Enhancing Few-Shot Topic Classification with Verbalizers A Study on Automatic Verbalizers and Ensemble Methods

Quang Anh Nguyen[*‡], **Nadi Tomeh**[†], **Mustapha Lebbah**[*], **Thierry Charnois**[†], **Hanene Azzag**[†], **Santiago Cordoba Muñoz**[‡]
[*]Université Paris-Saclay - DAVID Lab, UVSQ, Versailles, France
[†]Université Sorbonne Paris Nord - LIPN CNRS UMR 7030, Villetaneuse, France
[‡]Groupe BPCE - Paris, France

Abstract

As pretrained language model emerge and consistently develop, prompt-based training has become a well-studied paradigm to improve the exploitation of models for many natural language processing tasks. Furthermore, prompting demonstrates great performance compared to conventional fine-tuning in scenarios with limited annotated data, such as zero-shot or few-shot situations. Verbalizers are crucial in this context, as they help interpret masked word distributions generated by language models into output predictions. This study introduces a benchmarking approach to assess three common baselines of verbalizers for topic classification in few-shot learning scenarios. Additionally, we find that increasing the number of label words for automatic label word searching enhances model performance. Moreover, we investigate the effectiveness of template assembling with various aggregation strategies to develop stronger classifiers that outperform models trained with individual templates. Our approach achieves comparable results to prior research while using significantly fewer resources. Our code is available at https://github.com/quang-anh-nguyen/verbalizer_benchmark.git.

1. Introduction

Fine-tuning pre-trained language models (PLMs) have led to significant improvements across various Natural Language Processing (NLP) tasks. Traditional methods involve replacing the PLM's masked language modeling head with a task-specific head and fine-tuning the entire model (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020). However, such introduction of parameters require a substantial amount of labeled data, making them unsuitable for few-shot or zero-shot scenarios. Inspired by by the approach introduced in GPT-3 (Brown et al., 2020), prompting has emerged as a new paradigm, where downstream tasks are adapted to align with the pretraining objective.

Prompt-based finetuning allows to exploit PLMs' knowledge and reduces the gap between pretraining and finetuning (Petroni et al., 2019; Chen et al., 2022). In this framework, templates and verbalizers (Schick and Schütze, 2021a; Gao et al., 2021) are crucial mapping between task-specific inputs and labels, to textual data for the PLM. Given the importance of verbalizers, our objective is to establish an evaluation for the manual verbalizer (Schick and Schütze, 2021a), the soft verbalizers (Hambardzumyan et al., 2021) and the automatic verbalizers (Schick et al., 2020) (see section 3). In addition, focusing on (Schick et al., 2020), we also study the performance of PETAL with varying numbers of label words, using only limited resources, i.e. without supplementary unlabeled data for distillation. Also, template ensemble allows to boost the performance of individuals and plays the role of template selection. The proposed evaluation involves repetition over multiple samplings of labeled data, giving more robustness and less dependence on the sampled instances.

Our contribution is summarized as: (i) evaluating the performance of manual, soft and automatic verbalizers for the topic classification problem, on three public datasets and a real-world French dataset; (ii) demonstrating the importance of the number of label words for the automatic verbalizer algorithm; (iii) showing that model ensembling with multiple templates can improve prompting over individual templates, and eliminate the need of prompt selection; (iv) achieving comparable performance to previous work using significantly less data.

2. Related Work

Prompt-based finetuning In this framework, the input is wrapped with a task-specific *template* to reformulate the target task as language modeling. The *verbalizer* then reprojects the distribution of `MASK` into the answer space. For textual templates and verbalizer, their selection has a significant influence on the classification performance (Gao et al., 2021). (Schick and Schütze, 2021a,b) use task-specific manual templates and verbalizers that work efficiently. However, their construction requires both domain expertise of downstream tasks and understanding of the PLMs, otherwise the searching process of these elements may be exhaustive with a large number of classes. Meanwhile, (Lester et al., 2021; Liu et al., 2022; Li and Liang, 2021) propose to freeze the PLM and instead optimize prompt tokens. Despite be-

LREC-COLING 2024, pages 5956–5965
20-25 May, 2024.

ing human-independent and storage-saving, continuous prompts have only been studied in data-abundant scenarios, and produce uninterpretable tokens. Here we study textual templates and focus on the search of label words for the verbalizer. A method not studied here is KPT (Hu et al., 2022) where an external knowledge base helps to search for words related to the topic titles. For benchmarking purposes, we exclude additional data or knowledge base from available resources (section 4.1). Additionally, (Cui et al., 2022) uses contrastive learning to learn class prototypes can be viewed as extended soft verbalizers.

