Using Pseudo-Labelled Data for Zero-Shot Text Classification

Congcong Wang[1][0000−0002−6800−7642], Paul Nulty[2][0000−0002−7214−4666], and David Lillis[1][0000−0002−5702−4463]

[1] School of Computer Science, University College Dublin
congcong.wang@ucdconnect.ie, david.lillis@ucd.ie
[2] Department of Computer Science and Information Systems, Birkbeck, University of London
p.nulty@bbk.ac.uk

Abstract. Existing Zero-Shot Learning (ZSL) techniques for text classification typically assign a label to a piece of text by building a matching model to capture the semantic similarity between the text and the label descriptor. This is expensive at inference time as it requires the text paired with every label to be passed forward through the matching model. The existing approaches to alleviate this issue are based on exact-word matching between the label surface names and an unlabelled target-domain corpus to get pseudo-labelled data for model training, making them difficult to generalise to ZS classification in multiple domains, In this paper, we propose an approach called P-ZSC to leverage **p**seudo-labelled data for **z**ero-**s**hot text **c**lassification. Our approach generates the pseudo-labelled data through a matching algorithm between the unlabelled target-domain corpus and the label vocabularies that consist of in-domain relevant phrases via expansion from label names. By evaluating our approach on several benchmarking datasets from a variety of domains, the results show that our system substantially outperforms the baseline systems especially in datasets whose classes are imbalanced.

Keywords: Text classification · Zero-shot learning · Weakly-supervised learning.

1 Introduction

Recent years have seen numerous studies achieving great success in applying neural network models to text classification [3,5,21,22]. However, most are based on supervised learning, requiring human annotation for the training data. To mitigate the annotation burden, attention has been increasingly paid to seeking semi-supervised or unsupervised learning approaches for classification tasks via model training that uses minimal or no annotated data from the target task. Zero-shot learning (ZSL) is one example of this [16,17,27,29,14]. Yin et al. [28] define ZSL as having two categories: *label-partially-unseen* and *label-fully-unseen*. The former refers to a situation where a classifier is learnt from labelled examples from a set of known classes and tested on the union of these previously-seen classes and a set of unseen classes. The latter restricts a classifier from seeing any task-specific labelled data in its model development. It is a more challenging problem. Our work falls in the context of *label-fully-unseen* ZSL.

In the context of *label-fully-unseen* ZSL, a matching model is commonly trained to capture the semantic similarity between text pairs. At training time, the text pair usually consists of a document and a label. This approach does not rely on any task-specific labelled data for model training. Semantic matching between general text pairs can then be transferred to downstream tasks. However, an example needs to be paired with every candidate label to pass forward through the matching model. This results in inefficiencies at inference time, especially with many possible labels.

In order to alleviate this problem, we present a classifier-based ZSL. Inference is more efficient because the classifier outputs a single class distribution given a single text as the input. Classifier-based ZSL, which is sometimes described as a type of extremely weakly-supervised learning [12,13,23], generates pseudo-labelled data for model training using label names only [3]. Existing work [12,13,23] generate the pseudo-labelled data based on exact-word matching between the label names and an unlabelled (task-specific) corpus. For corpora that have a heavy class imbalance, this can lead to minority classes without any pseudo-labelled examples. Hence, we propose P-ZSC: a simple yet effective approach for zero-shot text classification. In our approach, we propose a technique based on sentence embeddings (semantic matching) for label phrase expansion. To get the pseudo-labelled data, we use a confidence-based algorithm that assigns a label to an example based on a matching score between the label's expanded phrases and the example text. We pre-train the classifier on the pseudo-labelled data at document level and then self-train the classifier on the remaining unlabelled corpus. The results show that our system outperforms baseline systems by a large margin on a variety of datasets with different characteristics.

2 Related Work

As our work focuses on label-fully-unseen classifier-based ZSL. we examine related work on **Label-fully-Unseen ZSL** and **Weakly-supervised classification**.

