Group 11: Few Shot Natural Language Inference using MAML for Indian Languages

Boppana Tej Kiran Shivam Aggarwal Naga Durga Krishna Mohan Eaty
Aman Aryan Aman Singh
20111070, 20111058, 20111037, 20111009, 20111010
{tejkiranb20, shivama20, kmohan20, ryn20, amansingh20}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

Today the success of Deep learning models in various domains is well known, but one problem associated with these models is that they require a massive amount of data. In NLP, most of the data available are in the English language, which makes the task of directly applying deep learning models to other low-resource languages very challenging. In this work, we proposed and implemented a cross-lingual model that uses MAML and TL. On a high level, the idea is to take a pre-trained BERT (Indic BERT) [1], which was pre-trained on multiple Indian languages and add a classification layer on top of BERT and fine-tune the resulting model by meta training on three Indian language NLI(Natural Language Inference) data sets and perform prediction on target Language NLI data set which the model did not see at least once. Although we could not make the model learn the task, we have learned few valuable insights from the experimentation that can be extended to future work.

1 Introduction

Natural Language Processing refers to set of techniques that aim to make computers Process and Generate natural languages. Processing is the ability of computers to process natural text, and Generation is predicting the next word tokens given a context (set for previous tokens). Initially, both these tasks were achieved by rule-based systems[2]. Later deep learning models with recurrent neural architectures such as LSTM and GRU and feed forward networks like Transformers are used. These models were able to serve the purpose and produce state-of-the-art results, but there are two main problems with these models.

The first problem is, these models need huge compute resources, training time and are data hungry. So, it is not feasible to train these models in all scenarios. Particularly when we don't have time needed to train them, large training data, or compute resources. Creating data sets and building compute resources cost a lot of money.The problem can be addressed by using Transfer Learning.
Lot of time and money can be saved if we could use a trained model directly for an NLP task by tweaking it a bit and not training it from scratch. That is what Transfer Learning (TL) does. Transfer Learning can viewed as "how humans try to learn from different related activities and apply that inference to new tasks intuitively."

The second problem is, since these models are trained only on a particular task and for a particular data set, they don't work well in real world setting where data can be very generic. This problem can be addressed by using Model Agnostic Meta-learning (MAML) techniques while training our model. The MAML [3] is an optimization-based Meta-learning technique. Meta-Learning is about "Learning how to learn" and model-agnostic means it is compatible with any model based on the Gradient Descent algorithm. The main objective of the MAML is to train the parameters of the model such that we can get good results with very few examples in a short time for any new task.

In our work we want to use both the Transfer learning (TL) and Model Agnostic Meta-learning (MAML) techniques to address one of the main problems in NLP research which is application of deep-learning models for low resource languages. There is lack of annotated text corpora in many languages. Currently, all the NLP data sets are concentrated around 4-5 languages like English, German, Chinese etc. If we take Hindi and other Indian native languages like Telugu, Bengali there are any hardly any annotated data sets. So for doing NLP tasks on these languages cross-lingual learning is the go to option. Cross-lingual learning means training our model on one language and making predictions for another language. In this paper we propose a cross-lingual technique which involves the above two techniques MAML and TL.

There are many pre-trained models like BERT, XLM, RoBERTa, etc. available in open source libraries like Hugging Face. These pre-trained models already contain a lot of information about natural languages encoded in them. We can add some hidden layers on top of these models and depending on use case and fine-tune them. We have chosen NLI for Indian Languages as our use case. NLI task is a classification task, in which given a sentence pair (Context & Hypothesis), the model need to classify the pair into one of the two labels "Entailed" or "not-Entailed". The idea is to use the pre-trained model called Indic BERT provided by Hugging Face library which was trained on many Indian languages like Hindi, Telugu, Bengali, and add a classification layer on top of the Indic BERT. The resulting model is then meta-trained on multiple tasks, where the individual task is to train the model on NLI task for three different Indian languages. We have chosen Gujarati, Bengali, Hindi languages for this purpose. Once the model is meta-trained on these NLI tasks, we will use this meta trained model to make prediction on NLI dataset of the target language which it has not seen before. In this approach we are using TL technique in terms of using a pre-trained model and MAML techniques in model's training procedure. The NLI dataset required is fetched from Midas repository [1] which contained Hindi NLI data from different source. However NLI data is not available for Gujarati, Bengali, Telugu languages. So, the Hindi NLI data is translated into the other three languages using Google Translate. The implementation and experimentation for the work is maintained in GITHUB repository - `https://github.com/aggarwal-shivam/NLP-Research-Project`