Few shot learning setting The few-shot accuracy of PLMs is sensitive to many factors, mainly the prompt formulation (Perez et al., 2021). Effective model selection, crucial to guarantee performance was usually done on a large validation set. However, from a practical aspect, we follow the procedure proposed by (Zheng et al., 2022), using a small validation set, and evaluate the test set results obtained on different training data samplings. This setup allows us to achieve a robust and global evaluation of learning algorithms.

Ensemble modeling Given the sensitivity of prompt-based methods in few-shot context, each prompt can be more or less effective towards eliciting knowledge from the PLM. Ensemble approach provides an efficient way to reduce instability across prompts and stronger classifiers (Schick and Schütze, 2021a; Jiang et al., 2020). We study in this work the impact of aggregating strategy on the performance of assembled models.

3. Methodology

Let $\mathcal{M}$ be a language model with vocabulary V. Following (Schick and Schütze, 2021a,b), we define the template - verbalizer pair. Let $(\mathbf{x}, y)$ be an example of the classification problem, where $\mathbf{x}$ represents one or many sentences and y is its label in the label set $\mathcal{Y}$. A template T maps $\mathbf{x}$ into a masked sequence $T(\mathbf{x})$ of tokens in $V \cup \{\texttt{MASK}\}$. A verbalizer $v : \mathcal{Y} \rightarrow \mathcal{P}(V)$ maps each label to a set of words characterizing the class (called label words). The probability of the label conditioned on the input is then modeled by the logits of its label words conditioned on the masked sequence:

$$p(y|\mathbf{x}) \propto \exp\left(\frac{1}{|v(y)|}\sum_{w \in v(y)} \mathcal{M}\left(w|T(\mathbf{x})\right)\right) \quad (1)$$

This work aims to evaluate the three following baselines for few-shot topic classification (section 5.2), as well as to study the effect of several factors to their performance (sections 5.3 and 5.4).

Manual The label words can be predefined manually. It has been shown that different choices of label words can have major importance for the model performance (Gao et al., 2021).

Soft WARP (Hambardzumyan et al., 2021) proposes to represent each label y by a vector v_y instead of concrete words, initialized with static embeddings of the manual label words and optimized alongside the PLM, such that:

$$p(y|\mathbf{x}) \propto \exp\left(v_y \cdot h\right) \quad (2)$$

With h the embedding of the $\texttt{MASK}$ token in $T(\mathbf{x})$.

Auto Among automatic methods, PETAL (Schick et al., 2020) allows identifying words suitable to represent classes from training data itself without additional data or knowledge. Consider the classification problem as many one-vs-rest binary problems to find label words for each class separately. For a label $\bar{y}$ of support, $\mathcal{D}_{\bar{y}} = \{(\mathbf{x}, y) \in \mathcal{D}_{\text{train}} \mid y = \bar{y}\}$, PETAL takes the top k words w that maximize the likelihood ratio (LR) of positive examples and minimize that of negative examples:

$$v(\bar{y}) = \underset{w}{\text{top-}k}\left[\frac{1}{|\mathcal{D}_{\bar{y}}|}\sum_{(\mathbf{x},y)\in\mathcal{D}_{\bar{y}}} \ell_{\text{LR}}\left(w, \mathbf{x}\right) - \frac{1}{|\mathcal{D}_{\text{train}} \backslash \mathcal{D}_{\bar{y}}|}\sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}\backslash\mathcal{D}_{\bar{y}}} \ell_{\text{LR}}\left(w, \mathbf{x}\right)\right] \quad (3)$$

Where:

$$\ell_{\text{LR}}(w, \mathbf{x}) = \log\frac{p_{\mathcal{M}}\left(w|T(\mathbf{x})\right)}{1 - p_{\mathcal{M}}\left(w|T(\mathbf{x})\right)} \quad (4)$$

$$p_{\mathcal{M}}\left(w|T(\mathbf{x})\right) = \text{softmax}(\mathcal{M}(w|T(\mathbf{x})) \quad (5)$$

We demonstrate experimentally in this paper that increasing the number k of label words per class improves the quality of the automatic verbalizer. In our experiments, without specifying differently, we take $k = 15$.

