Learn To Adapt

Shrushti Gupta
G01394828
Computer Science Department
George Mason University
sgupta32@gmu.edu

Sai Dharshini Akula
G01395223
Computer Science Department
George Mason University
sakula8@gmu.edu

1 Introduction

1.1 Task / Research Question Description

The paper aims to address the challenge of generalized zero-shot text classification. The research question being answered is how to develop an approach to classify textual instances from previously seen and incrementally emerging unseen classes. Specifically, the paper (Zhang et al., 2022) proposes a Learn to Adapt (LTA) network that explicitly learns over multiple learning episodes that mimic the generalized zero-shot learning setting during training, making the learning setting consistent with the test environment and improving generalization. The authors evaluate their proposed approach on five benchmark datasets and compare it with other state-of-the-art methods.

1.2 Importance of the Task

The problem of generalized zero-shot text classification is significant because it seeks to categorize textual occurrences from previously seen classes as well as incrementally emerging unknown classes. Most previous approaches fail to generalize because the learnt parameters are only optimal for seen classes and not unseen classes, and the parameters remain stable in prediction operations. The proposed Learn to Adapt (LTA) network (Zhang et al., 2022) solves these issues by learning across numerous learning episodes that explicitly replicate GZSL settings during training, making the learning setting consistent with the test environment and enhancing generalization.

This method could be used in real-world circumstances where new classes evolve over time, such as natural language processing tasks or online content monitoring. For example, LTA can be employed in natural language processing for text classification tasks such as sentiment analysis, topic categorization, and spam identification. LTA can also be used in online content moderation to categorize user-generated content into several categories, such as hate speech, cyberbullying, and fake news.

1.3 Motivation and Limitations of existing work

Prior work has attempted to address the challenge of generalized zero-shot text classification. The paper cites several related works that have proposed various methods for this task, such as attribute-based methods, generative models, and embedding-based methods.

Some models (Xian et al., 2019) learn the best parameters by minimizing the loss of instances from observed classes without explicitly calibrating the predictions on unseen classes. As a result, the issue of domain bias toward seen classes emerges.

Other works (Liu et al., 2019) consider the inter-class interaction while building prototypes for unseen classes; the models remain constant regardless of any additional classes introduced in the future. Hence, these models demonstrate a significant quality disparity across examples from seen classes.

The authors of (Zhang et al., 2022) argue that their proposed LTA network can overcome some of these limitations by leveraging meta-learning techniques and adapting to new classes at test time. Overall, the paper contributes an untried approach to generalized zero-shot text classification and comprehensively evaluates its performance on several benchmark datasets.

1.4 Proposed Approach

A meta-learning method for categorizing textual occurrences from both available classes and newly emergent classes is the Learn to Adapt (LTA) network. A feature extractor plus an adaptive classifier make up this system. The feature extractor converts the input text into fixed-dimensional embeddings, and the adaptive classifier predicts class labels. Using both real and fictitious unseen classes, the LTA network (Zhang et al., 2022) simulates a generalized zero-shot learning scenario during training. In order to adjust to incoming, unknown classes, it learns to calibrate class prototypes and sample representations. The LTA network generalizes well to new classes it has not encountered during training by adjusting its parameters and representations based on its prior experience at test time.

1.5 Summary of Results

The proposed approach (Zhang et al., 2022) performs well on the CLINC dataset based on the results achieved experimentally. The base results suggest that using different approach components, such as initialization, self-attention, and attention mechanism, significantly impacts the model's performance. The approach achieved high accuracy, F1-score, and harmonic mean on both seen and unseen classes than previous state-of-the-art models.The reproduction results show that the approach's performance is consistent across different settings and architectures.

However, the results obtained for the robustness evaluation suggest that the model is sensitive to variations in the dataset. Introducing typos and negations to the dataset negatively impacted the model's performance, highlighting the need for additional robustness testing.

The multilingual evaluation results suggest that the approach performs well in a multilingual setting. However, the model's performance decreased compared to the base results when evaluating datasets with multiple languages. The results imply that further research is required to enhance the technique's performance in multilingual contexts.