In subsequent sections, we discuss about the Related Work**(Section 2)**, Proposed Idea in detail **(Section 3)**, Dataset/Corpus **(Section4)**, Experiments, Results and Error Analysis in **(Section 5,6,7)**.

2 Related Work

The primary motivation of our project is to introduce meta-learning approaches to low resource Indian Languages. Few Shot classification is a significant problem related to Transfer Learning and Meta-Learning. Initially, the meta-learning algorithm mainly was focused on few-shot image classification problems. Some of these works include Siamese Net by [4] Koch et al., Matching Net by [5] Vinyals et al., Prototypical Network [6]. But these model were not flexible. The MAML algorithm proposed by [3] Finn et al. fixed the flexibility issue. Then MAML was adopted by several natural language processing tasks in low resource scenario such as Domain Adaptation META-MT [7]. Nooralahzadeh et al. [8] applies MAML for multilingual natural language tasks such as Natural Language Inference and Question Answering. XMAML is used on XNLI and MLQA data set for few-shot zero-shot cross-lingual learning. Zhan et al.[7] proposes a further improvement Meta Curriculum Learning. This paper uses a language curriculum on top of the MAML to introduce order in the learning procedure.

3 Proposed Idea

3.1 RESEARCH PROBLEM:

Modern NLP applications enjoyed great success utilizing neural network models in languages like English. This is not the case for most languages, especially low-resource ones with insufficient annotated

[1] `https://github.com/midas-research/hindi-nli-data`

training data. For Indian languages, this is a major issue. Only some of the 22 official Indian languages, which are a subset of the numerous languages spoken and written in India, have enough resources for training a deep learning model. Applying deep learning techniques for low resource languages is a major research problem in NLP. In our work, we want to address this problem by using cross-lingual learning. Cross-lingual learning can be done using Transfer learning and Model Agnostic Meta-Learning. We have chosen Natural Language Inference as our use case for our work. The idea is to take a pre-trained model and fine-tune it by meta training it on three of the four Indian Languages NLI data sets (Gujarati, Bengali, Hindi, Telugu) except the other language as target language NLI data set and perform predictions on the target language NLI data set.

In machine learning terminology, TL can be explained as given a source domain D_S and source task T_S as well as a target domain D_T and target task T_T. The objective of transfer learning (TL) is to learn the model for the target task T_T in the domain D_T with the information gained from D_S and T_S. The advantages that TL brings are, it allows for faster convergence on the target task, and the requirement of supervised annotated data is much less compared to training the model from scratch[2].

For MAML, an entire task T_i is equivalent to a single training example. This task T_i is sampled from a distribution of similar tasks $p(T)$. Training a model using MAML happens in two phases. The first phase is task-specific learning, and the second phase is meta-optimization across tasks.

The training procedure of MAML is as follows: First, we initialize model parameter θ randomly, then for each epoch sample K tasks $T_1, \ldots ...T_K$ from the task distribution $p(T)$. For each sampled task, train a copy of the model on training data D_i^{train} using the gradient descent algorithm.

$$\textbf{Task-specific learning: } \theta_i = \theta - \alpha \nabla_\theta L_i(\theta, D_i^{train}) \; \forall \; i \in [1, k]$$

In the meta-optimization phase, optimize the original parameter θ using the adapted parameters from task-specific learning after **s** gradient steps. In this phase, use the test data for each sampled task D_i^{test}).

$$\textbf{Meta-optimization phase: } \theta = \theta - \beta \nabla_\theta \sum_{I=1}^{K} L(\theta_i^s, D_i^{test})$$

Here α and β are the gradient-based update learning rates. This trained model can be used on any new task sampled from the same distribution $p(T)$. The primary motivation behind MAML is to establish a framework that can be used across various architectures and problem settings. MAML can be applied to classification, regression, policy gradient-based reinforcement learning with very slight changes. The primary application of MAML is in the field of zero-shot, one-shot, and few-shot learning.