After identifying label words, the PLMs are fine-tuned based on the chosen template and verbalizer, by minimizing the cross entropy loss between the predicted probabilities in equation (1) and the correct labels. Following the ensemble methods, the logits of individual models trained on different templates are aggregated into the final prediction, following three aggregation strategies: (vote) majority vote from individual predictions, (proba) averaging individual class probabilities, and (logit) averaging individual class logits (see Appendix B for explicit formulations). For the two latter, (Schick et al., 2020) shows that weighted averaging does not gain clear difference, thus we perform simply the uniform averaging.

4. Experiments

4.1. Setups

From the original training set, we sample a labeled set $\mathcal{D}$, of cardinality N. For each run, split $\mathcal{D}$ into two equal halves: $\mathcal{D}_{\text{train}}$ is used for fine-tuning with the template - verbalizer pair and $\mathcal{D}_{\text{valid}}$ for validation, on which the best checkpoint is retained.

The underlying PLM is `RoBERTa-large` (Liu et al., 2019) as in (Schick et al., 2020), except for FrN we use `CamemBERT-large` (Martin et al., 2020). Hyperparameters are inspired by (Schick and Schütze, 2021a). Each experiment is repeated 3 times with different samplings of $\mathcal{D}$, to evaluate the result variation with different training data. Details can be found in Appendix A.

4.2. Datasets

Our experiments are done on three public English datasets and a real-world dataset in French. For each dataset, four textual templates are described in detail in Appendix C. Manual verbalizers of these datasets are listed in Appendix D.

AG AG's News (Zhang et al., 2015) is a news classification dataset. Given a headline, a news need to be classified into one of 4 categories.

Yahoo Yahoo! Answers (Zhang et al., 2015) consists of questions and answers from Yahoo!, from 10 categories. The fields included are question title, question content and best answer.

DBpedia The DBpedia ontology dataset (Zhang et al., 2015) (Lehmann et al., 2015) consists of 14 classes from DBpedia 2014, each has 40 000 training and 5 000 testing samples. Each sample includes the title, its description, and its category.

FrN Our colleagues provide us with a collection of more than 5 millions real-world press articles. A small number of articles in French are assigned to 28 sectors by expert analysts. In this work, we extract 1048 articles from the 10 most frequent sectors, then keep 536 examples for testing, and use the rest to sample $\mathcal{D}$.

5. Results

5.1. Pilot Experiment

We examine the FrN dataset in zero-shot and in few-shot context with $N = 64$, with the manual verbalizer provided by our colleagues of 15 words per class. By retaining the k most important words, we observe the influence of the number of label words. Figure 1 shows a clear improvement from 5 label words for zero-shot and 10 for few-shot. Moreover, results are more stable with more label words. This correlation is highly dependent on the ordering of importance of $v(y)$, therefore on human decision. However, the observation motivates us to raise the number of label words for an automatic search algorithm. The study of this phenonmenon is presented in section 5.3.

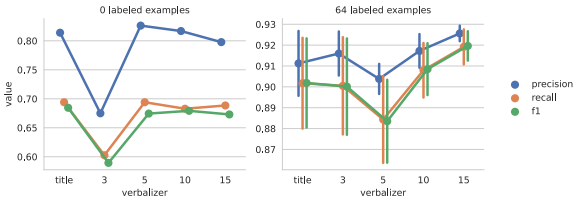

Figure 1: Study of different sizes for the manual verbalizer on the FrN dataset. *Title* means using class names as label words (see Appendix D).

5.2. Main Results

Table 1 shows the result over four datasets, for different quantity of available data.

Although initialized manually and operating in the embedding space, soft verbalizers do not perform better than manual verbalizers. The gap is visible for low N, and becomes negligible for $N \geq 128$. Intuitively, each topic covers many label words, thus can not be characterized by only one vector. The embedding space is continuous and larger than the vocabulary V, but less expressive than $\mathcal{P}(V)$, which suggests the importance of employing multiple label words per class.

