multi-label Text Classification. For the setting where labels are organized in a hierarchical structure, most recent methods train a multi-label classifier in a supervised fashion using GCN based architectures to encode the hierarchical relations (Peng et al., 2018; Huang et al., 2019; Zhou et al., 2020). Few methods have addressed the weakly supervised setting; an exception is the method proposed by Shen et al. (2021), which requires only label surface names in addition to category-subcategory relations represented as a directed acyclic graph. The method involves calculating a similarity score between each document-label pair using a pre-trained NLI model, and then traversing the hierarchy tree based on the similarity scores. The approach performs very well, but is limited to the setting where there is a strict hierarchy among the labels.

Table 1: Summary of problem settings with varying supervision signals

Problem Setting	Contextual Supervision					External Supervision		
	$\mathcal{D}$	$\mathcal{L}$	κ	λ_l	$\mathcal{D}_\text{A}$	$f_\text{Tokenizer}$	f_NLI	f_WE
Annotation-Free	✓	✓	✓	✓		✓	✓	✓
Scarce-Annotation	✓	✓			✓	✓	✓	✓
Domain-Supervisor	✓	✓	✓	✓	✓	✓	✓	✓

3 Problem Statement

Given a set of labels $\mathcal{L} = \{l, \varphi_l\}_{l=1}^{L}$ where $\varphi_l \in \Phi$ represents the textual description of the l^th label and L is the total number of unique labels, in multi-label text classification a sample i is associated with input text content $\vartheta_i \in \Theta$ and a subset of labels $S_i \subset \mathcal{L}$. The aim is to design a classifier that can predict output labels by input text content $f : \Theta \mapsto \mathbb{S} = \mathcal{P}(\mathcal{L}) \backslash \emptyset$ where $\mathcal{P}(.)$ denotes the power set function. Let $\hat{S}_i$ denote the label subset predicted by the classifier for sample i i.e., $f(\vartheta_i) = \hat{S}_i$. The quality of estimation can be evaluated by a variety of performance metrics that measure the similarity between the predicted $\hat{S}_i$ and the ground-truth label subset S_i. In our experiments we employ Hamming distance between binary label vectors as the primary performance metric, but we compare algorithms using multiple other assessment criteria.

In this work, we consider three scenarios with different levels of supervision used for learning $f(\cdot)$ during training. Overall, the available supervision resources under consideration are defined as follows:

- *Contextual resources*
  1. *Training data:* We are given a collection of samples $\{\vartheta_i\}_{i \in \mathcal{D}}$ without the ground-truth labels for learning $f(\cdot)$.
  2. *Label descriptions:* Labels are meaningful, i.e., they are not a set of codes or indexes, and there is a sequence of words associated with each label that provides a description. We denote the set of possible labels and their corresponding descriptions by $\mathcal{L} = \{l, \varphi_l\}_{l=1}^{L}$, where φ_l represents the textual description of the l^th label and L is the total number of unique labels. We assume that L is known and covers both training and test samples.
  3. *Average subset cardinality:* The expected number of labels per sample $\kappa = \mathbb{E}\left(|S|\right)$ is provided.
  4. *Label observation probabilities:* For each label, *a priori* probability of inclusion of that label in a subset $\lambda_l = p(l \in S)$ is provided.
  5. *Annotated data:* There is a set of training data $\{\vartheta_i, S_i\}_{i \in \mathcal{D}_\text{A}}$ such that $\mathcal{D}_\text{A} \subset \mathcal{D}_\text{train}$ and $|\mathcal{D}_\text{A}| \ll |\mathcal{D}_\text{train}|$ with provided ground truth label subsets.
- *External resources*
  1. *Tokenizer:* We are given access to a pre-trained tokenization function with vocabulary $\mathcal{V}$ which is able to convert the input text content and label descriptions into a sequence of tokens, i.e., given an input text where $\tau \in \Theta \cup \Phi$, $f_\text{tokenizer}(\tau) = (t_1, \ldots, t_s)$ such that $t_i \in \mathcal{V}$ and s is the length of the input sequence.
  2. *Language model:* We are given access to a pre-trained natural language inference model $f_\text{NLI}(\mathcal{H}, \mathcal{P})$ with vocabulary $\mathcal{V}$ that calculates true (entailment) q, undetermined (neutral) $\tilde{q}$ or false (contradiction) $\bar{q}$ probabilities of a hypothesis sequence $\mathcal{H} = (h_1, \ldots, h_{s_h})$ where $h_i \in \mathcal{V}$, given a premise sequence $\mathcal{P} = (p_1, \ldots, p_{s_p})$ where $p_j \in \mathcal{V}$ such that $q + \tilde{q} + \bar{q} = 1$.
  3. *Word embeddings:* We are given access to a set of pre-trained d-dimensional word embeddings for the (tokens composing) label descriptions, i.e., $f_\text{WE}(t) = \mathbf{e}$ where $\mathbf{e} \in \mathbb{R}^d$ denotes embeddings of token $t \in \mathcal{V}$.

We consider three different scenarios of supervision. In all scenarios, we are given *training data*, *label descriptions* and *external resources*. The inputs of each scenario are summarized in Table 1. We use a test set $\{j, \vartheta_j, S_j\}_{j \in \mathcal{D}_\text{Test}}$ such that $\mathcal{D} \cap \mathcal{D}_\text{Test} = \emptyset$ to evaluate the performance in all cases.

- *Annotation-Free:* In this scenario, we do not use any annotated data to learn the classifier but require supervision on average subset cardinality and label observation probabilities.
- *Scarce-Annotation:* In this scenario, we have access to a small set of annotated data used for training, however average subset cardinality and label observation probabilities are not provided.
- *Domain-Supervisor:* In this scenario, both a small set of annotated data and information regarding average subset cardinality and label observation probabilities are available.

4 Methodology

Our proposed framework Balanced Neighbourhoods and Collective Loss (BNCL) for multi-label text classification consists of three components: (1) *input transformation*, which maps input text into preliminary label predictions by natural language inference; (2) *parameter preparation*, which involves calculation of a label dependency graph and mean data statistics; and (3) *model update*, which updates the predictions obtained at the first stage. In this section, we share the details for each of these procedures.