Overall, the findings demonstrate the promise of the proposed model LTA as well as the need for additional research to address the study's shortcomings.

2 Approach

The Learn to Adapt (LTA) network (Zhang et al., 2022) is a meta-learning approach that addresses the challenge of classifying textual instances from previously seen and incrementally emerging unseen classes. The task of generalized zero-shot text classification is challenging because the model needs to generalize sufficiently to new classes that it has yet to see during training. This can be difficult due to these classes' lack of labeled data. Additionally, the model needs to be able to adapt its parameters and representations to new classes at test time, which requires a flexible and adaptive learning approach.

The LTA network consists of two main components: a feature extractor and an adaptive classifier.

1. Feature Extractor: The feature extractor encodes the input text into a fixed-length vector representation that the adaptive classifier can utilize for classification. Each word in a sentence is encoded into a vector representation by a word-level encoder. These word embeddings are then processed by a Transformer Encoder Layer to produce contextualized word embeddings. Finally, mean pooling is used to these vectors to create a fixed-length sentence representation.

2. Adaptive Classifier: The adaptive classifier uses feature representation to predict the class label of an input text. In order to simulate a generalized zero-shot learning (GZSL) scenario according to the test time, LTA trains an adaptive classifier using both seen and virtual unseen classes. At the same time, it simultaneously learns to calibrate the class prototypes and sample representations to make the learned parameters adaptable to incoming unseen classes.

A BERT encodes a textual input x with T words into a sequence of hidden vectors H. Then, the text embedding is achieved by averaging the T hidden vectors.

During training, the LTA network learns over multiple learning episodes that explicitly

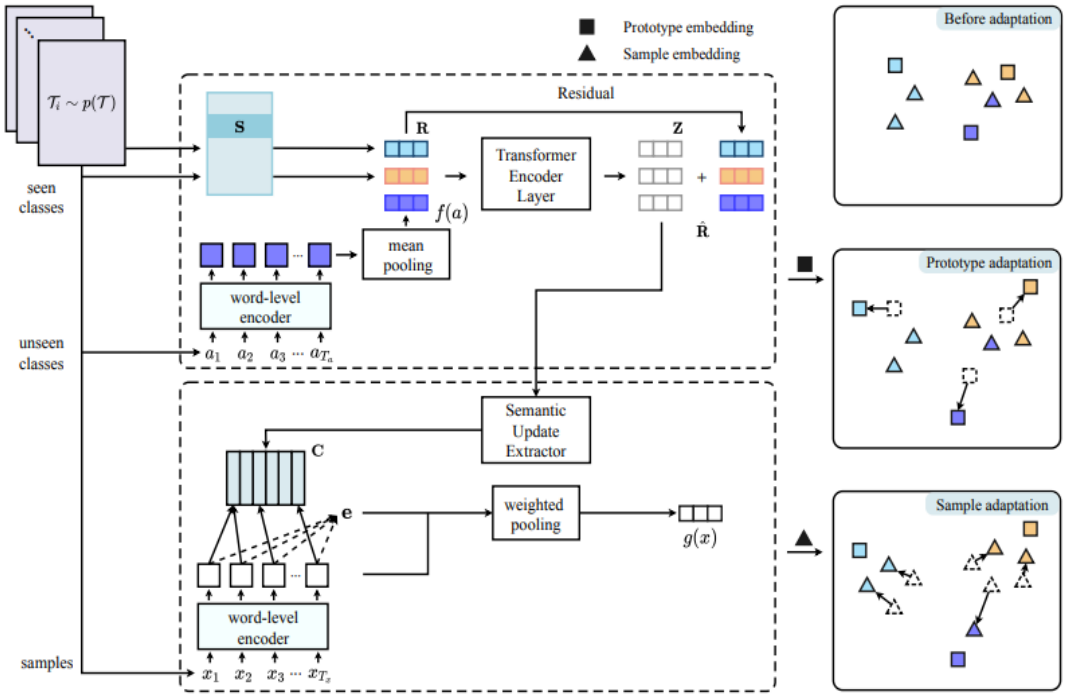

Figure 1: The LTA framework is depicted in the diagram above (Zhang et al., 2022). The right section indicates the prototype adaptation and sample adaptation. The dotted and solid borders demonstrate before and after adaption, respectively.