The automatic verbalizer can perform similarly, if not exceed, the manual verbalizer for all datasets (with $N \geq 32$ for AG and $N \geq 128$ for others). Compared to PETAL where $N = 50$ and $\mathcal{M} =$ `RoBERTa-large`, our results with $N = 64$ surpass on AG but not on Yahoo. Our best automatic verbalizer with $N = 256$ is comparable PETAL with $N = 1000$. Note that PETAL procedure also includes one final step of knowledge distillation to annotate abundant unlabeled data for supervised sequence classification finetuning, while our implementation includes validation for early stopping with half of $\mathcal{D}$. The results show that we are able to achieve the same level of performance while using significantly less data.

5.3. Effect of Label Word Number on Automatic Searching

Figure 2 illustrates the performance of the automatic verbalizer while varying the number k for label word searching.

N	Verbalizer	AG	DBpedia	Yahoo	FrN
0	Majority class	25.00	7.14	10.00	16.79
	Manual	**72.14**	**73.17**	**58.91**	**69.40**
	Soft	71.89	54.57	52.34	64.74
32	Manual	83.96 ± 2.11	**91.68 ± 1.58**	**61.84 ± 1.17**	**81.16 ± 3.08**
	Soft	81.82 ± 3.30	85.95 ± 1.12	50.76 ± 2.84	74.63 ± 5.54
	Auto	**86.44 ± 1.89**	79.24 ± 7.98	50.08 ± 4.39	73.63 ± 1.35
50	PET (manual)	86.3		66.2	
	PETAL (auto)	84.2		62.9	
64	Manual	**88.14 ± 0.07**	**96.75 ± 0.33**	**65.29 ± 0.98**	**90.17 ± 2.18**
	Soft	87.37 ± 0.45	94.62 ± 2.06	64.64 ± 1.10	84.20 ± 0.88
	Auto	88.00 ± 0.46	92.01 ± 2.92	56.73 ± 5.05	86.38 ± 3.64
128	Manual	88.43 ± 0.33	**96.66 ± 1.14**	66.71 ± 0.61	**94.28 ± 1.32**
	Soft	87.32 ± 0.56	96.56 ± 2.00	65.93 ± 0.86	93.47 ± 2.44
	Auto	**88.86 ± 0.10**	95.75 ± 1.87	**67.42 ± 0.36**	93.47 ± 0.56
256	Manual	88.95 ± 0.46	98.24 ± 0.14	**70.63 ± 0.50**	**93.84 ± 0.81**
	Soft	88.51 ± 0.32	**98.27 ± 0.17**	69.81 ± 0.76	93.66 ± 1.04
	Auto	**89.64 ± 0.58**	98.23 ± 0.28	70.36 ± 1.03	93.16 ± 0.60
1000	PET (manual)	86.9		72.7	

Table 1: Accuracy on of three baselines, compared to PET and PETAL results extracted from (Schick and Schütze, 2021a; Schick et al., 2020). The automatic verbalizer uses $k = 15$ label words per class, except for PETAL with $k = 3$. The ensembling strategy is logit averaging. **Bold** are the best baselines.

A global trend confirms that increasing k produces more efficient verbalizers and raises the accuracy for limited data. Moreover, the effect is more visible for small N. This finding is different from the conclusion in (Schick et al., 2020) that the k has no impact on the global accuracy. We also remark that $k = 15$ can push the automatic performance close to the manual verbalizer, which was not achieved with $k = 3$ in the original PETAL. It can be concluded that increasing k for the automatic search can improve the ensemble models but has little effect on the distilled model trained on unlabeled data.

The conclusion about the effect of the label word number is less intuitive than it may seem. It is not trivial that using larger k in a few-shot learning model can actually help, since more parameters can not be trained well with very little data, and potential of adding noise also arises. In some cases, we notice that using more label words may compensate for annotating more data as an alternative strategy. On AG and DBpedia, using $k = 100$ for $N = 64$ almost reaches the same level as $N = 96$. On Yahoo, using $k = 50$ for $N = 32$ achieves a similar result as $k = 3$ for $N = 64$.

5.4. Effectiveness of Model Ensemble

Results using individual templates and by assembling the following three methods, for three baselines of verbalizers are illustrated in figures 2 and 3. In most cases, assembled models produce more reliable predictions, surpassing the most efficient template. Ensembles also enhance stability and replace the need of prompt selection, particularly when the performance of different templates are substantially divergent.

Comparing the three aggregating methods, we notice that voting performs worse than probability and logit averaging in general, but the difference is negligible compared to the gain between assembling and individual templates.