4.1 Input Transformation

The aim in natural language inference (NLI) is to determine whether a *hypothesis* is true (entailment), undetermined (neutral) or false (contradiction) based on a given *premise*. Yin et al. (2019) formulate text classification as an NLI problem by treating input text as a premise and converting labels into hypotheses. To exemplify, let us consider a topic detection task on customer reviews with two possible topics of 'product' and 'delivery', for which the premise-hypothesis pairs could be developed as follows:

Premise:	Hypothesis:	Anticipated NLI output:
The material is very soft.	This review is about **product**.	entailment
The material is very soft.	This review is about **delivery**.	contradiction
The parcel did not arrive on time.	This review is about **delivery**.	entailment

When multiple true classes are not allowed (i.e., single-label classification), the entailment probabilities are compared and the largest is selected as the predicted class. In the multi-label scenario the entailment and contradiction probabilities are compared for each label independently in terms of entailment and contradiction probabilities, i.e., the problem is converted to binary relevance by ignoring neutral probabilities. For text classification, a predicted neutral for a label-specific hypothesis can be interpreted as the hesitancy of the language model to make a decision. The initial component of our proposed framework involves transforming the input text samples into a set of label-specific hypothesis probabilities using the NLI approach. This operation translates the input feature space into a 3-channel label space (i.e., entailment, neutral and contradiction probabilities). The procedure can be summarized as follows:

Step 1: Convert corpus into premises, labels into hypotheses. Following Yin et al. (2019), we build the hypothesis corresponding to label l as "This is about $\{\varphi_l\}$", where φ_l is the label description, and we calculate the corresponding sequence of tokens $\mathcal{H}_l = f_{\text{Tokenizer}}$("This is about $\{\varphi_l\}$"). Similarly, we treat each input text with content ϑ as a premise and calculate the corresponding sequence of tokens, $\mathcal{P} = f_{\text{Tokenizer}}(\vartheta)$.

Step 2: Query premise and hypothesis pairs. Given a premise $\mathcal{P}$, we query $f_{\text{NLI}}(.)$ with all hypotheses $\{\mathcal{H}_l\}_{l\in\mathcal{L}}$ to calculate $\{(q_l, \tilde{q}_l, \bar{q}_l)\}_{l\in\mathcal{L}}$. So now an input text is represented as a set of entailment, neutral and contradiction probabilities over labels:

$$\vartheta \xmapsto[\mathcal{L}=\{l,\varphi_l\}_{l=1}^{L}]{f_{\text{NLI}}(.)} \begin{array}{l} \mathbf{q} = (q_1, \ldots, q_L) \in [0,1]^L \\ \tilde{\mathbf{q}} = (\tilde{q}_1, \ldots, \tilde{q}_L) \in [0,1]^L \\ \bar{\mathbf{q}} = (\bar{q}_1, \ldots, \bar{q}_L) \in [0,1]^L \end{array}, \tag{1}$$

where q_l, $\tilde{q}_l$ and $\bar{q}_l$ correspond to the probability of the hypothesis corresponding to label l being true, undetermined, or false, respectively, given the premise ϑ. Note that $q_l + \tilde{q}_l + \bar{q}_l = 1$. So if the representation is reduced to entailment and contradiction probabilities there is no loss of information.

Predicting entailment, neutral and contradiction probabilities for label-hypotheses by this procedure does not require any training with labelled data, and therefore does not incur any substantial annotation cost. However, the predictions rely only on external resources of language modelling and not tailored to the context associated with the dataset. We argue that the classification decisions can be enhanced by (1) modelling label dependencies with the help of label-specific features; and (2) incorporating supervision cheaper to obtain compared to mass amount of annotated data.

4.2 Parameter Preparation

Given label-hypothesis representations of inputs $\{(\mathbf{q}_i, \tilde{\mathbf{q}}_i, \bar{\mathbf{q}}_i)\}_{i\in\mathcal{D}}$ as features, we learn an update function which requires (1) a signed label dependency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$, where labels are represented as vertices $\mathcal{V} = \{1, \ldots, L\}$, and edges are defined as tuples $(u, v) \in \mathcal{E}^+ (\in \mathcal{E}^-)$ indicating a positive (negative) dependency edge between labels u and v; and (2) average subset cardinality κ and label observation probabilities λ_l as dataset-specific hyper-parameters.

We form the label dependency graph using the following procedure. Given label descriptions $\mathcal{L}$ and word embeddings f_{WE}, for each label l in $\mathcal{L} = \{l, \varphi_l\}$, the label description is tokenized and the corresponding label embedding is calculated as the average of the word embeddings of the tokens that compose the labels, i.e., $\mathbf{e}_l = \sum_{t \in \mathcal{T}_l} \mathbf{e}_t / s_l$ where $\mathcal{T}_l = f_{\text{tokenizer}}(\varphi_l)$ denotes the sequence of tokens and $s_l = |\mathcal{T}_l|$ is the length of sequence. Afterwards, we calculate the cosine similarity between the embeddings of all label pairs $u, v \in \mathcal{L}$ by $d_{u,v} = \frac{\mathbf{e}_u . \mathbf{e}_v}{\|\mathbf{e}_u\| \|\mathbf{e}_v\|}$. Finally, the distances between label pairs are binarized by comparison to positive and negative edge thresholds δ^+ and δ^-, $\mathcal{E}^+ = \{(u, v) : d_{u,v} \geq \delta^+\}$ and $\mathcal{E}^- = \{(u, v) : d_{u,v} \leq \delta^-\}$.

Average subset cardinality and label observation probabilities are assumed to be provided in *annotation-free* and *domain-supervisor* settings. In *scarce-annotation*, we estimate both average statistics using the annotated set of data, i.e., $\hat{\kappa} = \frac{\sum_{i \in \mathcal{D}_\text{A}} |S_i|}{|\mathcal{D}_\text{A}|}$ and $\hat{\lambda_l} = \frac{\sum_{i \in \mathcal{D}_\text{A}} \mathbf{1}_{l \in S_i}}{|\mathcal{D}_\text{A}|}$.