mimic the generalized zero-shot learning setting. First, a pre-trained and learnable look-up table is introduced, which stores embeddings of the seen prototypes and retrieves the "fake" seen classes from the look-up table. Using "fake" unseen class descriptions, a BERT encoder encodes the "fake" unseen prototypes into a matrix.

Specifically, it uses both seen and virtual unseen classes to simulate a generalized zero-shot learning scenario per the test time. The model simultaneously learns to calibrate the class prototypes and sample representations to adapt the learned parameters to incoming unseen classes. Given a new textual instance from an unseen class at test time, the LTA network adapts its parameters and representations based on its previous experience with similar classes during training. This allows it to generalize well to new classes it has not seen during training.

The LTA network extends its ability from two views: prototype adaptation and sample adaptation.

1. Prototype adaptation: It involves adapting the class prototypes based on their similarity to other prototypes in the embedding space. Transformer uses flexible self-attentions to capture the inter-class relationship between seen and unseen classes, which is advantageous for developing globally discriminative prototypes. The model can quickly represent and distinguish between the newly introduced classes thanks to the prototype adaptations, which simultaneously update both seen and unseen classes. This step will lessen the variance brought on by the sampling episode sequences.

2. Sample adaptation: It involves adapting individual samples based on their similarity to other samples in their respective classes. The zero-shot learning tasks are likely to result in semantics loss, where some features would be eliminated during training if they were essential for identifying classes that had not yet been seen but were nonetheless discriminating for those that had. Due to the significant imbalance between seen and unseen classes, a similar issue is made worse in the GZSL assignment. The issue is addressed by incorporating sample adaptation, which uses semantic components to direct the adaptation of sample

embeddings.

3 Experiments

3.1 Datasets

The LTA model (Zhang et al., 2022) uses five datasets. SNIPS-SLU, SMP-18, ATIS, and CLINC are intent classification datasets. In addition, the Quora Question Pairs dataset is a question classification dataset. The paper randomly chooses 70 percent of samples from each seen class to serve as the training set, 30 percent as the seen test, and all samples from unseen classes to serve as the unseen test. These datasets, along with the splits, are publicly available in the GitHub repository of the research paper. See https: https://github.com/Quareia/LTA.

Table 1: Dataset Description

Dataset	Classes		Samples	
	seen	unseen	total	avg
SNIPS-SLU	5	2	13802	1384
SMP-18	24	6	2460	60
ATIS	12	5	4972	245
CLINC	120	30	22500	105
Quora	1360	340	17394	7

CLINC dataset is used in the reproducibility code. The dataset contains various requests or queries from several domains, including travel, business, food, music, and others. For example, some inquiries include credit card information, bank accounts, and bill payments, while others concern airline schedules, reservations, and luggage tracking. There are also inquiries about playlist management and language settings. There are also questions about general knowledge, such as trivia and the meaning of life. The CLINC dataset has five files:

1. train_seen:It contains training data with the columns "Text" and "Label". "Text" refers to the sentences in string format, while "Label" refers to the associated label of the sentences in integer format. This label represents which seen class the corresponding sentence belongs to. There are 12600 instances in this dataset.

2. seen_class: It contains information about seen classes with the columns "Text" and "Label". "Text" refers to the sentences in string format, while "Label" refers to the associated label of the sentences in integer format. There are 120 seen classes labeled from 0 to 199.

3. test_seen: It contains testing data with the columns "Text" and "Label". "Text" refers to the sentences in string format, while "Label" refers to the associated label of the sentences in integer format. This label represents which seen class the corresponding sentence belongs to. There are 5400 instances in this dataset.