2.1 Label-Fully-Unseen ZSL

For label-fully-unseen ZSL, most work has explored the problem through indirect (or distant) supervision from other problems. Pushp and Srivastava [17] propose three neural networks to learn the relationship between text pairs consisting of news headlines along with their SEO tags. Yin et al. [28] propose an entailment approach that applies pre-trained BERT to learn the relationship between text pairs that consist of the premises and hypotheses from three textual entailment datasets. Muller et al. [14] apply siamese networks and label tuning to tackle inefficiency issue at inference time in the entailment approach. Puri et al. [16] achieve zero-shot model adaptation to new classification tasks via a generative language modelling task. Other similar works using indirect supervision can be found in [9] and [15], who study zero-shot relation extraction by transforming it into a machine comprehension and textual entailment problem respectively. However, most of these works fall into the matching-based ZSL category.

[3] Since the label surface names are available at testing time with no need for human supervision, we describe it as classifier-based ZSL. In addition, no task-specific labelled data is used, thus meeting the definition of *label-fully-unseen* ZSL in [28].

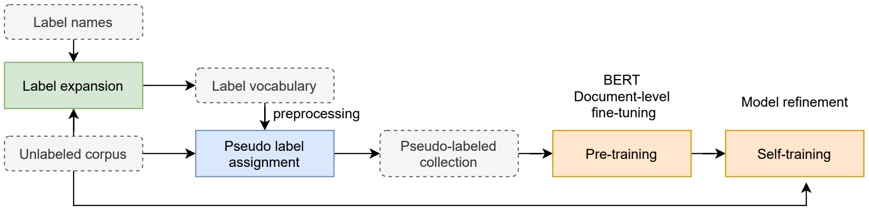


Fig. 1: The architecture of our system

2.2 Weakly-supervised Classification

Due to the inefficiency of matching-based ZSL, another line of work has looked into achieving classifier-based ZSL via label names in a weakly-supervised fashion. WeST-Class is a framework for weakly-supervised learning neural text classification [12]. In [11], some metadata entities (e.g., authors, venues) are used as a source of weak supervision for text classification (META). ASTRA uses a few labeled data along with a self-training procedure on the (task-specific) unlabelled data for text classification [4]. ConWea uses a few human-provided seed-words to enrich the raw label names for pseudo-labelling data [10]. Despite weak supervision for offering seed words or a few labelled data, human effort is still involved in these approaches. The most recent works similar to ours are LOTClass [13] and X-Class [23], which use only label names for text classification, known as extremely weakly-supervised learning. LOTClass obtains pseudo-labelled data via a masked category prediction (MCP) task of BERT [3] at the token level. X-Class first leverages pre-trained BERT to obtain class-oriented document representations and then pseudo-labels data by applying clustering to the representations. However, both are exact-word matching based approaches, bringing difficulty in generalising to domains where the label names do not usually appear in the raw corpus. One goal of the P-ZSC proposed in this paper is to overcome these limitations.

3 Method

Formally, the problem that our system aims to solve is defined as follows: for a text classification task $\mathcal{T}$, given n label names $\mathcal{Y} : \{y_1, y_2, ...y_n\}$ and an unlabelled corpus $\mathcal{D} : \{d_1, d_2, ...d_m\}$ containing m documents from this task domain, the objective is to train a model f that can assign one or more labels from $\mathcal{Y}$ to an example x based on its probability estimation over $\mathcal{Y}$, namely, $f(x) : p(y_1|x), p(y_2|x), ..., p(y_n|x)$.

As illustrated in Figure 1, our system has three stages: **label expansion**, **pseudo label assignment**, **pre-training and self-training**.

3.1 Label Expansion

In a typical classification task, the label names are only one or two words, which are insufficient to convey their potentially broad meaning. To enrich a label's meaning, we propose a simple **label expansion (LE)** algorithm to find the most semantically-similar words or phrases to the label from the unlabelled in-domain corpus $\mathcal{D}$.