4.3 Model Update

Given the signed label dependency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$, let $\mathbf{A}^+ \in \{0, 1\}^{L \times L}$ and $\mathbf{A}^- \in \{0, 1\}^{L \times L}$ denote the adjacency matrices corresponding to positive $\mathcal{E}^+$ and negative $\mathcal{E}^-$ edges, respectively.

$$\mathbf{A}^+_{ij} = \begin{cases} 1, & \text{if there is a positive edge between } i \text{ and } j \\ 0, & \text{otherwise} \end{cases}, \mathbf{A}^-_{ij} = \begin{cases} 1, & \text{if there is a negative edge between } i \text{ and } j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Since the label dependency graph is signed, finding the k-hop neighbourhoods for $k{>}1$ requires considering the interaction of negative and positive edges. Derr et al. (2018) extend the graph convolutional network (GCN) (Kipf & Welling, 2017) to signed networks based on balance theory, which states that a triad is balanced if and only if the number of negative edges is even; *the friend of my friend is my friend and the enemy of my enemy is my friend.* Based on balance theory, we define the k-hop dependencies $\mathbf{D}^{(k,+)}$ and $\mathbf{D}^{(k,-)}$ recursively as follows:

$$\mathbf{D}^{(k,+)} = \left(\mathbf{A}^+\right)^\text{T} \mathbf{D}^{(k-1,+)} + \left(\mathbf{A}^-\right)^\text{T} \mathbf{D}^{(k-1,-)} \tag{3}$$

$$\mathbf{D}^{(k,-)} = \left(\mathbf{A}^+\right)^\text{T} \mathbf{D}^{(k-1,-)} + \left(\mathbf{A}^-\right)^\text{T} \mathbf{D}^{(k-1,+)} \tag{4}$$

where $\mathbf{D}^{(1,+)} = \mathbf{A}^+$ and $\mathbf{D}^{(1,-)} = \mathbf{A}^-$. For $k = 2$, this procedure corresponds to the following neighbourhoods:

$$\begin{aligned} \mathbf{D}^{(1,+)} &= \mathbf{A}^+ && \rightarrow \text{friends} \\ \mathbf{D}^{(1,-)} &= \mathbf{A}^- && \rightarrow \text{enemies} \\ \mathbf{D}^{(2,+)} &= \left(\mathbf{A}^+\right)^\text{T} \mathbf{A}^+ + \left(\mathbf{A}^-\right)^\text{T} \mathbf{A}^- && \rightarrow \text{friends of friends + enemies of enemies} \\ \mathbf{D}^{(2,-)} &= \left(\mathbf{A}^+\right)^\text{T} \mathbf{A}^- + \left(\mathbf{A}^-\right)^\text{T} \mathbf{A}^- && \rightarrow \text{friends of enemies + enemies of friends} \end{aligned}$$

Finally, balanced neighbourhoods for label $v \in \mathcal{V}$ at hop $k \in \{1, \ldots, K\}$ are formed as follows:

$$\mathcal{N}_v^{(k,+)} = \{u : \mathbf{D}_{uv}^{(k,+)} > 0 \text{ for } u \in \mathcal{V}\}, \tag{5}$$

$$\mathcal{N}_v^{(k,-)} = \{u : \mathbf{D}_{uv}^{(k,-)} > 0 \text{ for } u \in \mathcal{V}\}. \tag{6}$$

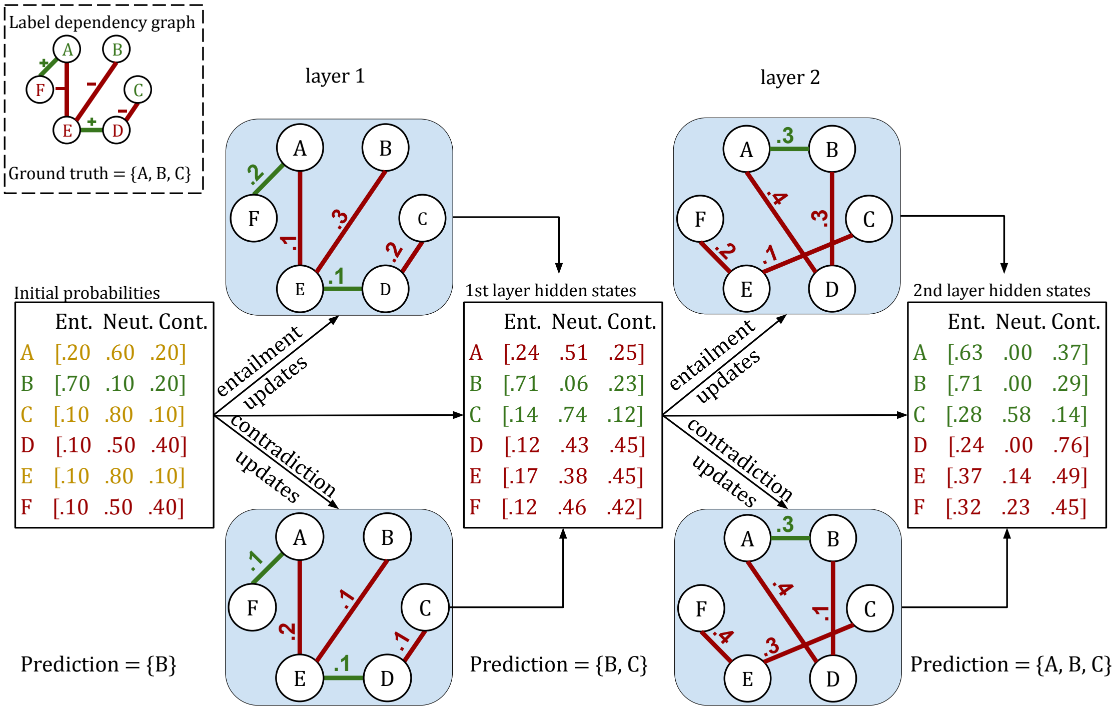
For each sample associated with entailment $\mathbf{q} = (q_1, \ldots, q_L)$ and contradiction $\bar{\mathbf{q}} = (\bar{q}_1, \ldots, \bar{q}_L)$ probabilities, we initialize its hidden representation by $\mathbf{h}^{(0)} = \mathbf{q}$ and $\bar{\mathbf{h}}^{(0)} = \bar{\mathbf{q}}$. Given the balanced neighbourhoods $\mathcal{N}_v^{(k,+)}$ and $\mathcal{N}_v^{(k,-)}$, the hidden states $\mathbf{h}^{(k)} = (h_1^{(k)}, \ldots, h_L^{(k)})$ and $\bar{\mathbf{h}}^{(k)} = (\bar{h}_1^{(k)}, \ldots, \bar{h}_L^{(k)})$ are updated at layer k as follows:

$$h_v^{(k)} = h_v^{(k-1)} + f_\text{ReLU}\left(\sum_{v \in \mathcal{N}_v^{(k,+)}} \mathbf{W}_{uv}^{(k,+)} h_u^{(k-1)}\right) + f_\text{ReLU}\left(\sum_{v \in \mathcal{N}_v^{(k,-)}} \overline{\mathbf{W}}_{uv}^{(k,-)} \bar{h}_u^{(k-1)}\right), \tag{7}$$

$$\bar{h}_v^{(k)} = \bar{h}_v^{(k-1)} + f_\text{ReLU}\left(\sum_{v \in \mathcal{N}_v^{(k,-)}} \mathbf{W}_{uv}^{(k,-)} h_u^{(k-1)}\right) + f_\text{ReLU}\left(\sum_{v \in \mathcal{N}_v^{(k,+)}} \overline{\mathbf{W}}_{uv}^{(k,+)} \bar{h}_u^{(k-1)}\right), \tag{8}$$

where $\mathbf{W}^{(k,+)}, \mathbf{W}^{(k,-)}, \overline{\mathbf{W}}^{(k,+)}, \overline{\mathbf{W}}^{(k,-)} \in \mathbb{R}^{L \times L}$ are learnable weights and $f_\text{ReLU}(.)$ denotes the Rectified Linear Unit function, i.e., $f_\text{ReLU}(x) = \max(0, x)$. Figure 1 depicts the layer udpates.

During the testing phase, for a sample i, the set of predicted labels is determined by comparing the entailment and contradiction probabilities of each label independently, i.e., $\hat{S} = \{l : p_{i,l} > \bar{p}_{i,l}, \text{ for all } l \in \{1, \ldots, L\}\}$.



Initial probabilities

	Ent.	Neut.	Cont.
A	[.20	.60	.20]
B	[.70	.10	.20]
C	[.10	.80	.10]
D	[.10	.50	.40]
E	[.10	.80	.10]
F	[.10	.50	.40]

1st layer hidden states

	Ent.	Neut.	Cont.
A	[.24	.51	.25]
B	[.71	.06	.23]
C	[.14	.74	.12]
D	[.12	.43	.45]
E	[.17	.38	.45]
F	[.12	.46	.42]

2nd layer hidden states

	Ent.	Neut.	Cont.
A	[.63	.00	.37]
B	[.71	.00	.29]
C	[.28	.58	.14]
D	[.24	.00	.76]
E	[.37	.14	.49]
F	[.32	.23	.45]

Figure 1: A toy example illustrating balanced neighbourhood update layers. Red and green edges represent negative and positive neighbourhoods at each layer. The initial and updated entailment, neutral, and contradiction probabilities are provided in tables for a sample with ground truth labels “A”, “B” and “C”. Edge attributes represent the learned weights that correspond to entailment (upper sequence) and contradiction (lower sequence) state updates. The initial predictions accepts “B”, rejects “D” and “F” and does not make a decision on “A”, “C” and “E”. The first hob update with signed label dependency graph improves the prediction by adding “C” to predictions however rejects relevant label “A”. At second hob, “A” is connected to one high entailment label “B” positively (as it is an enemy of enemy) and one high contradiction label “D” negatively (as it is friend of an enemy), which helps to improve the prediction by adding “A” to predictions. Note that the neutral probabilities drop through the updates.

Loss Function. Given the average subset cardinality κ and label observation probabilities λ_l, the updates of entailment and contradiction probabilities are guided by a loss function composed of four components. Denote the final hidden states for a sample $i \in \mathcal{D}$ by $\mathbf{p}_i = \mathbf{h}_i^{(K)}$ and $\bar{\mathbf{p}}_i = \bar{\mathbf{h}}_i^{(K)}$, where K is the total number of layers. The four components of the loss function are constructed as follows:

- By definition entailment and contradiction are mutually exclusive events for each sample and each label. Therefore, the sum of the two cannot be greater than one, and as the sum becomes closer to zero, the neutral probability (hesitation to make a decision on the presence or absence of the label for a given sample) increases. Since hesitancy is undesirable, we penalize the deviation of their summation from 1:

$$\mathbb{L}_1 = \sum_{i \in \mathcal{D}} ||\mathbf{p}_i + \bar{\mathbf{p}}_i - \mathbf{1}||_2. \tag{9}$$

  The entailment and contradiction representations are initialized to be non-negative. Since all the messages passed are non-negative, the hidden states remain non-negative (Equation 7). This, together with the $\mathbb{L}_1$ term, motivates the model to protect the probability interpretation of hidden states while reducing the neutral probability.

- For a specific training set, the classification decisions on training samples can impact each other. For example, a rare label may have very low entailment probability for all samples. Assuming the training data are representative, the samples to tag with that label can be selected by taking into account the expected observation probability. To ensure this, we penalize the difference between observed and expected probability for each label over training instances:

$$\mathbb{L}_2 = \sum_{l=1}^{L} \left(|\mathcal{D}| \times \lambda_l - \sum_{i \in \mathcal{D}} \mathbf{1}_{p_{i,l} > \bar{p}_{i,l}} \right)^2, \tag{10}$$

  where $\mathbf{1}_{p_{i,l} > \bar{p}_{i,l}}$ is 1 if entailment probability of label l on sample i is larger than its contradiction probability.

- It would be undesirable for some samples to have very high subset cardinality while others have zero. Therefore, we penalize the deviation from average subset cardinality for each sample:

$$\mathbb{L}_3 = \sum_{i \in \mathcal{D}} \left(\kappa - \sum_{l=1}^{L} \mathbf{1}_{p_{i,l} > \bar{p}_{i,l}} \right)^2 . \tag{11}$$

- In *scarce-annotation* and *domain-supervisor* settings, we have a small set of annotated data $\mathcal{D}_\text{A}$. Let $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,L})$ denote the binary vector that corresponds to ground truth labels S_i of sample i:

$$\mathbb{L}_4 = \sum_{i \in \mathcal{D}_\text{A}} \sum_{i=1}^{L} -y_{i,l} \log(p_{i,l}) + (1 - y_{i,l}) \log(\bar{p}_{i,l}). \tag{12}$$

In order to make the term $\mathbf{1}_{p_{i,l} > \bar{p}_{i,l}}$ differentiable, we use a sharpened version of the sigmoid with a constant $C > 1$:

$$\frac{1}{1 + e^{C \times (p_{i,l} - \bar{p}_{i,l})}} \approx \mathbf{1}_{p_{i,l} > \bar{p}_{i,l}} . \tag{13}$$

The final loss function follows:

$$\mathbb{L} = \mathbb{L}_1 + \alpha_2 \mathbb{L}_2 + \alpha_3 \mathbb{L}_3 + \alpha_4 \mathbb{L}_4, \tag{14}$$

where $\{\alpha_j\}_{j=2}^{4}$ are hyperparameters used to scale the individual components of the loss function.

