Enhancing Sentiment Knowledge via Self-Supervised Meta-Learning

Hyunjong Kim
Department of Industrial Engineering
Seoul National University
hjkim@bdai.snu.ac.kr

Sungzoon Cho
Department of Industrial Engineering
Seoul National University
zoon@snu.ac.kr

Abstract

Few-shot learning and meta-learning have attracted increasing attention in recent years. Meta-learning in NLP has also shown great progress, including subset masked language modeling tasks (SMLMT). Although SMLMT has led to improved few-shot generalization, its potential for further uses has not been explored. In this paper, we propose SentiSMLMT, an extension of SMLMT which injects sentiment knowledge into the model by utilizing sentiment lexicon within the self-supervised framework. Experimental results show that our approach is simple but effective, achieving significant improvements in sentiment-related tasks.

1 Introduction

Pre-trained language models have achieved significant progress on diverse tasks of NLP (Howard and Ruder, 2018; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019). Along with the success of pre-trained language models, a number of novel approaches have been proposed for sentiment analysis (Araci, 2019; Yin et al., 2020; Zhou et al., 2020; Tian et al., 2020; Sosea and Caragea, 2021; Barbieri et al., 2022; Ein-Dor et al., 2022). Some approaches concentrate on associating the pre-training objective with sentiment knowledge (Zhou et al., 2020; Tian et al., 2020; Sosea and Caragea, 2021), and some make use of existing lexicons (Khanpour and Caragea, 2018; Zhou et al., 2020; Sosea and Caragea, 2021).

Meanwhile, meta-learning methods (Vinyals et al., 2016; Finn et al., 2017; Snell et al., 2017) have shown to be effective in various applications, especially in few-shot scenarios. Although most of the research in meta-learning aims to solve vision tasks, recent work has also shown the usefulness of meta-learning in the field of NLP (Guo et al., 2018; Yu et al., 2018; Geng et al., 2019; Mi et al., 2019; Bansal et al., 2020a,b, 2021; Wang et al., 2021). In particular, Bansal et al. (2020b) proposed the first self-supervised approach in NLP using meta-learning, inspired by the Cloze task (Taylor, 1953).

In this paper, we propose an effective method to enhance sentiment knowledge by integrating knowledge of a sentiment lexicon into the self-supervised meta-training objective proposed by Bansal et al. (2020b). We design meta-training tasks based on the assumption that the relationship between words plays a crucial role in the guidance of sentiment-oriented meta-training. Experimental results show that our method achieves significant improvements over baselines, where our focus is mainly on few-shot text classification tasks. Moreover, we conduct extensive experiments for ablation studies on two factors of the training process.

2 Proposed Approach

2.1 Background

In this work, we leverage Subset Masked Language Modeling Tasks (SMLMT) (Bansal et al., 2020b), a self-supervised approach to create a large number of meta-learning tasks from unlabeled text. It is fundamentally an N-way k-shot classification task, where each label is a vocabulary word and k sentences containing each word are given as support set with the corresponding word masked out. The model learns to classify sentences where each sentence includes one mask token.

Bansal et al. (2020b) showed that meta-training SMLMT tasks on large-scale yields better few-shot performance across diverse text classification tasks. However, designing meta-learning tasks to be better aligned with the target task can lead to further improvements. Such task-guided meta-training can produce better initial points for fine-tuning, which is crucial in few-shot settings. In particular, we focus on sentiment analysis, and propose methods to create sentiment-aware SMLMT tasks that effectively inject sentiment knowledge into the model.

2.2 Sentiment-aware Meta-training

We present SentiSMLMT, or sentiment-aware SMLMT, an extension of SMLMT which is specifically designed for inducing sentiment-specific bias. In SentiSMLMT, we fully exploit sentiment-carrying words when creating a task, instead of randomly sampling words from the entire vocabulary. We leverage sentiment lexicon as the source of sentiment-carrying words.

Lexicon. We use AFINN sentiment lexicon (Nielsen, 2011) to create SentiSMLMT tasks. AFINN sentiment lexicon[1] is a list of 3,382 English words annotated with an integer sentiment rating r, where $-5 \leq r \leq 5$. In order to employ words that indicate stronger sentiments, we define *positive words* and *negative words* as words that satisfy $r \geq 3$ and $r \leq -3$ respectively. P, N and $S = P \cup N$ each denotes the set of positive words, negative words and *sentiment words*.

Corpus. We use Yelp review dataset[2] as the source text to create meta-training tasks since it is a corpus rich with sentiment information and emotional expressions. Among 6,990,280 English reviews in the corpus, we only use a small part of it by randomly sampling 100,000 reviews. We obtain vocabulary V from this sampled corpus.