Given $\mathcal{Y}$ and $\mathcal{D}$, first we combine all examples from $\mathcal{D}$ and split them into n-gram phrases consisting of all 1-grams, 2-grams and 3-grams (including overlaps): $\mathcal{G} : \{g_1, g_2, g_3, ..., g_L\}$. Then, we calculate the similarity between $\mathcal{Y}$ and $\mathcal{G}$ via a sentence embedding model. Basically, given a sentence embedding model E,[4] the matching score $\hat{s}_{i,j}$ between a label y_i and an n-gram g_j is calculated by the cosine similarity between the label's embedding and the n-gram's embedding:

$$\hat{s}_{i,j} = \text{cosine}(E_{avg}(y_i), E_{avg}(g_j)) \tag{1}$$

For any y_i or g_j with more than one token, E_{avg} takes the average pooling as the output embedding. After this, each label maintains a vocabulary of expanded phrases ranked by the similarity score. The vocabulary for each label y_i is denoted by $\hat{\mathcal{V}}_i : \{(\hat{v}_{i,k}, \hat{s}_{i,k})\}|_{k=1}^{\hat{K}}$ where $\hat{v}_{i,k}$ represents the kth expanded phrase in the vocabulary and $\hat{s}_{i,k}$ is the corresponding matching score.

Label vocabulary pre-processing: To maintain the quality of the label vocabulary, we include a further pre-processing step to optimise the original vocabulary $\hat{\mathcal{V}}_i$ and we denote a label's pre-processed vocabulary as: $\mathcal{V}_i : \{(v_{i,k}, s_{i,k})\}|_{k=1}^{K}$. We select only those phrases in $\hat{\mathcal{V}}_i$ where $\hat{s}_{i,k} \geq 0.7$, maintaining a minimum of 2 phrases and maximum of 100 per label[5]. We then apply a discounting on the phrases that co-occur across different labels, which is calculated by

$$s_{i,k} = \hat{s}_{i,k} * log_e\left(\frac{n}{LF(v_{i,k})}\right) \tag{2}$$

where n is the number of labels and $LF(v_{i,k})$ is the frequency of phrase $v_{i,k}$ across the vocabularies of all labels.

3.2 Pseudo label assignment

After expansion, we next construct a labelled collection (pseudo-labelled data) for model training. Due to the lack of annotated data, the alternative is to construct the collection via the process of **pseudo label assignment (PLA)**. We adopt a simple approach for PLA, which is described as follows:

A document $d_j \in \mathcal{D}$ is matched with a label's vocabulary $\mathcal{V}_i : \{(v_{i,k}, s_{i,k})\}|_{k=1}^{K}$ (from the previous section) through a cumulative scoring mechanism. To assign a label y_i to a document d_j, a matching score between them is first calculated by

$$s^*_{j,i} = \sum_{k=0}^{K} s_{i,k}[v_{i,k} \in G_j] \tag{3}$$

where G_j is the set of n-grams of d_j. For consistency with what was used in label expansion, here the n-grams also range from n=1 to n=3. With $s^*_{j,i}$, we denote s^*_j :

[4] We use the `deepset/sentence_bert` breakpoint from Huggingface model hub [18,24].

[5] The similarity threshold was chosen through preliminary experimentation on a another dataset.

$\{s^*_{j,i}\}|^n_{i=0}$ as the matching score of d_j with every label from $\mathcal{Y}$. To decide if d_j is assigned one or more labels, a threshold ϵ is defined. For single-label tasks, if the maximum value of s^*_j at index i is greater than ϵ, then y_i is assigned to d_j. For multi-label tasks, if the value of s^*_j at any index is larger than ϵ, then the label at that index is assigned to d_j. Thus only the examples achieving high matching scores (high-confidence) with the labels are likely to be pseudo-labelled.