4. test_unseen: It contains testing data with the columns "Text" and "Label". "Text" refers to the sentences in string format, while "Label" refers to the associated label of the sentences in integer format. This label represents which unseen class the corresponding sentence belongs to. There are 4500 instances in this dataset.

5. unseen_class: It contains information about unseen classes with the columns "Text" and "Label". "Text" refers to the sentences in string format, while "Label" refers to the associated label of the sentences in integer format. There are 30 unseen classes labeled from 120 to 149.

All these datasets are in CSV format, then converted into a pickle file for the model to read.

Table 2: CLINC Dataset Description

CSV Files	Rows	Columns	Class Label Range
train_seen	12600	2	
seen_class	120	2	0 to 199
test_seen	5400	2	
test_unseen	4500	2	
unseen_class	30	2	120 to 149

Robustness:

As the paper (Ribeiro et al., 2020) recommends, the perturb function from the checklist library is used to make minor changes to the original text. It enables the application of numerous perturbation operations to input texts, such as altering words, shifting sentences, adding noise, or inserting grammatical errors. By generating perturbed variants of the text and analyzing how the model performs on these variations, one can learn about the model's sensitivity to various alterations and potential vulnerabilities.

First, typos are introduced into the dataset. Second, the dataset is preprocessed with spacy,

then negations are added. Finally, a function for substituting human names in the dataset was implemented, but it did not affect the model's sensitivity; therefore, it was not employed. Only the train_seen dataset was updated here, while the rest remained unchanged.

Multilinguality:
Because the research (Zhang et al., 2022) is only in English, adding more languages will evaluate the model's performance in different languages and uncover potential language-specific problems or biases. This aids in understanding the model's cross-lingual generalization capabilities.

The current train_seen dataset is being supplemented with four more languages: Chinese, Hindi, Telugu, and German. This is accomplished by converting the existing dataset using the procedures described in Homework 2. The translator function from the googletrans library is employed to generate the new dataset.

3.2 Baseline Methods

To validate the benefits of the proposed LTA approach (Zhang et al., 2022), it is compared against several other methods in three aspects:

1. Supervised Learning Methods: The performance of seen classes is evaluated using supervised learning instead of the Generalized Zero-Shot Learning (GZSL) setting. Specifically, a linear Softmax classifier uses the BiLSTM (Schuster and Paliwal, 1997) and BERT (Devlin et al., 2018) encoders.

2. Metric Learning Methods: Three metric-based embedding methods are commonly used as baselines for GZSL. These methods are:

- EucSoftmax (Snell et al., 2017): Uses squared Euclidean distance as the metric and Softmax classification.
- Zero-shot DNN (Kumar et al., 2017): Uses squared Euclidean distance and triplet loss to maintain a margin for different classes.
- CosT (Gidaris and Komodakis, 2018): Uses cosine distance as the metric with a learnable temperature scalar.

3. SOTA Methods: The LTA model is compared against two recent state-of-the-art (SOTA) methods:

- ReCapsNet (Liu et al., 2019): It employs a dimensional attention-based intent capsule network and a matrix transformation method for GZSL.
- SEG (Yan et al., 2020): This outlier detection approach can be directly applied to ReCapsNet. SEG first determines whether a test sample belongs to seen or unseen classes and then classifies it within their respective domains.

Table 3: Comparing Baseline Methods and reproduced results on CLINC dataset

Baseline Methods				
Model	Seen Acc	Seen F1	Unseen Acc	Unseen F1
Bi-LSTM	92.07	92.06	0.00	0.00
BERT	97.37	97.37	0.00	0.00
EucSoftmax	96.02	87.07	58.02	66.00
Zero-shot	95.31	86.65	58.49	65.89
CosT	96.31	87.33	62.73	70.28
ReCapsNet	88.53	69.83	4.24	3.33
+ SEG	81.04	78.89	9.07	5.44
Reproduction				
LTA	Seen Acc	Seen F1	Unseen Acc	Unseen F1
w / o Init	93.13	88.04	70.75	69.88
w / o SA	86.89	83.68	59.89	61.18
w / o A	92.77	86.78	69.34	72.54