3.2 MODEL ARCHITECTURE:

To perform NLI on Indian Languages, we use a pre-trained Bert model named Indic Bert, pre-trained in Indian Languages. We will use the MAML algorithm to meta-train this model on three Indian languages (Bengali, Gujarati, and Hindi). Finally, we will use the model to make predictions on the NLI Telugu dataset.

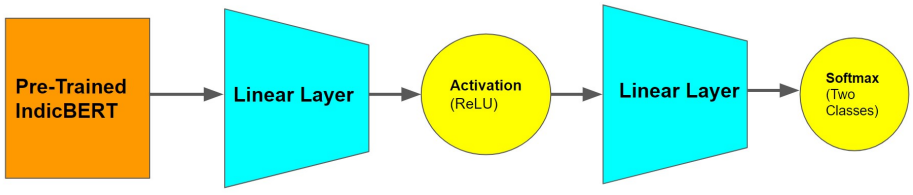

Figure 1: Model Architecture

Since NLI is a binary classification task, we can not directly use the Indic BERT architecture. We need to add a Linear layer on top of the Indic Bert model to perform the classification. BERT encodes

all the information in the starting token of the sequence for classification tasks, a dense representation of the size 768 (BERT hidden size). As shown in the Figure 1, we will feed this representation into a series of Linear layers with activation functions and dropout.

3.3 MODEL PIPELINE:

1. **Preprocessing:** The BERT expects us to feed both the context and hypothesis in the form of a sequence of tokens along with their 'attention mask' and 'segment ids.' The attention mask prevents the model from seeing pad indexes. Segment ids distinguish the context and Hypothesis part. So initially, to generate the sequence of tokens, we will concatenate both the context and Hypothesis sentences with a ['SEP'] token between them. Then we will add ['CLS'] token before the starting of 'Context' and another ['SEP'] token at the end of 'Hypothesis.' Then, we will use Indic BERT tokenizer to convert the sentence pair into tokens. Segment ids are generated by assigning value '0' in place of the context part of the sentence pair and value '1' in place of the hypothesis part of the pair. We will pass the sentence pairs in batches to the model, so we will pad the sentences with pad tokens up to a fixed length for every batch. To generate the attention mask, we will keep '0' in place of a pad token and '1' in other places.
2. **Meta-training Procedure:** As discussed above, we will meta-train the model on three tasks. We divide each task into 'support' and 'query' sets. These divisions are like train and development sets. The model learns with the help of 'support' tasks, and its performance is evaluated with the help of the 'query' tasks. Based on the performance on 'query' sets of three tasks (one from each language), the model's parameters are updated. We have kept 200 instances in each of the 'support' set and 100 instances in each of the 'query' set for our training.
3. **Few-shot learning procedure:** We have also trained the model for the few-shot NLI task. We will meta-train the model on three languages with a limited number of training instances (50 from each language) and will meta test on the fourth language using 990 instances.

4 Dataset/Corpus

One of the main challenges for Textual entailment evaluation for low-resource languages is the lack of annotated data. The corpus used is a recasted dataset for natural language inference in Hindi. We have used the BBC News dataset for our NLI task. The data corpus contains 15556 context and hypothesis pairs for 6 different classes like news, sport, science etc. For Meta training we need data in different Indian languages so we translated the Hindi corpus in three different Indian languages Gujarati, Bengali and Telugu. We have used google translator to translate given Hindi BBC News corpus to Gujarati, Bengali and Telugu. We have translated approximately 2K sentences for 5 different classes i.e. India, news, international, entertainment and sport.

Annotating data is expensive and requires language specific knowledge. In Figure 2 the original sentence and sentiment label from original data corpus is used for the recasting process. The original sentence (context) is appended with sentences belonging to all possible sentiment classes (hypothesis) to generate NLI data corpus with labels entailed and non-entailed depending upon the actual sentiment label from the original data corpus.