At this point, we get a pseudo-labelled collection denoted as $\hat{\mathcal{D}} : \{(x_i, y_i)\}|^N_{i=1}$ where N is the number of pseudo-labelled examples. To ensure the quality of $\hat{\mathcal{D}}$, the threshold ϵ should be chosen carefully. It will generate poor quality pseudo-labelled data if it is too low, but will result in zero examples for some labels if it is too high. Since the matching score $s_{i,k}$ is normalised by Equation 2, we set ϵ to be $log_e n$. However, this value can lead to zero pseudo-labelled examples for insufficiently-expanded labels (e.g., their vocabularies contain few phrases: particularly common in class-imbalanced datasets). Hence, we reduce ϵ by half when the label with fewest expanded phrases has fewer than 10.

3.3 Pre-training and Self-training

With $\hat{\mathcal{D}}$ as the supervision data, the next step fits the model f to the target classification task by learning from the pseudo-labelled data. Instead of training from scratch, we use the pre-trained `bert-base-uncased` [3] as the base model (f^*_θ) and this can be replaced by other pre-trained language models easily [24]. To be specific, for a task $\mathcal{T}$, we add a classification head on top of the base model. The classification head takes the `[CLS]` output (denoted as $\boldsymbol{h}_{[\text{CLS}]}$) of the base model as input and outputs the probability distribution over all classes:

$$\begin{aligned} \boldsymbol{h} &= f^*_{\boldsymbol{\theta}}(x) \\ p(\mathcal{Y} \mid x) &= \sigma(\boldsymbol{W}\boldsymbol{h}_{[\text{CLS}]} + \boldsymbol{b}) \end{aligned} \tag{4}$$

where $\boldsymbol{W} \in \mathbb{R}^{h \times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$ are the task-specific trainable parameters and bias respectively and σ is the activation function (softmax if $\mathcal{T}$ is a single-label task, or sigmoid for a multi-label task). In model training, the base model parameters $\boldsymbol{\theta}$ are optimized along with $\boldsymbol{W}$ and $\boldsymbol{b}$ with respect to the following cross entropy loss (on the pseudo-labelled data):

$$\mathcal{L}_{pt} = -\sum_{i=0}^{n} y_i \log p(y_i \mid x) \tag{5}$$

Since $\hat{\mathcal{D}}$ is a subset of $\mathcal{D}$, there are many unlabelled examples not seen in the model's pre-training. As indicated in LOTClass [13], the unlabelled examples can be leveraged to refine the model for better generalisation via **self-training**. Hence, we subsequently use $\mathcal{D}$ for model self-training. We first split $\mathcal{D}$ into equal-sized portions (assume each portion has M examples) and then let the model make predictions for each portion in an iteration with the predictions denoted as the target distribution Q. In each iteration, the model is trained on batches of the portion with the current distribution as P. The model is then updated with respect to the following KL divergence loss function [13]:

$$\mathcal{L}_{st} = \mathrm{KL}(Q \| P) = \sum_{i=1}^{M} \sum_{j=1}^{n} q_{i,j} \log \frac{q_{i,j}}{p_{i,j}} \tag{6}$$

In deriving the target distribution Q, it can be applied with either soft labeling [26] or hard labeling [8]. As soft labeling overall brings better results [13], we derive Q with the soft labeling strategy:

$$\begin{gathered} q_{i,j} = \frac{p_{i,j}^2 / p_j^*}{\sum_{j'} \left(p_{ij'}^2 / p_{j'}^* \right)}, p_j^* = \sum_i p_{i,j} \\ p_{i,j} = p(y_j \mid x_i) = \sigma(\boldsymbol{W}(f_{\boldsymbol{\theta}}^*(x_i))_{\text{[CLS]}} + \boldsymbol{b}) \end{gathered} \tag{7}$$

This strategy derives Q by squaring and normalising the current predictions P, which helps boost high-confidence predictions while reducing low-confidence predictions.

4 Experiments

Having described the system components, next we conduct extensive experiments to demonstrate its effectiveness. This section reports the experimental details the results.