3.3 Implementation

The re-implementation code can be accessed here: https://github.com/Shrushti1999/Learn-To-Adapt.git

3.4 Evaluation Metrics

The metrics used to evaluate the model include accuracy, F1-score, harmonic mean, precision, and recall. These metrics are commonly used for assessing the effectiveness of machine learning models and are frequently employed in tasks involving natural language processing. Precisely, accuracy measures the proportion of correctly classified samples, F1-score evaluates the trade-off between precision and recall, the harmonic mean is the weighted average of precision and

recall, precision evaluates the proportion of correctly identified positive samples, and recall evaluates the proportion of actual positive samples that were correctly identified. These metrics allow for assessing the model's effectiveness and identifying potential issues.

3.5 Results

Table 4: Comparing published and reproduced results on CLINC Dataset

Published Results				
LTA	Seen Acc	Seen F1	Unseen Acc	Unseen F1
w / o Init	93.07	88.19	73.80	77.54
w / o SA	92.46	87.30	69.27	73.26
w / o A	93.81	88.12	70.11	74.58
Reproduction				
LTA	Seen Acc	Seen F1	Unseen Acc	Unseen F1
w / o Init	93.13	88.04	70.75	69.88
w / o SA	86.89	83.68	59.89	61.18
w / o A	92.77	86.78	69.34	72.54

As depicted in Table 4, the reproduced results differ slightly from the published ones, which might occur due to hardware and software changes.

Table 5: Reproduced results for Robustness

LTA	Seen Acc	Seen F1	Unseen Acc	Unseen F1
w / o Init	30.20	26.26	13.06	12.54
w / o SA	34.18	32.14	12.50	11.79
w / o A	30.20	26.25	08.33	08.24

The reproduced results on the robustness of the LTA model shown in Table 5 are due to the introduced typos and the negations in the dataset. These modifications made the data more noisy and inconsistent, making it more challenging for the model to identify inputs correctly. These adjustments caused discrepancies between the training and test data sets, which harmed the model's performance and resilience.

Table 6: Reproduced results for Multilinguality

LTA	Seen Acc	Seen F1	Unseen Acc	Unseen F1
w / o Init	65.75	63.99	55.95	54.29
w / o SA	63.82	62.03	51.45	52.55
w / o A	64.02	62.27	51.09	53.87

The reproduced results on the multilinguality of the LTA model shown in Table 6 have undergone a loss of accuracy. There are several causes for the accuracy loss while switching from a monolingual to a multilingual dataset. The performance in other languages might be much poorer for language models trained on monolingual datasets since they are frequently optimized for a few languages. The multilingual dataset's additional languages may have yet to teach the language models how to accurately represent their linguistic quirks and subtleties. Moreover, the decrease in accuracy in multilingual cases can be due to a combination of factors, including language-specific characteristics, differences in data distributions, and insufficient training data.

3.6 Discussion

According to the owner of the research paper, the BERT configurations and other datasets have been lost from the GitHub repository . The model only accepts pickle files as input. Since only the CLINC pickle file was available, the model could only be tested on that dataset. Furthermore, the code did not work in the first few runs. It ran after some changes and commenting a few lines, but the output labels were not saved anywhere (variable or file). Accessing those output labels took a significant amount of time.

Table 7: Sensitivity Analysis with different hyperparameters

tau	8.0	8.0	8.0	8.0
lr	1e-2	1e-2	1e-4	1e-4
iterations	20	20	30	30
hidden_size	64	32	32	32
d_r	64	32	32	32
train_batch_size	64	32	32	32
histogram	False	False	True	True
step_size	500	500	500	400
gamma	0.1	0.1	0.1	0.2
Best performance	42.67	44.45	63.97	63.97

After conducting a sensitivity analysis by running multiple experiments with different random seeds or hyperparameters settings to assess the robustness of the proposed method, as portrayed in Table 7, it was concluded that the hyperparameters suggested by the research paper were the best.