6. Conclusion and Discussion

In this paper, we provided a complete and detailed procedure for robust evaluation of three types of verbalizers, serving as baselines for future works of verbalizers using textual templates. With small validation sets, our results achieve a similar performance level as previous works on automatic verbalizer, while using less training data and excluding semi-supervised training via distillation. Experimental results also leverage the advantage of using more label words for automatic label word searching, in comparison to more annotated data. Our work also confirms the effectiveness of ensemble models with multiple templates, which allows surpassing the best template and eases the need for template selection.

For many NLP tasks, in a full-data scenarios, state-of-the-art large-scale LMs give impressive overall results. However, small LMs can potentially yield better downstream and domain-specific abilities. In few-shot learning, simpler models can actually generalize given the limited amount of data. Large LM's sheer size and black box nature is less

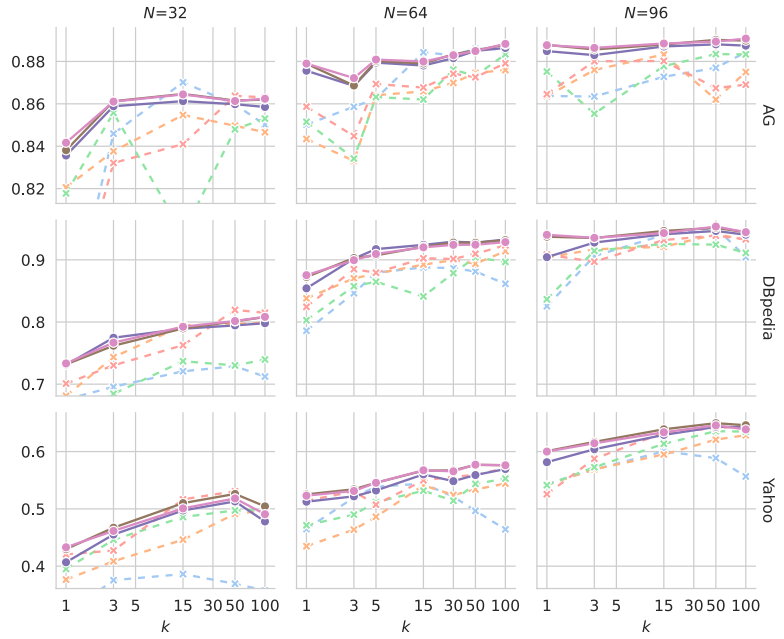

Figure 2: Accuracy of automatic verbalizers by number of label words, on three datasets for $N \in \{32, 64, 96\}$, in function of number of the label words k. Dashed color lines for templates: 0, 1, 2, 3. Solid color lines for ensembles: vote, proba, logit.

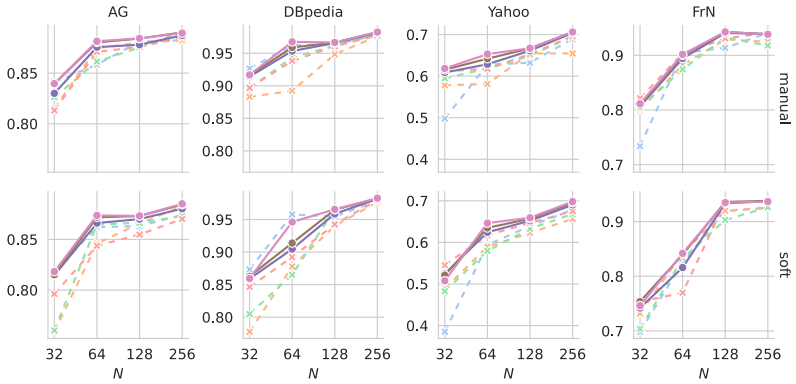

Figure 3: Ensembling templates with manual (first row) and soft (second row) verbalizers across four datasets with varying quantity of data. Dashed color lines for templates: 0, 1, 2, 3. Solid color lines for ensembles: vote, proba, logit.

flexible for custom operations and domain-specific finetuning. In this work, to leverage the importance of verbalizers, using a classic LM may be beneficial for comparing verbalization methods.

In reality, qualified manually annotated text data is indeed hard to achieve, especially in large quantity. Also, manual keywords demand understanding of both the domain and categories, along with the understanding of LMs' charateristics. The insights from this work may benefit non-specialist users, and suggest that increasing the number of label words for the automatic searching algorithm is a simple yet efficient way to compensate for labeling additional data, which is extremely costly.