4.1 Datasets

Table 1 lists four datasets chosen for evaluation: **Topic** [30], **Situation** [25], **UnifyEmotion** [7] and **TwiEmotion** [19]. The former three are the benchmarking datasets used in [28]. **Situation** is a multi-label dataset and only **Topic** is class-balanced. We also include **TwiEmotion**, which is another emotion dataset that does not overlap with **UnifyEmotion**. Overall, these datasets are varied in their domains, types (single-label or multi-label), class distributions (class-balanced or imbalanced) and label abstractness.[6] Following [28], we choose *label-wise weighted F1* as the primary evaluation metric for all datasets except for **Topic**, for which *accuracy* is reported, as it is class-balanced.

[6] Emotion labels like “joy”, “sadness” are more abstract than topic labels like “sports” and “politics & government.”

	Type	# Train	# Test	Class dist.	Avg len.	Label surface names
Topic	single	1300000	100000	balanced	107.6	{Education & Reference, Society & Culture, Sports, Entertainment & Music, Politics & Government, Computers & Internet, Family & Relationships, Science & Mathematics, Health, Business & Finance}
Situation	multi.	4921	1789	imbalanced	16.5	{shelter, search, water, utilities, terrorism, evacuation, regime change, food, medical, infrastructure, crime violence}
UnifyEmotion	single	34667	14000	imbalanced	47.5	{surprise, guilt, fear, anger, shame, love, disgust, sadness, joy}
TwiEmotion	single	16000	2000	imbalanced	19.2	{sadness, joy, love, anger, fear, surprise}

Table 1: Datasets used. Avg len. is the average number of words in training examples.

4.2 Experimental Details

As generalisation is crucial, we avoid our system being dependent on any specific target dataset. Thus a separate dataset [1] was initially used for exploratory experiments to investigate configuration options for our system (e.g. the value for ϵ, selection and pre-processing of phrases). This was done before the system was exposed to any examples from any of the target testing datasets.

The training set of each target dataset is used as the unlabelled corpus $\mathcal{D}$ and the surface names (see Table 1) as the label set $\mathcal{Y}$. For model pre-training on the pseudo-labelled data, we fine-tune `BERT-base-uncased` on batch size of 16 using Adam [6] as the optimiser with a linear warm-up scheduler for increasing the learning rate from 0 to $5e-5$ at the first 0.1 of total training steps and then decreasing to 0 for the remaining steps. For self-training, we follow the hyper-parameters used in [13], with batch size 128 and update interval 50, which results in the number of training examples in each iteration (namely, M) being 50×128. As guided by the average example length in Table 1, we set the maximum input length for model pre-training and self-training to be 256 for **Topic** and 128 for the remaining three datasets.

4.3 Baselines

In experiments, we compare our system to multiple baselines, described as follows.

Label similarity [20] uses pre-trained embeddings to compute the cosine similarity between the class label and every 1-gram to 3-gram of the example. For single-label tasks, the label with the highest similarity score is chosen. For multi-label tasks, any label with a similarity score greater than 0.5 is chosen.

Entail-single and **Entail-ensemble** correspond to the best *label-fully-unseen* ZSL single and ensemble results reported in [28]. As **TwiEmotion** was not used in that paper, we followed their methodology to create a similar setup by fine-tuning three variants of BERT on three inference datasets (RTE/MNLI/FEVER). We choose the best of these in each category to report as "entail-single" and "entail-ensemble".

ConWea is a contextualised weakly-supervised approach for text classification, which uses few human-provided seed words for label expansion [10]. In our experiments, we feed at least 3 seed words per class to this approach. As a comparison, weak human supervision (few seed words) entails this approach unlike ours using label names only.

X-Class [23] uses label names only by building class-oriented document representations first and then using GMM to obtain the pseudo-labelled data. In our experiments, we use the pseudo-labelled data by X-Class for model fine-tuning (`bert-base-uncased`) and report the performance on the fine-tuned model.

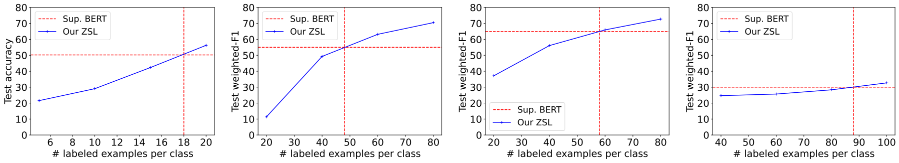
WeSTClass is the system proposed in [12]. Although it is configurable to accept up to three sources of supervision, we run this system using label names as the only supervision resource so to be consistent with our study.