3.7 Resources

A significant amount of time was invested in understanding the research paper and also the code given by the authors. It was necessary to ensure that any changes to the code did not result in any implementation changes. The paper was reimplemented over the course of two weeks.

The computation time was considerably low for the model, nevertheless, ORC sent an email warning that the processes might be killed due to excessive node load.

Overall, reproducing research results required careful attention to detail and a thorough understanding of the proposed method and experimental setup.

3.8 Error Analysis

The authors do not provide any direct error analysis. Evaluation metrics like Accuracy and Micro-F1 are presented, and Harmonic Mean is calculated for seen and unseen classes.

Three LTA variations were evaluated by the authors in order to better understand the role that each component of the approach plays. The "w / o Init" model randomly initializes adapted prototypes R instead of pretrained prototypes. The model that solely employs prototype adaptation without "sample adaptation" is referred to as "w/o SA." "w / o A" denotes the absence of any adaptation phase. As is evident, LTA that includes both prototype adaptation and sample adaptation outperforms LTA that does not include any adaptation steps.

The authors could have performed additional error analyses, such as analyzing misclassified instances to identify common patterns or characteristics contributing to errors.

The error analysis tables 8-11 show the results of various tests performed on the system.

Table 8: Minimum Functionality Test (MFT)

Test cases	814
Fails	425
Fail rate percentage	52.2
Example fails	-does have good reviews have great reviews -you sound like a bot -do they serve good tacos at the buffet

Table 8, which presents the Minimum Functionality Test (MFT), shows that over half of the 814 test cases failed. The 814 sentences were classified into positive and negative sentences with the help of specific keywords ("good" or "like" for positive and "bad" or "awful" for negative). The examples that failed were related to sentiment analysis, especially negation. The predictions for failures that should be negative are positive, and vice versa.

Table 9: Invariance Test for Typos

Test cases	12600
Fails	324
Fail rate percentage	2.6
Example fails	-can yous top -whatk ind of petrol goes in the tank -that isn ot right

In Table 9, the Invariance Test for Typos, the system was tested for its ability to handle typos, resulting in a much lower failure rate of 2.6 percent. The failed cases were due to spelling mistakes, and the system struggled to grasp the intended meaning when typos were present.

Table 10: Invariance Test for Negations

Test cases	11692
Fails	56
Fail rate percentage	0.5
Example fails	-i'm not afraid i've forgotten the pin for my 401k account -incorrect, it is not certainly a false statement -i'm not afraid i don't know how to answer that

The system's performance was better in Table 10, the Invariance Test for Negations, where the system's failure rate was only 0.5 percent when the input contained negations. Table 10 demonstrates that the system struggled with negations

since it could not detect the proper sentiment when negations were present.

Table 11: Invariance Test for Name changes

Test cases	178
Fails	0
Fail rate percentage	0.0

Finally, Table 11, the Invariance Test for Name changes, showed that the system was able to handle name changes without any failures.

Overall, the error analysis indicates that the system may need further improvement to handle complicated language patterns. Future work should enhance the system's capacity to handle multiple inputs.

4 Related Work

4.1 Zero-shot learning

The earliest ZSL studies (Frome et al., 2013; Zhu et al., 2019; Xia et al., 2018; Nam et al., 2016) try to learn a matching model between instance embedding and class prototype embeddings that are represented by additional data, such as class-level attributes, text descriptions, or a combination of both. However, since these models were developed using data from observed classes, they are unable to adjust gradually to the emergence of new classes.

Other studies (Xian et al., 2018; Schonfeld et al., 2019; Song et al., 2021; Zhang et al., 2019) try to employ generative models to create virtual samples for unseen domains to optimize the model for unseen classes, assuming that additional knowledge about unseen classes is provided. Unfortunately, due to the fact that often neither the test data nor their label descriptions are available during the training phase, these assumptions are not very realistic.

In contrast, the proposed model LTA (Zhang et al., 2022) can incorporate all classes—both seen and unseen—jointly during inference; in essence, it is taught to continuously generalize for new classes, making it capable of dynamically adapting to newly introduced classes.