5 Experiments

Datasets. For our experiments we use two multi-label text classification datasets: Reuters21578[1] (Lewis et al., 2004), which is a collection of newswire stories; and StackEx-Philosophy[2] (Charte & Charte, 2015), which is a collection of posts in Stack Exchange Philosophy forums. The dataset statistics are provided in Appendix A. For both datasets, the label set is formed by the topics of the sample texts. For example, in Reuters21578 "interest rates" and "unemployment" are label descriptions, and in StackEx-Philosophy "ethics" and "skepticism" are label descriptions.

Metrics. In addition to Hamming accuracy (HA), we use example based F1 score (ebF1), subset accuracy (ACC), micro-averaged F1 score (miF1), and macro-averaged F1 score (maF1) as metrics to evaluate the performance of our method. Subset accuracy measures the fraction of times that an algorithm identifies the correct subset of labels for each instance. The example-based F1 score is aggregated over samples and the macro-averaged F1 score over labels. The micro-averaged F1 score takes the average of the F1 score weighted by the contribution of each label, and thus takes label imbalance into account. Expressions for the metrics are provided in equations 16 - 20 in Appendix B.

Baselines. To the best of our knowledge, the problem settings under consideration in this study have not been explored directly in the literature. The weakly supervised text classification problem (in which only label descriptions are given) has been studied primarily in the single-label classification context (Meng et al., 2020; Mekala & Shang, 2020; Zhang et al., 2021; Zeng et al., 2022). Some works impose constraints such as the requirement that all label indicative keywords should be seen in the corpus. We attempted to adapt one of the state-of-the-art methods, LOTClass (Meng et al., 2020), to the multi-label scenario, but encountered errors regarding these constraints. Execution was only possible if more than half of the labels in the dataset were excluded. TaxoClass (Shen et al., 2021) considers multi-label classification with no annotated data, but it requires a hierarchy tree which represents category-subcategory types of relations between labels. Furthermore, all labels must be aligned exactly with the hierarchy tree. We compare to:

- **0Shot-TC (Yin et al., 2019)** (multi-label version), which uses the NLI formulation of the text classification task. Estimated entailment/contradiction probabilities per label are used directly to make classification decisions. Since the same formulation is used for our input transformation, this comparison reveals the impact of our "model update" module on multi-label classification performance in contrast to using raw language model output.
- **ML-KNN (Zhang & Zhou, 2007)**, a multi-label classifier originally designed for the supervised setting, which finds the nearest examples to a test class using the k-Nearest Neighbors algorithm and then selects the assigned labels using Bayesian inference.
- **ML-ARAM (Benites & Sapozhnikova, 2015)**, a multi-label classifier designed for the supervised setting, which use Adaptive Resonance Theory (ART) based clustering and Bayesian inference to calculate label probabilities.

[1] available at https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[2] available at https://archive.org/download/stackexchange/philosophy.stackexchange.com.7z

Table 2: Comparison between the proposed method and 0Shot-MLTC in the Annotation-Free setting. The table also shows performance for the Scarce-Annotation and Domain-Supervisor settings. All results are calculated over 10 random initialization on the original train-test data splits.

		Reuters21578					*StackEx-Philosophy*				
		ACC	HA	ebF1	miF1	maF1	ACC	HA	ebF1	miF1	maF1
0Shot-MLTC		0.0834	0.9799	0.2778	0.2981	0.1844	0.001	0.8802	0.0924	0.0665	0.1528
BNCL	mean	0.3159	0.9917	0.4613	0.5053	0.2184	0.0382	0.9902	0.2119	0.2423	0.2292
Annotation-Free	*std*	*0.0446*	*0.0007*	*0.0393*	*0.0343*	*0.0034*	*0.0035*	*0.0001*	*0.0042*	*0.0035*	*0.0054*
BNCL	mean	**0.5083**	**0.9944**	0.6318	0.6595	0.2340	0.0547	**0.9913**	**0.2655**	**0.3024**	0.2304
Scarce-Annotation	*std*	*0.0134*	*0.0001*	*0.0131*	*0.0101*	*0.0091*	*0.0030*	*0.0002*	*0.0055*	*0.0047*	*0.0091*
BNCL	mean	0.5078	**0.9944**	**0.6320**	**0.6606**	**0.2353**	**0.0551**	**0.9913**	0.2648	0.3021	**0.2309**
Domain-Supervisor	*std*	*0.0120*	*0.0001*	*0.0145*	*0.0108*	*0.0085*	*0.0028*	*0.0001*	*0.0058*	*0.0059*	*0.0090*