LOTClass is another text classification approach using only label names [13].

Entail-Distil [2] attempts to overcome the inference inefficiency issue of the entailment matching-based models [27]. The training data is pseudo-labelled first by the matching model (`bert-base-uncased` fine-tuned on MNLI) and then the pseudo-labelled data is used for downstream model fine-tuning (`bert-base-uncased`).

Sup. BERT is included so that a comparison with a fully-supervised approach can be done. This uses `bert-base-uncased`, fine-tuned on the training sets. A ZSL approach will be unlikely to match the performance of a fully-supervised approach, it is important to illustrate how large that performance gap actually is and to illustrate the degree to which our approach contributes towards closing it.

4.4 Results and Discussion



(a) **Topic** (b) **Situation** (c) **TwiEmotion** (d) **UnifyEmotion**

Fig. 2: (a), (b), (c) and (d): The performance of P-ZSC is close to that of BERT trained on 18, 48, 58 and 88 labelled documents per class from **Topic**, **Situation**, **TwiEmotion**, and **UnifyEmotion** respectively.

In this section, we compare our system (P-ZSC) with the baselines in ZS text classification. We also dissect each component of P-ZSC by conducting an ablation study and investigate the quality of the pseudo-labelled data it generates.

Comparing to the baselines In comparing our system with the baselines (Table 2), P-ZSC substantially outperforms the semantic matching based runs including Label similarity, Entail-single and Entail-ensemble.

We found label similarity is a strong baseline. Although it is a simple matching between the sentence embeddings of label names and document phrases, interestingly, it outperforms the entailment runs for all datasets except **Topic**. This indicates that semantically matching a document's phrases with the label names can help determine the document's class. It should also be noted that the matching-based runs are around n (number of the labels) times slower than the classifier-based runs. Regarding the effectiveness of the classifier based runs, ours outperforms WeSTCLass and Entail-Distil. Entail-Distil achieves similar scores to ours on **UnifyEmotion** but the difference is substantially wider for the other datasets. Comparing to ConWea that uses seed words for label expansion, we find that our system outperforms it across the four datasets. For the more recent label-names-only approach X-Class, it performs well in the class-balanced dataset **Topic** while not in the rest of datasets as compared to our approach.

Likewise, it is interesting that LOTCLass performs well in **Topic** but exhibits poorer performance in the other datasets, suggesting that LOTClass does not generalise particularly well. By analysis, to expand labels, LOTClass identifies unlabelled examples with exact-word matches to label names. These are then expanded using BERT masked

	Topic	Situation	UnifyEmotion	TwiEmotion
Semantic matching based runs				
Label similarity [20]	34.62	40.75	26.21	56.03
Entail-single [28]	43.80	37.20	24.70	49.60
Entail-ensemble [28]	45.70	38.00	25.20	50.16
Classifier based runs				
ConWea [10]	49.81	25.91	21.39	47.34
X-Class [23]	48.12	39.27	15.19	42.21
WeSTClass [12]	34.96	28.40	15.45	22.54
LOTClass [13]	**52.07**	5.85	7.19	16.82
Entail-Distil [2]	44.47	37.85	29.43	48.87
P-ZSC	50.68	**55.02**	**30.22**	**64.47**
Sup. BERT [3]	74.86	85.27	40.10	92.02

Table 2: Performance of all methods on test sets of four datasets. Metrics shown are label-wise weighted F1 for all datasets except Topic, for which accuracy is used.