4.2 Episode-based training

The proposed method (Zhang et al., 2022) primarily relies on the episodic training paradigm, which is successfully used to few-shot learning (FSL).

Weight generators are used in (Gidaris and Komodakis, 2018) to update unseen prototypes in generalized FSL (GFSL). (Ye et al., 2021) and (Shi et al., 2019) use graph neural networks in GFSL and attention mechanisms to update both seen and unseen prototypes. These methods solely consider the prototype adaption; the sample embeddings remain static regardless of the unknown classes.

In contrast, by adapting prototypes and sample embeddings, the proposed method (Zhang et al., 2022) simultaneously addresses knowledge transferring and domain bias in an episodic training framework. It generates a rapid adaptation to the novel classes without incurring significant damage in discriminating the seen classes.

5 Conclusion and Future Work

The paper (Zhang et al., 2022) is partially reproducible because of the lack of pickle-type datasets. The GitHub repository is poorly organized, as many files were in the wrong folders, leading to 'ModuleNotFoundError.' The paper provides detailed information on the proposed method, experimental setup, and evaluation metrics used in the experiments. It also references publicly available datasets and briefly describes each dataset. However, the paper does not provide information on train/dev/test splits used for these datasets.

The reproducibility presents extensive experiments on CLINC datasets, including invariance tests for typos, negations, and name changes. These modifications made the data more noisy and inconsistent, which harmed the model's performance and resilience. The model is sensitive to variations in the dataset and requires additional robustness testing.

Accuracy has been lost in the reproduced results for the LTA model's multilingualism. Additionally, the decline in accuracy in multilingual scenarios may be brought on by several elements, such as language-specific traits, variations in data

distributions, and a lack of training data.

The paper (Zhang et al., 2022) presents a promising generalized zero-shot text classification approach and provides insights into its strengths and limitations. The proposed LTA network has shown significant performance improvements compared to other state-of-the-art methods, but further research is required to enhance its robustness and multilingual contexts.

The paper (Zhang et al., 2022) proposes future directions, including exploring advanced outlier detection methods for improved performance of GZSL models, investigating sophisticated metric learning methods to capture complex relationships between text samples and classes, exploring other types of semantic information, and extending their approach to related tasks.

One feasible concrete next step for the project is to investigate cross-lingual transfer learning and zero-shot learning methods to improve the model's performance on multilingual datasets. This could involve investigating different pre-training techniques, such as multilingual pre-training, to leverage knowledge from multiple languages and improve cross-lingual transferability. Other metrics, such as language-specific and similarity metrics, can be used better to evaluate the model's performance on multilingual datasets.

Using more flexible data can be a concrete next step in enhancing the robustness performance of the project. This could involve incorporating data from different domains or using diverse samples within the same domain to improve the model's ability to handle different types of inputs. Additionally, more advanced outlier detection methods could be explored to handle complex data distributions better and improve the robustness of the model.

References

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

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26.

Spyros Gidaris and Nikos Komodakis. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375.

Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Bjorn Hoffmeister. 2017. Zero-shot learning across heterogeneous overlapping domains.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert YS Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. 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)*, pages 4799–4809.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in text: Learning document, label, and word representations jointly. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. 2019. Relational generalized few-shot learning. *arXiv preprint arXiv:1907.09557*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Congzheng Song, Shanghang Zhang, Najmeh Sadoughi, Pengtao Xie, and Eric Xing. 2021. Generalized zero-shot text classification for icd coding. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4018–4024.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2018. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*.

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265.

Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert YS Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1050–1060.

Han-Jia Ye, Hexiang Hu, and De-Chuan Zhan. 2021. Learning adaptive classifiers synthesis for generalized few-shot learning. *International Journal of Computer Vision*, 129:1930–1953.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. *arXiv preprint arXiv:1903.12626*.

Yiwen Zhang, Caixia Yuan, Xiaojie Wang, Ziwei Bai, and Yongbin Liu. 2022. Learn to adapt for generalized zero-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 517–527.

Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. 2019. Generalized zero-shot recognition based on visually semantic embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2995–3003.