Manual Verbalizer Enrichment for Few-Shot Text Classification

Anonymous ACL submission

Abstract

With the emerging and continuous development of pre-trained language models, prompt-based training has become a well-adopted paradigm that drastically improves the exploitation of models for many NLP tasks. Prompting also shows great performance compared to traditional fine-tuning when adapted to zero-shot or few-shot scenarios where the number of annotated data is limited. In this framework, verbalizers play an important role in interpreting masked word distributions produced by language models into output predictions. In this work, we propose MaVEN, a new approach for verbalizer construction by enrichment of class labels using neighborhood relation in the embedding space of words. In addition, we elaborate a benchmarking procedure to evaluate typical baselines of verbalizers for document classification in few-shot learning contexts. Our model achieves state-of-the-art results while using significantly fewer resources. We show that our approach is particularly effective in cases with extremely limited supervision data. Our code is available at `https://anonymous.4open.science/r/verbalizer_benchmark-66E6`.

1 Introduction

Fine-tuning PLM (Devlin et al., 2019a; Zhuang et al., 2021; Brown et al., 2020) resulted in large improvements in various NLP tasks. Classic approaches replace the PLM's output layer with a task-specific head and fine-tune the entire model (Devlin et al., 2019a; Liu et al., 2019; Raffel et al., 2020). However, additional classification layers import a great amount of randomly initialized parameters that need a sufficient amount of labeled data to be trained. Classical fine-tuning, therefore becomes inapplicable for few-shot or zero-shot scenarios (Yin et al., 2019; Zhang et al., 2023).

Motivated by GPT-3 (Brown et al., 2020), prompting has become a novel paradigm where downstream tasks are transformed to suit the pre-training objective. Prompt-based fine-tuning allows to exploit PLMs' knowledge while reducing the gap between pre-training and fine-tuning (Petroni et al., 2019; Chen et al., 2022). In this framework, templates and verbalizers (Schick and Schütze, 2021a; Gao et al., 2021) are crucial elements to map between task-specific inputs and labels, to textual data for the LM. The roles of templates and verbalizers are described as follows. For example, we are given a piece of text:

$$\mathbf{x} = \text{“Dollar rises against euro...”}$$

and the task is to predict if this text belongs to which class among politics, sports, science, or economics. A *template* T first transforms the given text into a cloze-question. For instance, one may choose for this task

$$T(\mathbf{x}) = \text{“\_\_\_ news: Dollar rises against euro...”}$$

The task now changes from predicting a label without textual meaning to identifying whether the most probable choice for the masked position ___ is “politics”, “sports”, “science” or “economics“. This task, namely masked language modeling aligns coherently with the pre-training of a variety of masked LMs, notebly BERT (Devlin et al., 2019b), RoBERTa (Zhuang et al., 2021).

A masked LM takes the wrapped text, marks the missing position with its `MASK` token, and produces probabilities for the masked token over the vocabulary. Ideally in this case, one would expect that the probability of the word “economics” is higher than that of “sports”. In particular, this straightforward approach maps each class to a single word, its textual name. In general, a *verbalizer* refers to this mapping from the label space to the vocabulary space, where each label is mapped to multiple vocabulary tokens.

In many cases, verbalizers are defined *manually* using human knowledge of the downstream task, to choose words that semantically represent the meaning of class labels (Schick and Schütze, 2021a,b; Gao et al., 2021). There are also other constructions of verbalizers such as *soft* verbalizers (Hambardzumyan et al., 2021; Cui et al., 2022). Algorithms for *automatic* label word searching exist in the literature. One such example is PETAL (Schick et al., 2020), where label words are mined based on their likelihood on supervised data. We remark that the procedure presented in (Schick and Schütze, 2021a; Schick et al., 2020) includes semi-supervised learning and therefore additional unlabeled data. One another example is KPT (Hu et al., 2022) where an external knowledge base such as WordNet (Miller, 1994) and ConceptNet (Speer and Havasi, 2012) are used to expand label words from the class name. Our motivation in this work is to propose a method to enrich the manual verbalizer without resorting to external resources.

Our contribution in this paper is summarized as follows:

(i) We propose MaVEN, a new method to enrich the manual verbalizer by neighbors in the embedding space. Our method achieves improved performance over previous work, particularly with an extremely limited amount of data.