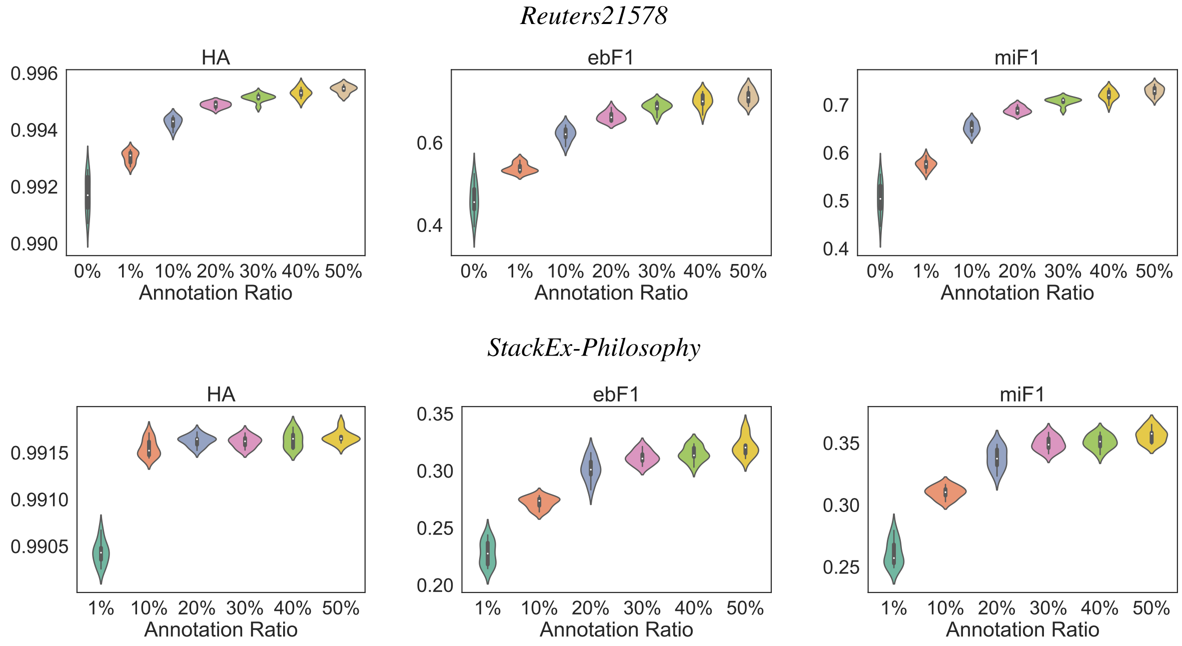
Experimental settings. We examine performance in three different experimental settings, as identified in the problem statement. In "Annotation-Free", no annotated data is available for training, but we assume knowledge of average subset cardinality and label observation probabilities. In "Scarce Annotation", a small annotated dataset is available. In our experiments, the annotated dataset size is set to L. In "Domain-Supervisor", in addition to the annotated dataset, knowledge concerning the average subset cardinality and label observation probabilities is available.

Implementation. We transform the input using the pre-trained model BART (Lewis et al., 2020) and its corresponding tokenizer, which is fine-tuned on a large corpus, MNLI (Williams et al., 2018), composed of hypothesis-premise pairs. For both datasets, the maximum sequence length of the tokenizer is set to 128. We use GloVe (Pennington et al., 2014) to generate word embeddings to calculate the label graph from the label descriptions. The positive and negative edge thresholds δ^+ and δ^- that control label graph density are set by top-bottom percentiles of the overall distribution of the distances between label embeddings. For each dataset, it is selected from the following list of percentile pairs $[(5\%, 95\%), (10\%, 90\%), (30\%, 70\%)]$. When the average subset cardinality and label observation probabilities are assumed to be provided, they are calculated based on the whole set of training data. In the scarce-annotation and domain-supervisor settings, the size of the annotated dataset is L (i.e., $|\mathcal{D}_\mathrm{A}| = |\mathcal{L}| = L$.). The annotated examples are randomly selected from the training set. The sigmoid sharpening factor C is set to 10. The procedure to select this value was: (1) sample a small set of examples; (2) compare their entailment and contradiction probabilities to determine predicted label subsets; (3) calculate the sharpened sigmoid function value, successively increasing the integer C by one; and (4) choose the smallest integer C such that the output for all sample/predicted label pairs is greater than 0.9999. The loss function scaling factors $\{\alpha_j\}_{j=2}^4$ are tuned using grid search over $\alpha_2, \alpha_3 \in \{0.1, 0.5, 1\}$ and $\alpha_4 \in \{1, 10, 100\}$. The selected values for both datasets are at $\alpha_2 = 0.1, \alpha_3 = 0.5, \alpha_4 = 100$. If not stated otherwise, the number of update layers is set to 2 because a smaller number caused validation performance to be too sensitive to label graph density, and a greater number reduced the performance on the validation data. The model is trained with a batch size of 128 for 30 epochs as it is observed that validation performance does not improve after 30 epochs. The Adam (Kingma & Ba, 2015) optimizer is used to compute gradients and update parameters with the initial learning rate of 1×10^{-3} and beta coefficients of $(0.8, 0.9)$. The learning rate is updated with a step size 10 for a 10% decay rate. The results for ML-KNN and ML-ARAM are obtained by implementations provided in the scikit-learn library (Pedregosa et al., 2011). These algorithms are both trained using the full sets of training data. In the comparison with these supervised algorithms, we train our algorithm using 50% of the annotations.

Comparison with 0Shot-MLTC. Table 2 compares the performance of 0Shot-TC adapted to multi-label scenario and the performance of our proposed method, BNCL. We examine how BNCL performs in three settings. In the annotation-free setting, we see that BNCL achieves much better performance for all metrics. This indicates how valuable it is to construct the signed label dependency graph and use it to update the embeddings using the signed graph convolution network. Moving from the annotation-free to the scarce-annotation setting, subset accuracy improves by 63% and 43%, example-based F1 score by 37% and 25% and micro-averaged F1 score by 31% and 25%, on Reuters21578 and StackEx-Philosophy, respectively. This shows that having a small set of annotated data is very helpful. There is no meaningful performance difference between the scarce-annotation and domain-supervisor settings, which suggests that when a small amount of annotation is available, there is no need for supervision in terms of average label subset cardinality and label observation probabilities.

Table 3: Comparison with supervised baseline methods. BNCL outperforms the baselines with in the 100% Annotation setting and achieves equivalent performance for 50% Annotation. Performance degradation as the annotation level decreases is graceful.

	Reuters21578					*StackEx-Philosophy*				
	ACC	HA	ebF1	miF1	maF1	ACC	HA	ebF1	miF1	maF1
ML-KNN	0.6513	0.9956	0.7101	0.7228	0.2536	**0.0904**	**0.9925**	0.2866	0.3198	0.0993
ML-ARAM	0.4742	0.9923	0.6734	0.6265	0.1633	0.0622	0.9888	0.2045	0.1796	0.0075
BNCL-100%	**0.6674**	**0.9961**	**0.7772**	**0.7720**	**0.2784**	0.0803	0.9917	**0.3394**	**0.3749**	0.2387
BNCL-50%	0.6039	0.9954	0.7120	0.7286	0.2589	0.0723	0.9917	0.3283	0.3642	**0.2586**
BNCL-20%	0.5618	0.9949	0.6881	0.7028	0.2459	0.0643	0.9916	0.2948	0.3241	0.2418
BNCL-5%	0.3784	0.9922	0.5763	0.5853	0.2439	0.0331	0.9880	0.2696	0.2813	0.2266


Reuters21578

StackEx-Philosophy

Figure 2: Sensitivity study with different amounts of annotated data

Comparison to Supervised Learning Methods. Table 3 compares the results of two supervised baseline methods which are trained with full set of training data to BNCL, which is trained with various ratios of data annotation. We find that BNCL with only 50% annotation achieves similar performance to the supervised baseline methods. We also find that BNCL with 100% performs better than the supervised methods, even though it is designed for the limited annotation settings and uses less annotated data in this setting.