Task Proposal. In this work, we focus on designing the subset of vocabulary words that constitute an SMLMT task. We propose three methods based on the relationship between words, making use of the knowledge obtained from the sentiment lexicon. We mainly consider $n = 2$ tasks and assume m tasks are being created.

SentiSMLMT$_{\text{random}}$ includes two random sentiment words, either positive or negative.

$$T_{random} = \{(s_1, s_2)_i \mid s_1, s_2 \in V \cap S, s_1 \neq s_2\}_{i=1}^{m}$$

SentiSMLMT$_{\text{binary}}$ includes one sentiment word and one neutral word.

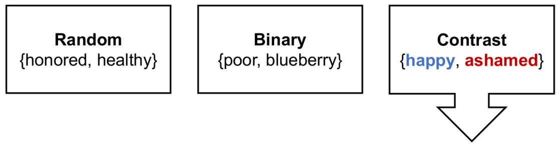
$$T_{binary} = \{(s, u)_i \mid s \in V \cap S, u \in V - V \cap S\}_{i=1}^{m}$$

SentiSMLMT$_{\text{contrast}}$ includes one positive word and one negative word.

$$T_{contrast} = \{(p, n)_i \mid p \in V \cap P, n \in V \cap N\}_{i=1}^{m}$$

[1] https://github.com/fnielsen/afinn/blob/master/afinn/data/AFINN-en-165.txt

[2] https://www.yelp.com/dataset



	Sentence	Class
Support Set	Overall, the coffee was good, Wi-Fi was stable, I was **[MASK]**.	1
	The owner was very accommodating and was more than **[MASK]** to show me different colors.	1
	I sometimes am **[MASK]** to live on this earth with people who are such self-righteous people.	2
	Absolutely horrible experience now and they should be **[MASK]** of their customer service.	2
Query Set	I would be **[MASK]** if this were my store .. Clean it up guys.	?
	They gave my Yorkie a great looking cut and grooming, and when I came to pick him up, he was **[MASK]**, not stressed.	?

Figure 1: Examples of SentiSMLMT word pairs. A 2-way 2-shot task for SentiSMLMT$_{\text{contrast}}$ is also shown. Note that more samples are used for support set and query set in actual experiments.

Note that we use uniform sampling when sampling words from their corresponding set. An example of each approach is displayed in Figure 1. For meta-training, we use the same model architecture as in Bansal et al. (2020b), a BERT encoder with an additional parameter generator. It is meta-trained using the MAML framework (Finn et al., 2017). Instead of training from scratch, we initialize the model with SMLMT model checkpoints[3] which is meta-trained with 2 million general SMLMT tasks. It can be viewed as a type of post-training (Xu et al., 2019; Zhuang et al., 2021) or inter-training (Ein-Dor et al., 2022), where a large-scale language model is adapted for a specific purpose before the fine-tuning step, without any use of labeled data.

3 Experiments

3.1 Experimental Setup

We meta-train each model with 10000 tasks of each type. For all tasks, 20 samples are used for both support set and query set. Adam optimizer (Kingma and Ba, 2014) is used for both inner loop and outer loop. Other hyperparameters can be found in Appendix A.

3.2 Evaluation Methodology

We evaluate our models on multiple few-shot text classification tasks. Among the diverse downstream tasks used in Bansal et al. (2020b), we choose all tasks that are related to sentiments; airline[4], Amazon rating classification and Amazon

[3] Available at https://github.com/iesl/metanlp.

[4] Sentiment classification on tweets about airlines.