language modelling (MLM). Masked Category Prediction (MCP) is then used to pseudo-label the unlabelled examples at the token level. For some tasks, this works well since the label names (e.g. “education”, “sports”) are straightforward and usually have enough exact-word matches within unlabelled examples. Thus LOTClass performs well in the **Topic** dataset. However, for datasets like Situation or Emotion detection, classes such as “utilities” and “sadness” are more abstract and have a more unbalanced distribution, and are not contained directly in unlabelled examples. This leads to few examples in the label name data subsequently used by MLM and MCP. Thus LOTClass obtains relatively poor results for **Situation**, **UnifyEmotion** and **TwiEmotion**. As a comparison, our approach overall performs well across the datasets with different characteristics, indicating strong generalisability. Despite this, there is still a gap between our unsupervised runs and the supervised BERT run. This suggests that although our results indicate substantial progress, zero-shot (label-fully-unseen) text classification of multiple domains remains challenging and cannot yet be considered to be a solved problem.

	Topic	Situation	UnifyEmotion	TwiEmotion
P-ZSC	50.68	55.02	30.22	64.47
- Self-training	44.18	50.51	25.35	59.83
- PLA	49.83	46.20	29.86	56.35
- PLA + LE	46.33	43.26	20.97	48.59

Table 3: Ablation study. Metrics shown are label-wise weighted F1 for all datasets except Topic, for which accuracy is used.

Ablation study To examine the contribution of each component of our system, we conducted an ablation study, with the results reported in Table 3. This shows the performance of the entire P-ZSC system, and also separate results with the self-training step omitted, with the pseudo-label assignment (PLA) omitted and with both PLA and label expansion (LE) omitted. In each case, the removal of any phase results in a decline in performance for all datasets. Although this decrease is minor in some situations, the system performance suffers dramatically for at least one dataset in every case. This indicates that all phases are important to maintain peak effectiveness.

Pseudo-labelled data quality In our system, the only "supervision" resources for the downstream model training is from the pseudo-labelled data that is obtained via LE and PLA. In the pipeline of our system, pseudo-labelled data is obtained without any human supervision, using only the label names and an unlabelled corpus of the target task. Given the importance of the pseudo-labelled data in the final performance, we construct the pseudo-labelled data that is a subset of the unlabelled corpus (i.e., only the high-confidence ones are pseudo-labelled). To quantify the quality of pseudo-labelled data, we follow a similar methodology to [13]. We compare P-ZSC with the Sup. BERT run fine-tuned on different numbers of actually-labelled examples per class. From Figure 2, we notice that our pseudo-labelling can equal the performance of 18, 48 and 58 and 88 actually-labelled documents per class on **Topic**, **Situation**, **TwiEmotion** and **UnifyEmotion** respectively. This shows that there is some room to improve the pseudo-labelled data on datasets like **Topic**. This motivates us to explore adaptive PLA approaches on dataset characteristics (e.g., label abstractness) for generating better quality pseudo-labelled data for multi-aspect ZS text classification in the future.

5 Conclusion

Having identified drawbacks of existing ZS approaches, either in inference efficiency or in the classification of multiple domains, we have proposed a novel classifier-based approach that uses label expansion from the label names and pseudo-label assignment (PLA). Four datasets with different characteristics were selected, along with a number of benchmarks from recent state-of-the-art ZSL works. The experimental results show that our system (P-ZSC) can outperform the baselines and overall generalise well to zero-shot text classification of multiple domains. Although the pseudo-labelled data constructed by our system represents high quality in some aspects, there remains some room to improve it, such as combining with a few annotated examples in a semi-supervised learning fashion, PLA at the document level with confidence control and adaptive PLA approaches on more domain characteristics (e.g. label abstractness).