(ii) We systematically compare MaVEN to manual, soft, and automatic verbalizers for the text classification task, on three English public datasets previously studied in the literature. We also present new results on two French datasets.

2 Related Work

Prompt-based fine-tuning In this framework, the input is wrapped with a task-specific *template* to reformulate the classification task as language modeling as described in section 1. The *verbalizer* then transforms the distribution of the `MASK` token into label prediction (see section 3 for formal definitions). The choice of textual templates and verbalizer, have a significant influence on the classification performance (Gao et al., 2021).

PET and iPET (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 biases in the `MASK` distribution produced by the PLMs. Otherwise, the search process for an optimal template and verbalizers may be computationally 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 being human-independent and storage-saving, continuous prompts have only been studied in data-abundant scenarios, and produce tokens that are hard to interpret. Here, we study textual templates instead and focus on the search for label words for the verbalizer.

In section 3, we present in detail the *manual*, *soft*, and *automatic verbalizers* as baselines for comparison with our proposed method. Other than these, many constructions of verbalizers exist in the literature. Prototypical verbalizers (Cui et al., 2022) is an improved version of soft verbalizers where contrastive learning helps maximize intra-class similarity and minimize inter-class similarity between embeddings of data instances. PTR (Han et al., 2021) proposes to incorporate logic rules to compose task-specific prompts with several simple sub-prompts.

Enrichment of manual verbalizer Previous works also propose methods to improve the semantics of label words for a given manual verbalizer. KPT (Hu et al., 2022) incorporates external knowledge into the verbalizers, along with multiple steps of refinement and calibration to obtain words with wide coverage of given classes. Still, such knowledge bases may not always be available. Therefore, we are motivated to derive a method to improve the manual verbalizer independently from additional resources. On the other hand, NPPrompt (Zhao et al., 2023) searches for cognates of initial manual words using the embedding layer of the same PLM. This approach attains greater coherence in later PLM fine-tuning. However, NPPrompt is designed and optimized exclusively for zero-shot learning, thus our motivation to develop this idea for few-shot learning by enrichment of manual verbalizers.

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:

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

Where $\mathcal{M}(w|T(\mathbf{x}))$ denotes the logit of MASK being predicted as w by the LM conditional on the masked sequence $T(\mathbf{x})$.

3.1 Baseline Verbalizers

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). In this work, the manual verbalizers derive directly from the names of classes.

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

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

With h the embedding of the 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 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 of positive examples and minimize that of negative examples:

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

Where:

$$\ell(w,\mathbf{x}) = \log\frac{p_{\mathcal{M}}\left(w|T(\mathbf{x})\right)}{1-p_{\mathcal{M}}\left(w|T(\mathbf{x})\right)} \tag{4}$$

Without specifying differently, for comparison analysis, we take $k = 15$.

3.2 Proposition: MaVEN

In this paper, we propose **Ma**nual **V**erbalizer **E**nrichment by Nearest **N**eighbors' Embeddings (MaVEN). Noting that the probability mass that the LM assigns to a specific topic conditioned on the input text is dispersed over multiple label words, we hypothesize that the manual verbalizer captures only a part of this mass and thus is sensitive to the choice of label words. Our motivation therefore is to automatically extend the verbalizer to capture more probability mass by including semantically related words.

In most practical scenarios, a natural manual verbalizer can often be obtained using the names of classes, as class names naturally encode the semantic meaning of texts belonging to the class. We assume that for our classification problem, let v be the initial manual verbalizer. In our case, $v(y)$ includes words extracted directly from the name of the class y. Let E be a word embedding function, the word embedding layer of the LM for example. For each core word $w_0 \in v(y)$, we define the neighborhood of w_0 as:

$$\mathcal{N}_k(w_0) = \{w_0\} \cup \underset{w}{\text{top}-k}\left[s\left(w_0,w\right)\right] \tag{5}$$

Where s is the cosine similarity in this embedding space E:

$$s(w_0,w) = \frac{E(w_0)}{\|E(w_0)\|} \cdot \frac{E(w)}{\|E(w)\|} \tag{6}$$

In case $v(y)$ includes multiple words, we enlarge the verbalizer $v(y)$ as the union of neighborhoods of all initial words:

$$\hat{v}(y) = \bigcup_{w_0\in v(y)} \mathcal{N}_k(w_0) \tag{7}$$

The hyperparameter k represents the size of the neighborhood in the embedding space around the initial core words. In our experiments, without specifying differently, we take $k = 15$.