Task	N	k	BERT†	SMLMT†	SentiSMLMT		
					random	binary	contrast
Airline	3	4	42.76 ± 13.50	42.83 ± 06.12	54.38 ± 07.22	46.58 ± 07.90	**55.05** ± 08.37
		8	38.00 ± 17.06	51.48 ± 07.35	60.29 ± 07.04	51.39 ± 07.11	**64.47** ± 03.88
		16	58.01 ± 08.23	58.42 ± 03.44	64.69 ± 05.29	59.90 ± 08.17	**66.69** ± 06.01
Rating Books	3	4	39.42 ± 07.22	34.96 ± 03.94	37.70 ± 09.40	38.32 ± 05.53	**44.55** ± 07.85
		8	39.55 ± 10.01	37.20 ± 04.15	39.91 ± 08.92	40.56 ± 06.33	**44.36** ± 09.01
		16	43.08 ± 11.78	43.62 ± 04.59	47.58 ± 06.85	46.00 ± 04.44	**52.85** ± 05.85
Rating DVD	3	4	32.22 ± 08.72	38.26 ± 03.62	39.55 ± 05.77	40.49 ± 04.90	**40.50** ± 05.50
		8	36.35 ± 12.50	37.92 ± 03.61	38.60 ± 05.50	37.90 ± 03.37	**44.48** ± 04.87
		16	42.79 ± 10.18	41.87 ± 04.30	43.64 ± 04.19	44.18 ± 03.37	**49.19** ± 04.06
Rating Electronics	3	4	39.27 ± 10.15	37.69 ± 04.82	39.73 ± 05.86	36.00 ± 05.15	**43.76** ± 07.63
		8	28.74 ± 08.22	39.98 ± 04.03	46.17 ± 04.73	43.37 ± 05.79	**48.40** ± 08.47
		16	45.48 ± 06.13	45.85 ± 04.72	50.56 ± 03.91	48.32 ± 06.85	**53.51** ± 06.60
Rating Kitchen	3	4	34.76 ± 11.20	40.75 ± 07.33	42.69 ± 08.88	43.79 ± 05.22	**47.54** ± 08.52
		8	34.49 ± 08.72	43.04 ± 05.22	47.78 ± 06.29	47.13 ± 06.54	**48.90** ± 09.81
		16	47.94 ± 08.28	46.82 ± 03.94	52.65 ± 04.67	49.12 ± 07.83	**54.72** ± 07.25
Sentiment Books	2	4	54.81 ± 03.75	55.68 ± 02.56	59.66 ± 05.45	55.35 ± 03.12	**65.74** ± 06.09
		8	53.54 ± 05.17	60.23 ± 05.28	65.51 ± 06.76	61.98 ± 05.18	**72.13** ± 05.90
		16	65.56 ± 04.12	62.92 ± 04.39	68.64 ± 05.52	67.53 ± 03.26	**73.84** ± 05.32
Sentiment DVD	2	4	54.98 ± 03.96	52.95 ± 02.51	54.92 ± 04.60	54.29 ± 03.54	**59.54** ± 06.21
		8	55.63 ± 04.34	54.28 ± 04.20	59.53 ± 04.65	54.83 ± 03.67	**64.66** ± 07.65
		16	58.69 ± 06.08	57.87 ± 02.69	61.16 ± 05.27	60.10 ± 03.01	**73.65** ± 03.96
Sentiment Electronics	2	4	58.77 ± 06.10	56.40 ± 02.74	60.54 ± 05.24	58.83 ± 04.47	**69.21** ± 06.43
		8	59.00 ± 05.78	62.06 ± 03.85	65.56 ± 05.60	61.65 ± 04.65	**70.40** ± 06.89
		16	67.32 ± 04.18	64.57 ± 04.32	69.90 ± 05.37	64.92 ± 05.63	**75.17** ± 04.00
Sentiment Kitchen	2	4	56.93 ± 07.10	58.64 ± 04.68	67.19 ± 05.15	62.22 ± 05.05	**72.54** ± 06.33
		8	57.13 ± 06.60	59.84 ± 03.66	68.35 ± 04.96	61.83 ± 04.81	**73.54** ± 07.04
		16	68.88 ± 03.39	65.15 ± 05.83	75.92 ± 04.67	68.05 ± 05.34	**79.67** ± 03.69
Overall Average		4	45.99 ± 07.97	46.46 ± 04.26	50.71 ± 06.40	48.43 ± 04.99	**55.38** ± 06.99
		8	44.71 ± 08.71	49.56 ± 04.59	54.63 ± 06.05	51.18 ± 05.27	**59.04** ± 07.06
		16	55.31 ± 06.93	54.12 ± 04.25	59.42 ± 05.08	56.46 ± 05.32	**64.37** ± 05.19

Table 1: k-shot accuracy on sentiment-related downstream tasks. Models with the best and the second-best performance are denoted in bold and underlined font respectively. Results marked † are from Bansal et al. (2020b).

sentiment classification[5].

We consider $k = 4, 8, 16$ settings for few-shot learning. Each model is fine-tuned on 10 different sets for each task and each k, and we report the average of 10 runs. For fine-tuning, we follow the settings of Bansal et al. (2020b).

3.3 Main Results

Full results are shown in Table 1. We compare our results with other unsupervised baselines, BERT (Devlin et al., 2019) and SMLMT (Bansal et al., 2020b). First, we observe that all three SentiSMLMT approaches improve upon the original SMLMT model. SentiSMLMT$_{\text{random}}$ and SentiSMLMT$_{\text{contrast}}$ improve on all tasks, while SentiSMLMT$_{\text{binary}}$ improves on most of the tasks.