For future works, it would be interesting to search for other constructions of verbalizer with maximum level of automation, as well as methods to optimize templates for few-shot problems. Studying soft templates and non-tuning methods can be beneficial and effective for applications of verbalizers with large-scale and modern LMs.

7. Acknowledgements

This work is a part of the project "GoldenEYE" at Regulatory IA, Digital & Payments, Groupe BPCE. Their support involves the real-world French dataset and computing resources.

8. Bibliographical References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. KnowPrompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*. ACM.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computa-*

tional Linguistics, pages 7203–7219, Online. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2022. True few-shot learning with Prompts—A real-world perspective. *Transactions of the Association for Computational Linguistics*, 10:716–731.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. 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)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Haode Zhang, Haowen Liang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Y.S. Lam. 2023. Revisit few-shot intent classification with PLMs: Direct fine-tuning vs. continual pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11105–11121, Toronto, Canada. Association for Computational Linguistics.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Differentiable prompt makes pre-trained language models better few-shot learners.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yanan Zheng, Jing Zhou, Yujie Qian, Ming Ding, Chonghua Liao, Li Jian, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang. 2022. FewNLU: Benchmarking state-of-the-art methods for few-shot natural language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 501–516, Dublin, Ireland. Association for Computational Linguistics.

Appendix A. Training details

The LMs are finetuned minimizing the cross entropy loss between predicted probabilities (equations (1) and (2)) with ground true labels, using the AdamW optimizer (Loshchilov and Hutter, 2019). The learning rate is reduced linearly from its maximum value 1×10^{-5} to 0. Each model is finetuned for 10 epochs, with training batch size 4.

The best checkpoints retained base on scores on the validation set. To avoid equality of metrics on $\mathcal{D}_{\text{valid}}$, we involve the validation loss into the compared score:

$$\texttt{score} = \texttt{mean\_of\_metrics} - \frac{\texttt{loss}}{100} \quad (6)$$

Where `mean_of_metrics` is the average of all metrics of the dataset (accuracy and macro F1 for FrN, accuracy for others).

Appendix B. Ensembling strategies

For a data instance x, let $q \in \mathbb{R}^{M \times \mathcal{Y}}$ the logits produced by M models for classes in $\mathcal{Y}$. We consider the following strategies for prediction assembling.

- **vote** taking the majority vote among predictions of M models:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \left| \left\{ j : q_{jy} = \max_{y'} q_{jy'} \right\} \right| \quad (7)$$

- **proba** averaging the normalized probabilities of classes across M models and take the class with maximum probability:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \frac{1}{M} \sum_{j=1}^{M} \frac{\exp q_{jy}}{\sum_{y'} \exp q_{jy'}} \quad (8)$$

- **logit** averaging the logits of classes across M models and take the class with maximum score:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \frac{1}{M} \sum_{j=1}^{M} q_{jy} \quad (9)$$

Appendix C. Templates

Here we specify 4 masked templates for each dataset, which would be processed by the LMs.

- **AG** For the headline $\mathbf{x}$:

$T_0(\mathbf{x}) =$ `MASK` news: $\mathbf{x}$
$T_1(\mathbf{x}) = \mathbf{x}$ This topic is about `MASK` .
$T_2(\mathbf{x}) =$ [Category: `MASK`] $\mathbf{x}$
$T_3(\mathbf{x}) =$ [Topic: `MASK`] $\mathbf{x}$

- **Yahoo** Let $\mathbf{x}$ be the concatenated text of `question_title`, `question_content` and `best_answer`:

$T_0(\mathbf{x}) =$ `MASK` question: $\mathbf{x}$.
$T_1(\mathbf{x}) = \mathbf{x}$ This topic is about `MASK` .
$T_2(\mathbf{x}) =$ [Topic: `MASK`] $\mathbf{x}$.
$T_3(\mathbf{x}) =$ [Category: `MASK`] $\mathbf{x}$.