The probability of the class y is aggregated over its augmented verbalizer as follows:

$$p(y|\mathbf{x}) \propto \exp\left(\frac{\sum_{w\in\hat{v}(y)} q_w^y \mathcal{M}(w|T(\mathbf{x}))}{\sum_{w\in\hat{v}(y)} q_w^y}\right) \tag{8}$$

The weights q_w^y represent the contribution of the word $w \in \mathcal{N}_k(w_0)$ in the class y. Each q_w^y is initialized by the similarity $s\left(w, w_0\right)$ of w to its core word w_0 and fine-tuned with the parameters of the PLM. Comparing to equation (1), observe that instead of averaging uniformly, we adopt a weighted average of the PLM scores, to quantify the relevance of each word in $\hat{v}(y)$ to the class y.

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 and the correct labels. Given the sensitivity of prompt-based methods in a few-shot context, each prompt can be more or less effective towards eliciting knowledge from the PLM. The ensemble approach provides an efficient way to reduce instability across prompts and stronger classifiers (Schick and Schütze, 2021a; Jiang et al., 2020). In this work, we study the impact of aggregating strategy on the performance of assembled models. 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. For the two latter, (Schick et al., 2020) shows that weighted averaging does not gain a clear difference. Thus, we perform uniform averaging.

4 Experiments

4.1 Settings

Five datasets (section 4.2) are considered context for three baselines (section 3) and MaVEN in few-shot prompt-based fine-tuning. For each dataset, 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 (Zheng et al., 2022). The best checkpoint is retained from the score obtained on the validation set. Details of hyperparameters can be found in appendix A.

The underlying pre-trained language model (PLM) is `RoBERTa-large` (Liu et al., 2019) as in (Schick et al., 2020) for datasets in English, or `CamemBERT-large` (Martin et al., 2020) for datasets in French. We report the average and standard deviation of accuracy from 3 repetitions with different samplings of $\mathcal{D}$, to evaluate the result variation with different training data instances. This setup allows us to achieve a robust and global evaluation of learning algorithms.

Our implementation is based on the toolkit OpenPrompt (Ding et al., 2022) and the Transformers package (Wolf et al., 2020). Experiments are executed on two types of GPUs: NVIDIA Tesla V100 and NVIDIA Quadro RTX 5000.

4.2 Datasets and templates

Our experiments are done on three public English datasets and two datasets in French. For each dataset, four textual templates are created, noted T_0, T_1, T_2 and T_3. A summary of these datasets can be found in table 1. The manual verbalizers for each dataset can be found in appendix B.

Dataset	Classes	Test set	Balanced
AG	4	7600	✓
DBpedia	14	75000	✓
Yahoo	10	60000	✓
FrN	10	536	✗
MLSUM Fr	10	10585	✗

Table 1: Dataset details.

AG AG's News (Zhang et al., 2015) is a news classification dataset. Given a headline $\mathbf{x}$, a news need to be classified into one of 4 categories. We define for this dataset:

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

DBpedia The DBpedia ontology classification dataset (Zhang et al., 2015) is constructed by picking 14 non-overlapping classes from DBpedia 2014. Each of these 14 ontology classes contains 40,000 training samples and 5,000 testing samples. Given a title $\mathbf{x}_1$ and its description $\mathbf{x}_2$, the task is to predict the category of the object in the title.

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

Yahoo Yahoo! Answers (Zhang et al., 2015) is a text classification dataset of questions from Yahoo!. Given a question (title and content) and its answer, one of ten possible categories has to be assigned. For a concatenation $\mathbf{x}$ of the question title, question