Sensitivity Study - Annotation Level. In order to observe the impact of the amount of annotated data, we perform a sensitivity study by changing the level of annotation in the domain-supervisor setting. In Figure 2, the performance for different levels of annotation is presented for both datasets in terms of Hamming accuracy, example-based F1 score, and micro-averaged F1 score. The pattern observable for all three metrics on both datasets is that even a small set of annotated data (as little as 1% of the training data) is capable of improving the performance. But as the amount of annotation increases, the improvement diminishes, and beyond 30%, there is much less value in further annotation.

Ablation Study - Loss Function Components. In order to understand the impact of individual loss components, we perform an ablation study. We examine the impact of removing $\mathbb{L}_2$, which targets matching of the label observation probabilities, and $\mathbb{L}_3$, which aims to balance the subset cardinality. The study is conducted in the annotation-free setting on the StackEx-Philosophy dataset. Table 4 compares the results of the original loss function to configurations

Table 4: Ablation study for loss function components on StackEx-Philosophy dataset

		ACC	HA	ebF1	miF1	maF1
BNCL	mean	**0.0382**	**0.9902**	0.2119	0.2423	**0.2292**
Original	*std*	*0.0035*	*0.0001*	*0.0042*	*0.0035*	*0.0054*
BNCL	mean	0.0236	0.9899	0.1921	0.2222	0.2267
Removing $\mathbb{L}_2$	*std*	*0.0028*	*0.0001*	*0.0048*	*0.0040*	*0.0063*
BNCL	mean	0.0347	0.9865	**0.2373**	**0.2458**	0.2025
Removing $\mathbb{L}_3$	*std*	*0.0085*	*0.0018*	*0.0136*	*0.0032*	*0.0083*
BNCL	mean	0.0004	0.7996	0.0509	0.0374	0.0736
Removing $\mathbb{L}_2$ *and* $\mathbb{L}_3$	*std*	*0.0005*	*0.0092*	*0.0027*	*0.0016*	*0.0034*

(1) without the $\mathbb{L}_2$ component; (2) without the $\mathbb{L}_3$ component; and (3) without both. The results show that using label observation probabilities to guide the update of label-hypothesis probabilities significantly improves the performance. On the other hand, removing the $\mathbb{L}_3$ component that is associated with balancing subset cardinality results in a relatively small deterioration in accuracy, and actually improves the example based and micro-averaged F1 scores. This is likely due to the power law distribution that label subset cardinalities follow in the StackEx-Philosophy dataset. Constraining each sample's label subset cardinality to an expected level may harm the example based performance especially in terms of most frequent labels (note that miF1 weighs infrequent labels less compared to maF1). When both components are missing the performance drops dramatically. This indicates that either component can provide valuable regularizing information, but without both, the model update module is drawn towards poor representations.

6 Conclusion

Summary and Contributions. In this study, we propose a framework for multi-label text classification in the absence of strong supervision signals. Our framework performs transfer learning using external knowledge bases, and exploits the benefits of modelling the dependencies between labels in order to focus the external supervision on domain-specific properties of the data. We project input text onto a label-hypothesis probability space using a pre-trained language model and then update representations using the guidance of label dependencies and aggregated predictions over the training data. To the best of our knowledge this is the first work that considers weakly supervised multi-label text classification problem when the label space is not strictly structured according to a set of label hierarchies.

Limitations and Future Work. Extreme Multi-label Learning (XML) involves finding the most relevant subset of labels for each data point from an extremely large label set. The number of labels can scale to thousands or millions. Using our method in an extreme classification setting would be infeasible due to the computational overhead of the input transformation process (we need to calculate probabilities for every candidate label for every text example). One future work direction we would like to explore is developing an active learning based framework in order to select the labels to query for each input text. Another limitation associated with the proposed method is the inability to handle noise in the values provided by a domain-supervisor with respect to average subset cardinality or label observation probabilities. We also do not take into account any uncertainty in the signed label graph constructed from the label descriptions. Therefore, desirable follow up work involves improving our methodology by incorporating Bayesian approaches to account for uncertainty in the estimated parameters and label dependency graph.

References

F. Benites and E. Sapozhnikova. Haram: A hierarchical aram neural network for large-scale text classification. In *Proc. IEEE Int. Conf. Data Mining Workshop*, pp. 847–854, 2015.

Francisco Charte and David Charte. Working with multilabel datasets in r: The mldr package. *R J.*, 7:149, 2015.

Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional network. In *IEEE Int. Conf. Data Mining (ICDM)*, pp. 1066–1075, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. Association for Computational Linguistics (ACL)*, pp. 4171–4186, 2019.

Hantian Ding, Jinrui Yang, Yuqian Deng, Hongming Zhang, and Dan Roth. Towards open-domain topic classification. In *Proc. NAACL - Human Language Technologies*, pp. 90–98, 2022.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. Variational pretraining for semi-supervised text classification. In *Proc. Association for Computational Linguistics (ACL)*, 2019.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *In. Proc. Conf. Information and Knowledge Management (CIKM)*, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *J. Machine Learning Research*, 5:361–397, 2004.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. Association for Computational Linguistics (ACL)*, pp. 7871–7880, 2020.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep learning for extreme multi-label text classification. In *Proc. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 115–124, 2017.

Dheeraj Mekala and Jingbo Shang. Contextualized weak supervision for text classification. In *Proc. Association for Computational Linguistics (ACL)*, pp. 323–333, 2020.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. Text classification using label names only: A language model self-training approach. In *Proc. Conf. Empirical Methods Natural Language Processing (EMNLP)*, pp. 9006–9017, 2020.

J. Nam, E. Loza Mencía, H. J. Kim, and J. Fürnkranz. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5413–5423, 2017.

M. Ozmen, H. Zhang, P. Wang, and M. Coates. Multi-relation message passing for multi-label text classification. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: machine learning in python. *J. Machine Learning Research*, 12:2825–2830, 2011.

Hao Peng, Jianxin Li, Y. He, Yaopeng Liu, Mengjiao Bao, L. Wang, Y. Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proc. World Wide Web Conf.*, 2018.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proc. Conf. Empirical Methods Natural Language Processing (EMNLP)*, 2014.