5 Experiments

Our objective was to investigate the effect of Meta-Learning on Cross-Lingual Natural Language Inference for Indian Sentences. We employed Indic BERT (Bidirectional Encoder Representations from Transformers) as a base model to capture the language representation for Indian Languages. Indic Bert is a multilingual ALBERT trained in 12 Indian Languages. It has fewer parameters compared to MultiBert and produces a comparable result. We used the torchmeta library for building a meta-model. MetaModule and Meta Layer provide us to build a model where we can use the model with custom parameters without applying it. We tried to use the various configuration of layers. In the

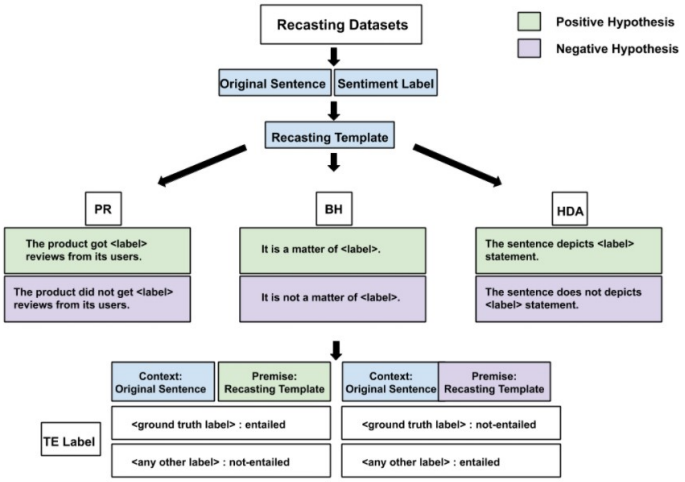

Figure 2: Illustration of recasting approach[9]

final model, we used two MetaLinear Layer and Softmax as the final activation layer. We freezed all the parameters of the BERT.

The training procedure we used was the MAML for meta learning. The model trained with MAML was used as initial model for target task. We used adam optimizer as meta optimizer with learning rate $\alpha = 1e-5$ and $\beta = 1e-6$ as mentioned in [8]. We used Cross Entropy loss function for calculating the loss. The MetaLinear layer in final model is xavier initialization with gain as 1.

The models were trained on Google Colab with minimum 11 GB GPU. It takes around one and a half hour for 150 epochs of meta train with 200 support and 100 query set per task. In each epoch there are three task. We experimented with various sizes of support, query and number of task. Then we used the sizes which were possible using the google colab GPU.

6 Results

This project is a binary classification task. Therefore, the evaluation metric used is accuracy. The various result obtained for various target task are tabulated below:

Target Language	Auxiliary Languages	Accuracy
Hindi	Gujrati, Bengali, Telugu	49.9
Gujrati	Hindi, Bengali, Telugu	50.5
Bengali	Gujrati, Hindi, Telugu	51.1
Telugu	Gujrati, Bengali, Hindi	49.5

Table 1: Few Shot Classification Accuracies

Target Language	Auxiliary Languages	Query Accuracy	Loss
Hindi	Gujrati, Bengali, Telugu	50.0	0.743
Gujrati	Hindi, Bengali, Telugu	50.0	0.726
Bengali	Gujrati, Hindi, Telugu	50.3	0.712
Telugu	Gujrati, Bengali, Hindi	50.0	0.728