- **DBpedia** For the title $\mathbf{x}_1$ and the description $\mathbf{x}_2$:

$T_0(\mathbf{x}) = \mathbf{x}_1 . \ \mathbf{x}_2$ In this sentence, $\mathbf{x}_1$ is `MASK` .
$T_1(\mathbf{x}) = \mathbf{x}_1 . \ \mathbf{x}_2 \ \mathbf{x}_1$ is `MASK` .
$T_2(\mathbf{x}) = \mathbf{x}_1 . \ \mathbf{x}_2$ The category of $\mathbf{x}_1$ is `MASK` .
$T_3(\mathbf{x}) = \mathbf{x}_1 . \ \mathbf{x}_2$ The type of $\mathbf{x}_1$ is `MASK` .

- **FrN** Let $\mathbf{x}$ be the concatenated text of `title`, `snippet` and `body`:

$T_0(\mathbf{x}) =$ Nouvelle `MASK` : $\mathbf{x}$
$T_1(\mathbf{x}) =$ Actualité `MASK` : $\mathbf{x}$
$T_2(\mathbf{x}) =$ `MASK` : $\mathbf{x}$
$T_3(\mathbf{x}) =$ [Catégorie: `MASK`] $\mathbf{x}$

Appendix D. Manual verbalizers

Table 2 shows label words for datasets in English, including AG, DBpedia and Yahoo.

Table 3 shows label words for FrN. Words in bold correspond to *Title* in figure 1 where label words are taken directly from the class title. For verbalizer size $k \in \{3, 5, 10, 15\}$ in figure 1, we utilize the k first words in table 3.

Dataset & Classes	Label words
AG	
World	world, politics
Sports	sports
Business	business
Sci/Tech	science, technology
DBpedia	
Company	company
EducationalInstitution	educational, institution
Artist	artist
Athlete	athlete, sport
OfficeHolder	office
MeanOfTransportation	transportaion
Building	building
NaturalPlace	natural, place
Village	village
Animal	animal
Plant	plant
Album	album
Film	film
WrittenWork	written, work
Yahoo	
Society & Culture	society, culture,
Science & Mathematics	science, mathematics
Health	health
Education & Reference	education, reference
Computers & Internet	computers, internet
Sports	sports
Business & Finance	business, finance
Entertainment & Music	entertainment, music
Family & Relationships	family, relationships
Politics & Government	politics, government

Table 2: Label words used for manual verbalizers on datasets in English

Class	Label words
AERONAUTIQUE-ARMEMENT	**aéronautique**, **armement**, flotte, rafale, marine, spatiale, pilote, défense, fusil, satellites, combat, missiles, militaire, réacteurs, hypersonique
AGRO-ALIMENTAIRE	**agroalimentaire**, **agriculture**, agricole, FAO, viticulture, sécheresse, plantation, biodiversité, alimentation, rurale, récolte, bio, terroir, paysanne, céréaliers
AUTOMOBILE	**automobile**, auto, carrosserie, voiture, motorisation, conduite, diesel, pney, mécanique, mobilité, Volkswagen, Renault, berline, concessions, SUV
DISTRIBUTION-COMMERCE	**distribution**, **commerce**, boutique, retail, vitrine, caisse, e-commerce, hypermarchés, ventes, distributeur, soldes, magasin, supermarchés, commercial, dropshipping
ELECTRICITE	**électricité**, energie, energy, éolienne, energetique, photovoltaique, nucléaire, gaz, carbone, combustion, solaire, électronique, generation, centrailes, hydrogène
FINANCE	**finance**, banque, bancaire, monétaire, bce, solvabilité, liquidité, bale, financière, dette, holding, investisseur, investissement, capital, prêts
PETROLE-GAZ	**pétrole**, **gaz**, energie, pétrolière, combustion, géo, forage, réserves, pipeline, oléoduc, gazoduc, rafinerie, liquefié, gisement, bitumeux
PIM	PIM, **immobilier**, foncière, gestion, biens, proprieté, location, **promotion**, projets, permis, programmes, promoteurs, immeubles, chantiers, aménageurs
TOURISME-HOTELLERIE-RESTAURATION	**tourisme**, **hôtellerie**, **restauration**, hotel, restaurant, vacances, vacanciers, séjour, auberges, camping, attraction, touristique, parc, croisiéristes, réservations
TRANSPORT	**transport**, avion, bateaux, ferroviaire, douane, circulation, passagers, aérien, terrestre, maritime, conteneurs, navires, cargos, aéroport, fret

Table 3: Label words used for manual verbalizers on FrN. Bold words are used for results reported in table 1 and *Title* in figure 1. Other words are used for other verbalizer size in figure 1.