Anthony Rios and Ramakanth Kavuluru. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proc. Conf. Empirical Methods Natural Language Processing (EMNLP)*, pp. 3132–3142, 2018.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. Taxoclass: Hierarchical multi-label text classification using only class names. In *Proc. NAACL - Human Language Technologies*, pp. 4239–4249, 2021.

Congzheng Song, Shanghang Zhang, Najmeh Sadoughi, Pengtao Xie, and Eric Xing. Generalized zero-shot text classification for icd coding. In *Proc. Int. Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4018–4024, 2020.

G. Tsoumakas and I. Katakis. Multi-label classification: an overview. *Int. J. Data Warehousing and Mining*, 3:1–13, 2007.

Manik Varma. Extreme classification: Tagging on wikipedia, recommendation on amazon and advertising on bing. In *Proc. The Web Conference*, pp. 1897, 2018.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. Association for Computational Linguistics (ACL)*, 2018.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proc. Conf. Empirical Methods Natural Language Processing (EMNLP)*, pp. 3914–3923, 2019.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Proc. Adv. Neural Information Processing Systems (NeurIPS)*, 2019.

Ziqian Zeng, Weimin Ni, Tianqing Fang, Xiang Li, Xinran Zhao, and Yangqiu Song. Weakly supervised text classification using supervision signals from a language model. In *Findings of the Association for Computational Linguistics: NAACL*, pp. 2295–2305, 2022.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. In *Proc. NAACL - Human Language Technologies*, pp. 1031–1040, 2019.

Lu Zhang, Jiadong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. Weakly-supervised text classification based on keyword graph. In *Proc. Conf. Empirical Methods Natural Language Processing (EMNLP)*, pp. 2803–2813, 2021.

M. Zhang and Z. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7): 2038–2048, 2007.

Yiwen Zhang, Caixia Yuan, Xiaojie Wang, Ziwei Bai, and Yongbin Liu. Learn to adapt for generalized zero-shot text classification. In *Proc. Association for Computational Linguistics (ACL)*, pp. 517–527, 2022.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and G. Liu. Hierarchy-aware global model for hierarchical text classification. In *Proc. Association for Computational Linguistics (ACL)*, 2020.

APPENDICES

A DATASET STATISTICS

Table 5: Dataset Statistics

Dataset	*# of Labels*	*# of Train*	*# of Test*	*Ave. Labels per Sample*	*Ave. Samples per Label*
Reuters21578	135	7694	3010	1.24	70.62
StackEx-Philosophy	294	3983	996	2.34	31.74

B EVALUATION METRICS

Instance based metrics: Given L number of labels and M number of samples, let $\mathbf{y}_i = (y_{i1}, \dots y_{iL})$ and $\hat{\mathbf{y}}_i = (\hat{y}_{i1}, ..., \tilde{y}_{iL})$ denote binary vectors that corresponds to ground-truth and predicted labels on sample i respectively. That is:

$$y_{il} = \begin{cases} 1, & \text{if } l \in S_i \\ 0, & \text{otherwise} \end{cases}, \hat{y}_{il} = \begin{cases} 1, & \text{if } l \in \hat{S}_i \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Subset accuracy is defined as follows:

$$\text{ACC} = \frac{1}{M} \sum_{i=1}^{M} I[\mathbf{y}_i = \hat{\mathbf{y}}_i] \tag{16}$$

Hamming accuracy is defined as follows:

$$\text{HA} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{L} I[y_{il} = \hat{y}_{il}] \tag{17}$$

Example-based F1 score is defined as follows:

$$\text{ebF1} = \frac{1}{M} \sum_{i=1}^{M} \frac{2 \sum_{l=1}^{L} y_{il} \hat{y}_{il}}{\sum_{l=1}^{L} y_{il} + \sum_{l=1}^{L} \hat{y}_{il}} \tag{18}$$

Label based metrics: Each label l is treated as a separate binary classification problem (each label l has its own confusion matrix of true-positives (tp_l), false-positives (fp_l), true-negatives (tn_l), false-negatives (fn_l).) Micro-averaged F1 score is defined as follows:

$$\text{miF1} = \frac{\sum_{l=1}^{L} 2tp_l}{\sum_{l=1}^{L} 2tp_l + fp_l + fn_l} \tag{19}$$

Macro-averaged F1 score is defined as follows:

$$\text{maF1} = \frac{1}{L} \sum_{l=1}^{L} \frac{2tp_l}{2tp_l + fp_l + fn_l} \tag{20}$$

C Sensitivity Study - Noise in Label Frequency Estimates.

We conducted an additional sensitivity study on the effect of noise in the input label frequency estimates. We (1) divide the labels into clusters by their observation frequency into k groups, e.g. group 1 higher than 70% frequency, . . . , group k less than 0.01% frequency and (2) associate each group member's expected observation frequency to the average observation frequency of its group. The change of resultant performance on annotation-free setting by the number of groups k are presented in Table 6.

Table 6: Sensitivity study with different numbers of observation frequency-based label groups (k) in annotation-free setting. All results are calculated over 10 random initialization on the original train-test data splits.

		Reuters21578					*StackEx-Philosophy*				
		ACC	HA	ebF1	miF1	maF1	ACC	HA	ebF1	miF1	maF1
0Shot-MLTC		0.0834	0.9799	0.2778	0.2981	0.1844	0.0010	0.8802	0.0924	0.0665	0.1528
BNCL	k = 2	0.0950	0.9903	0.1654	0.2401	0.1663	0.0226	0.9900	0.1899	0.2218	0.2290
	k= 4	0.2753	0.9918	0.3926	0.4612	0.1740	0.0355	0.9901	0.2078	0.2390	0.2294
	k = 6	0.2758	0.9918	0.3936	0.4621	0.1741	0.0358	0.9902	0.2084	0.2392	0.2302
	k = 8	0.2765	0.9918	0.3945	0.4631	0.1740	0.0363	0.9902	0.2096	0.2399	0.2298
	k = 10	0.2770	0.9918	0.3950	0.4637	0.1743	0.0370	0.9902	0.2108	0.2415	0.2303

Based on this sensitivity study, we conclude that if the domain-supervisor can reasonably group the labels into four by their observation frquency, e.g. almost unseen, uncommon, standard, common, the approximate identification leads to a dramatic improvement compared to the baseline performance on the datasets under interest.