Table 2: Meta Training Metrics

7 Error Analysis

1. We tried to implement MAML using Indic BERT for binary classification. The NLI Midas dataset contains the sentence in batches of 4 or 8, where the context is the same, and the hypothesis differs just by one word. On training the model using this dataset, the model immediately became biased toward the first example it faced during training. For all such pairs, the model is overfitting to the first example. We tried various learning rates ranging from 10 to 1e-10, initializations, increasing and decreasing the layers. A wrong initialization resulted in bad optima with 50% accuracy at the beginning only. A better initialization went to the bad optima after many epochs. We tried Multilingual Bert and Indic Bert(a smaller version of Multilingual Bert). We tried freezing the fraction of the layers of the BERT to increase the chance of the model distinguishing between sentences. We tried to implement a Matching Network (which compares the relation between two sentences to predict similarity), but it also faced the same problem. We implemented a simple artificial neural network consisting of few fully connected layer and activation function to observe the effect of this training data. Similar effects of overfitting to one class was observed.
2. The meta training procedure is an expensive computation process. It requires the losses of the query set to traverse through the computation of the support set. Therefore it needs to store the gradients during the training and the query loss for each task per epoch. Since google colab assign random GPUs every instance, we fixed the meta training batch size and number of task per epoch to adjust it to the 12 GB GPU configuration setting. Also, it was mandatory for meta training to freeze the layer of BERT, or else the GPU always went out of memory. The meta learning procedure always crashed when executed on CPU. The main limitation of the proposed model is its high demand for memory resources.

8 Individual Contribution

Name	Contribution in Survey	Project Implementation
Aman Singh	Xlnet, Albert, Types of TL, QA, Survey on TL, NLI, Text classification	Data Generation and Pre-Processing
Tej Kiran	ULMFiT, RoBERTa, XLM, TL in low resource multi-lingual domain	BERT for NLI task
Krishna Mohan	BERT, GPT, ELMo, Cross-Lingual TL techniques, Survey on TL	BERT for NLI task
Shivam Aggarwal	MAML++, Bayesian MAML, X-MAML, ATAML, few shot MAML techniques	Meta-training and testing
Aman Aryan	MAML, XMAML, TreeMAML, MetaCurriculum Learning, MAML in Few Shot NMT, Domain Adaptation, Cross-Lingual Tasks.	Meta-training and testing

9 Conclusion

In this work, we studied and understood various transfer learning and meta-learning models. It is clear that there is not much data available for low resource languages, especially Indian languages. This motivated us to implement a cross-lingual learning technique that uses both Transfer Learning and Model Agnostic Meta Learning. First we have taken a pre-trained IndicBERT model that is trained on few Indian Languages and using that to meta-train on different multi-lingual NLI tasks. For this implementation we have chose to train the model on NLI tasks. Although we could not make the model learn the task, we have learned few valuableinsights from the experimentation that can be extended to future work.

References

[1] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, and P. Kumar, "IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages," in *Findings of EMNLP*, 2020.

[2] A. Malte and P. Ratadiya, "Evolution of transfer learning in natural language processing," *arXiv preprint arXiv:1910.07370*, 2019.

[3] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, pp. 1126–1135, PMLR, 2017.

[4] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, Lille, 2015.

[5] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," *arXiv preprint arXiv:1606.04080*, 2016.

[6] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *arXiv preprint arXiv:1703.05175*, 2017.

[7] R. Zhan, X. Liu, D. F. Wong, and L. S. Chao, "Meta-curriculum learning for domain adaptation in neural machine translation," *arXiv preprint arXiv:2103.02262*, 2021.

[8] F. Nooralahzadeh, G. Bekoulis, J. Bjerva, and I. Augenstein, "Zero-shot cross-lingual transfer with meta learning," *arXiv preprint arXiv:2003.02739*, 2020.

[9] S. Uppal, V. Gupta, A. Swaminathan, H. Zhang, D. Mahata, R. Gosangi, R. R. Shah, and A. Stent, "Two-step classification using recasted data for low resource settings," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, (Suzhou, China), pp. 706–719, Association for Computational Linguistics, Dec. 2020.

[10] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[11] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets," *arXiv preprint arXiv:1906.05474*, 2019.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[14] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[15] Z. Chi, L. Dong, F. Wei, W. Wang, X.-L. Mao, and H. Huang, "Cross-lingual natural language generation via pre-training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 7570–7577, 2020.

[16] Y. Kim, Y. Gao, and H. Ney, "Effective cross-lingual transfer of neural machine translation models without shared vocabularies," *arXiv preprint arXiv:1905.05475*, 2019.

[17] V. Kumar, N. Joshi, A. Mukherjee, G. Ramakrishnan, and P. Jyothi, "Cross-lingual training for automatic question generation," *arXiv preprint arXiv:1906.02525*, 2019.