References

1. Alam, F., Qazi, U., Imran, M., Ofli, F.: Humaid: Human-annotated disaster incidents data from twitter with deep learning benchmarks. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 15, pp. 933–942 (2021)

2. Davison, J.: Zero-shot classifier distillation (2021), https://github.com/huggingface/transformers/tree/master/examples/research_projects/zero-shot-distillation
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
4. Karamanolakis, G., Mukherjee, S., Zheng, G., Hassan, A.: Self-training with weak supervision. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 845–863 (2021)
5. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1181, https://www.aclweb.org/anthology/D14-1181
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference for Learning Representations. San Diego (2015)
7. Klinger, R., et al.: An analysis of annotated corpora for emotion classification in text. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2104–2119 (2018)
8. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: In Workshop on challenges in representation learning, ICML (2013)
9. Levy, O., Seo, M., Choi, E., Zettlemoyer, L.: Zero-shot relation extraction via reading comprehension. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017). pp. 333–342. Association for Computational Linguistics, Vancouver, Canada (Aug 2017). https://doi.org/10.18653/v1/K17-1034, https://www.aclweb.org/anthology/K17-1034
10. Mekala, D., Shang, J.: Contextualized weak supervision for text classification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 323–333 (2020)
11. Mekala, D., Zhang, X., Shang, J.: Meta: Metadata-empowered weak supervision for text classification. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8351–8361 (2020)
12. Meng, Y., Shen, J., Zhang, C., Han, J.: Weakly-supervised neural text classification. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018). vol. 2018 (2018)
13. Meng, Y., Zhang, Y., Huang, J., Xiong, C., Ji, H., Zhang, C., Han, J.: Text classification using label names only: A language model self-training approach. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 9006–9017. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.724, https://www.aclweb.org/anthology/2020.emnlp-main.724
14. Müller, T., Pérez-Torró, G., Franco-Salvador, M.: Few-shot learning with siamese networks and label tuning. arXiv preprint arXiv:2203.14655 (2022)
15. Obamuyide, A., Vlachos, A.: Zero-shot relation classification as textual entailment. In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER). pp. 72–78 (2018)
16. Puri, R., Catanzaro, B.: Zero-shot text classification with generative language models. CoRR, abs/1912.10165 (2019)

17. Pushp, P.K., Srivastava, M.M.: Train once, test anywhere: Zero-shot learning for text classification. CoRR, abs/1712.05972 (2017)
18. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3973–3983 (2019)
19. Saravia, E., Liu, H.C.T., Huang, Y.H., Wu, J., Chen, Y.S.: Carer: Contextualized affect representations for emotion recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3687–3697 (2018)
20. Veeranna, S.P., Nam, J., Mencıa, E.L., Fürnkranz, J.: Using semantic similarity for multi-label zero-shot classification of text documents. In: Proceeding of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium: Elsevier. pp. 423–428 (2016)
21. Wang, C., Lillis, D.: A Comparative Study on Word Embeddings in Deep Learning for Text Classification. In: Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval (NLPIR 2020). Seoul, South Korea (December 2020). https://doi.org/10.1145/3443279.3443304
22. Wang, C., Nulty, P., Lillis, D.: Transformer-based Multi-task Learning for Disaster Tweet Categorisation. In: Adrot, A., Grace, R., Moore, K., Zobel, C.W. (eds.) ISCRAM 2021 Conference Proceedings – 18th International Conference on Information Systems for Crisis Response and Management. pp. 705–718. Virginia Tech, Blacksburg, VA (USA) (2021)
23. Wang, Z., Mekala, D., Shang, J.: X-class: Text classification with extremely weak supervision. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3043–3053 (2021)
24. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020). https://doi.org/10.18653/v1/2020.emnlp-demos.6, https://www.aclweb.org/anthology/2020.emnlp-demos.6
25. Xia, C., Zhang, C., Yan, X., Chang, Y., Philip, S.Y.: Zero-shot user intent detection via capsule neural networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3090–3099 (2018)
26. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International conference on machine learning. pp. 478–487. PMLR (2016)
27. Ye, Z., Geng, Y., Chen, J., Chen, J., Xu, X., Zheng, S., Wang, F., Zhang, J., Chen, H.: Zero-shot text classification via reinforced self-training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 3014–3024 (2020)
28. Yin, W., Hay, J., Roth, D.: Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3905–3914 (2019)
29. Zhang, J., Lertvittayakumjorn, P., Guo, Y.: Integrating semantic knowledge to tackle zero-shot text classification. 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). pp. 1031–1040 (2019)
30. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1. pp. 649–657 (2015)