Among the three approaches, SentiSMLMT$_{\text{contrast}}$ turns out to be the best-performing approach, achieving the highest accuracy on all 9 tasks. It outperforms other models by a large margin, where it shows accuracy gains of 8.92%, 9.48% and 10.25% compared to SMLMT, for $k = 4, 8, 16$ respectively. We believe this is because the training objective of SentiSMLMT$_{\text{contrast}}$ is most similar to the objective of target tasks, where the labels getting predicted are usually polarized. Also, sampling positive words and negative words in a fixed ratio might have worked as class balancing in supervised learning.

It is interesting that such an improvement is made with a relatively small number of tasks, demonstrating the effectiveness of sentiment-aware meta-training. Moreover, the consistent improvements on target tasks of all 4 domains imply that the sentiment knowledge acquired is general; it can be generalized across domains.

[5]Amazon tasks include 4 domains each.

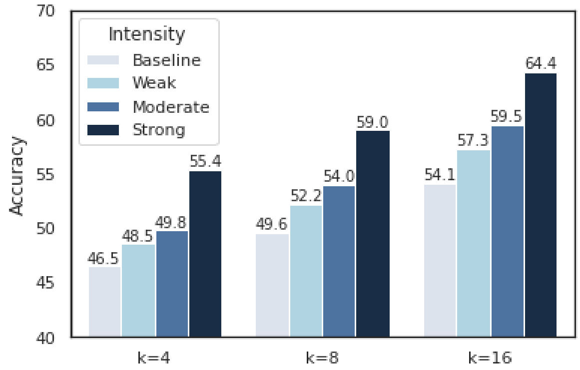


Figure 2: Performance of SentiSMLMT$_{\text{contrast}}$ models meta-trained with word pairs of different sentiment intensities. This figure shows the overall average of 9 downstream tasks that appear in Table 1. Baseline indicates the original SMLMT model.

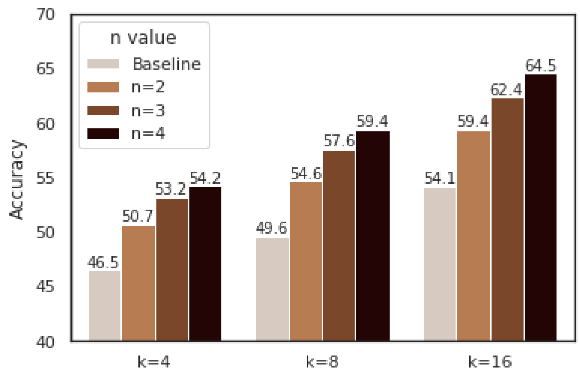


Figure 3: Performance of SentiSMLMT$_{\text{random}}$ models meta-trained with n-way tasks with different n, each $n = 2, 3, 4$. This figure shows the overall average of 9 downstream tasks that appear in Table 1. Baseline indicates the original SMLMT model.

3.4 Effect of Sentiment Intensity

As described in Section 2.2, we intentionally used sentiment words that display strong sentiments. In this section, we conduct extensive experiments to identify the significance of sentiment intensity, specifically in the case of SentiSMLMT$_{\text{contrast}}$. For each setting of *weak*, *moderate*, and *strong*[6] intensity, we only use words that satisfy $|r| = 1$, $|r| = 2$ and $3 \leq |r| \leq 5$ respectively. The results are presented in Figure 2. Interestingly, we find that sentiment intensity is a crucial factor in injecting sentiment knowledge. Utilizing words with stronger intensity produces greater improvements. This could be a motivating example of exploiting sentiment intensity to train the model more effectively.

[6]Identical to the original setting.

3.5 Effect of n

Unlike the other two, SentiSMLMT$_{\text{random}}$ can be easily extended by increasing the value of n. In Figure 3, we show the results obtained by using different values of n. The results indicate that it can be beneficial to make n larger (though it might saturate at some point), which is actually making the meta-training tasks harder. Results on $n = 4$ are comparable to that of SentiSMLMT$_{\text{contrast}}$, though it takes more time and computing resource to meta-train the model. This shows the scalability of SentiSMLMT, along with the possibility of further improvements by designing tasks with larger n values.