[18] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[20] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of bert," *arXiv preprint arXiv:1904.09077*, 2019.

[21] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1604.02201*, 2016.

[22] J.-K. Kim, Y.-B. Kim, R. Sarikaya, and E. Fosler-Lussier, "Cross-lingual transfer learning for pos tagging without cross-lingual resources," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 2832–2838, 2017.

[23] H. Huang, Y. Liang, N. Duan, M. Gong, L. Shou, D. Jiang, and M. Zhou, "Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks," *arXiv preprint arXiv:1909.00964*, 2019.

[24] Z. Alyafeai, M. S. AlShaibani, and I. Ahmad, "A survey on transfer learning in natural language processing," *arXiv preprint arXiv:2007.04239*, 2020.

[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*, pp. 270–279, Springer, 2018.

[26] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.

[27] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[28] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[29] Y.-A. Chung, H.-Y. Lee, and J. Glass, "Supervised and unsupervised transfer learning for question answering," *arXiv preprint arXiv:1711.05345*, 2017.

[30] Z. Shaheen, G. Wohlgenannt, and E. Filtz, "Large scale legal text classification using transformer models," *arXiv preprint arXiv:2010.12871*, 2020.

[31] R. Puri and B. Catanzaro, "Zero-shot text classification with generative language models," *arXiv preprint arXiv:1912.10165*, 2019.

[32] Y. Kim, P. Petrov, P. Petrushkov, S. Khadivi, and H. Ney, "Pivot-based transfer learning for neural machine translation between non-english languages," *arXiv preprint arXiv:1909.09524*, 2019.

[33] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A survey on recent approaches for natural language processing in low-resource scenarios," *arXiv preprint arXiv:2010.12309*, 2020.

[34] A. Antoniou, H. Edwards, and A. Storkey, "How to train your maml," *arXiv preprint arXiv:1810.09502*, 2018.

[35] X. Jiang, M. Havaei, G. Chartrand, H. Chouaib, T. Vincent, A. Jesson, N. Chapados, and S. Matwin, "On the importance of attention in meta-learning for few-shot text classification," *arXiv preprint arXiv:1806.00852*, 2018.

[36] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7343–7353, 2018.

[37] Z. Liu, R. Zhang, Y. Song, and M. Zhang, "When does maml work the best? an empirical study on model-agnostic meta-learning in nlp applications," *arXiv preprint arXiv:2005.11700*, 2020.

[38] S. Deng, N. Zhang, Z. Sun, J. Chen, and H. Chen, "When low resource nlp meets unsupervised language model: Meta-pretraining then meta-learning for few-shot text classification (student abstract)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13773–13774, 2020.

[39] T. Bansal, R. Jha, and A. McCallum, "Learning to few-shot learn across diverse natural language classification tasks," *arXiv preprint arXiv:1911.03863*, 2019.

[40] W. Yin, "Meta-learning for few-shot natural language processing: A survey," *arXiv preprint arXiv:2007.09604*, 2020.

[41] J. Gu, Y. Wang, Y. Chen, K. Cho, and V. O. Li, "Meta-learning for low-resource neural machine translation," *arXiv preprint arXiv:1808.08437*, 2018.

[42] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

[43] A. Sharaf, H. Hassan, and H. Daumé III, "Meta-learning for few-shot nmt adaptation," *arXiv preprint arXiv:2004.02745*, 2020.

[44] J. R. Garcia, F. Freddi, F.-T. Liao, J. McGowan, T. Nieradzik, D.-s. Shiu, Y. Tian, and A. Bernacchia, "Meta-learning with maml on trees," *arXiv preprint arXiv:2103.04691*, 2021.

[45] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *arXiv preprint arXiv:1903.03096*, 2019.

[46] Z.-Y. Dou, K. Yu, and A. Anastasopoulos, "Investigating meta-learning algorithms for low-resource natural language understanding tasks," *arXiv preprint arXiv:1908.10423*, 2019.

[47] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *arXiv preprint arXiv:1901.07291*, 2019.