3.6 Performance on Other Target Tasks

Task	N	k	SMLMT†	SentiSMLMT contrast
CoNLL	4	4	46.81	**50.55***
		8	**61.72**	61.36
		16	**75.82**	75.69
MITR	8	4	46.23	**51.57***
		8	61.15	**62.32**
		16	69.22	**71.03**
Disaster	2	4	**62.26**	61.94
		8	**67.89**	64.92
		16	72.86	**73.88**
Political Bias	2	4	57.72	**60.80**
		8	63.02	**64.47**
		16	66.35	**66.60**
Political Audience	2	4	**57.94**	57.88
		8	**62.82**	60.60
		16	**64.57**	63.46
Political Message	9	4	16.16	**17.31**
		8	19.24	**20.22**
		16	**21.91**	21.69
Scitail	2	4	50.68	**50.88**
		8	55.60	**55.94**
		16	56.51	**56.81**
Overall Average		4	48.26	**50.13**
		8	**55.92**	55.69
		16	61.03	**61.31**

Table 2: k-shot accuracy on downstream tasks not related to sentiments. * indicates statistical significance in a two-tailed t-test ($p < 0.05$) against SMLMT. Results marked † are from Bansal et al. (2020b).

In order to evaluate the effect of sentiment-aware meta-training on other tasks, we also compare the results of SMLMT and SentiSMLMT$_{\text{contrast}}$ on downstream tasks presented in Bansal et al. (2020b), but not used in Section 3.3. The results are presented in Table 2. As expected, the differences are marginal in most cases, not being statistically significant. While it aligns well with our purpose of building a sentiment-oriented model, it turns out that it doesn't hurt the performance of other target tasks.

4 Conclusion

In this work, we introduced SentiSMLMT, an approach that enables sentiment-oriented meta-training for improved performance on sentiment-related tasks. Although we limited the scope to a fully self-supervised setting to observe the characteristics of SentiSMLMT precisely without any interference of other training objectives, it still can be jointly trained with supervised tasks as presented in Bansal et al. (2020b), where additional performance gains are expected.

Limitations

As the task creation process of our method heavily relies upon the sentiment lexicon, its efficacy may depend on its quality. Although we could readily access a high-quality sentiment lexicon such as AFINN (Nielsen, 2011) in English, it might not be able in some languages, especially for low-resource languages. The proposed work might not be applicable to such languages.

Ethics Statement

We do not find any ethical concerns regarding the research presented in this paper.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C2093785).

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Trapit Bansal, Karthick Prasad Gunasekaran, Tong Wang, Tsendsuren Munkhdalai, and Andrew McCallum. 2021. Diverse distributions of self-supervised tasks for meta-learning in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5812–5824, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2020a. Learning to few-shot learn across diverse natural language classification tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5108–5123, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020b. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 522–534, Online. Association for Computational Linguistics.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liat Ein-Dor, Ilya Shnayderman, Artem Spector, Lena Dankin, Ranit Aharonov, and Noam Slonim. 2022. Fortunately, discourse markers can enhance language models for sentiment analysis. *arXiv preprint arXiv:2201.02026*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text 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 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703, Brussels, Belgium. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Hamed Khanpour and Cornelia Caragea. 2018. Fine-grained emotion detection in health-related online posts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1160–1166, Brussels, Belgium. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3151–3157.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In *Workshop on'Making Sense of Microposts: Big things come in small packages*, pages 93–98.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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

Tiberiu Sosea and Cornelia Caragea. 2021. eMLM: A new pre-training objective for emotion related tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 286–293, Online. Association for Computational Linguistics.

Wilson L Taylor. 1953. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4067–4076, Online. Association for Computational Linguistics.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Jixuan Wang, Kuan-Chieh Wang, Frank Rudzicz, and Michael Brudno. 2021. Grad2task: Improved few-shot text classification using gradients for task representation. *Advances in Neural Information Processing Systems*, 34:6542–6554.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. 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)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Da Yin, Tao Meng, and Kai-Wei Chang. 2020. SentiBERT: A transferable transformer-based architecture for compositional sentiment semantics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3695–3706, Online. Association for Computational Linguistics.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

Jie Zhou, Junfeng Tian, Rui Wang, Yuanbin Wu, Wenming Xiao, and Liang He. 2020. SentiX: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 568–579, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227.

A Implementation Details

Hyperparameters for meta-training not stated in Section 3.1 are listed in Table 3.

Hyperparameter	Value
Meta-training Epochs	1
Task Batch Size	5
Outer Loop Learning Rate	1e-05
Inner Loop Learning Rate	5e-05
Lowercase Text	False
Maximum Sequence Length	128

Table 3: Hyperparameters for meta-training.

All experiments were conducted on 1 NVIDIA TITAN V GPU with 12GB memory. Meta-training 10000 $n = 2$ tasks took about 12 hours. For meta-training, we used the code we reimplemented using PyTorch (Paszke et al., 2019) and Hugging Face Transformers (Wolf et al., 2020) library.