N	Verbalizer	AG	DBpedia	Yahoo	FrN	MLSUM Fr	Average
0	Majority	25.00	7.14	10.00	16.79	22.80	16.36
	Manual	72.14	73.17	**58.91**	**69.40**	51.45	**65.01**
	Soft	71.89	54.57	52.34	64.74	51.71	59.05
	MaVEN	**72.75**	**74.77**	56.34	62.69	**54.52**	64.21
32	Manual	83.96 ± 2.11	91.68 ± 1.58	**61.84 ± 1.17**	81.16 ± 3.08	58.42 ± 6.44	75.41
	Soft	81.82 ± 3.30	85.95 ± 1.12	50.76 ± 2.84	74.63 ± 5.54	60.53 ± 4.86	70.74
	Auto	**86.44 ± 1.89**	79.24 ± 7.98	50.08 ± 4.39	73.63 ± 1.35	56.38 ± 2.82	69.15
	MaVEN	83.97 ± 2.70	**94.01 ± 1.08**	61.58 ± 3.46	**91.11 ± 1.68**	**60.81 ± 1.93**	**78.30**
64	Manual	**88.14 ± 0.07**	96.75 ± 0.33	65.29 ± 0.98	90.17 ± 2.18	65.79 ± 2.69	81.23
	Soft	87.37 ± 0.45	94.62 ± 2.06	64.64 ± 1.10	84.20 ± 0.88	65.73 ± 2.68	79.31
	Auto	88.00 ± 0.46	92.01 ± 2.92	56.73 ± 5.05	86.38 ± 3.64	**67.17 ± 4.32**	78.06
	MaVEN	87.57 ± 0.88	**97.57 ± 0.29**	**66.17 ± 1.50**	**90.49 ± 3.00**	65.88 ± 3.76	**81.54**
128	Manual	88.43 ± 0.33	96.66 ± 1.14	66.71 ± 0.61	**94.28 ± 1.32**	69.13 ± 0.89	83.04
	Soft	87.32 ± 0.56	96.56 ± 2.00	65.93 ± 0.86	93.47 ± 2.44	68.29 ± 0.84	82.31
	Auto	**88.86 ± 0.10**	95.75 ± 1.87	67.42 ± 0.36	93.47 ± 0.56	**71.28 ± 2.46**	83.36
	MaVEN	88.65 ± 0.57	**97.85 ± 0.10**	**69.18 ± 0.66**	93.28 ± 0.67	68.22 ± 1.43	**83.44**
256	Manual	88.95 ± 0.46	98.24 ± 0.14	**70.63 ± 0.50**	93.84 ± 0.81	71.56 ± 1.54	84.64
	Soft	88.51 ± 0.32	**98.27 ± 0.17**	69.81 ± 0.76	93.66 ± 1.04	70.06 ± 1.09	84.06
	Auto	**89.64 ± 0.58**	98.23 ± 0.28	70.36 ± 1.03	93.16 ± 0.60	**71.65 ± 2.37**	84.61
	MaVEN	89.28 ± 0.55	98.05 ± 0.15	70.29 ± 0.47	**95.46 ± 0.60**	70.59 ± 1.21	**84.73**

Table 2: Accuracy of MaVEN compared to other verbalizers. The ensembling strategy is logit averaging. **Bold** are the best baselines. The last columns is the average accuracy over five datasets. Our proposed MaVEN achieves significant performance gain compared to others for $N \in \{32, 64\}$ and best average performance for few-shot scenarios.

content and the answer, we define:

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

MLSUM Fr Originated from MultiLingual SUMmarization dataset (Scialom et al., 2020), a large-scale dataset obtained from online newspapers. From this dataset, the French split is preprocessed and annotated for the task of topic classification by label grouping, by associating the topic tag of each document to one of ten categories[1].

FrN This real-world private dataset in French is provided by our collaborator in a private company, consisting of press articles. The dataset contains over 5 million articles with silver multi-label annotated among 28 sectors by the data aggregator Factiva[2]. Our collaborators have manually annotated 1,364 articles, of which 1,048 articles belonging to the 10 most frequent sectors are used for experiments in this paper.

For these two last datasets, let $\mathbf{x}$ be the concatenation of the title, the summary, and the body text, we define:

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

4.3 Main Results

Table 2 shows the result over five datasets and three baselines, for different quantity of data N.

For zero-shot learning where no data is available, we observe that MaVEN achieves similar performance to the manual verbalizers, with the exception of FrN.

For extremely low-data settings, such as $N \in \{32, 64\}$, we observe a clear superiority of MaVEN. Compared to the manual verbalizer, MaVEN achieves an improvement of 2.3% on DBpedia, 10.0% on FrN, and 2.4% on MLSUM Fr for $N = 32$. In other cases for $N \in \{32, 64\}$, MaVEN ranks as either the best or the second best among all verbalizers. For larger values of N, the gap between

[1] We follow the grouping procedure presented by reciTAL teams at `https://huggingface.co/lincoln/flaubert-mlsum-topic-classification`.

[2] `https://www.dowjones.com/professional/factiva/`

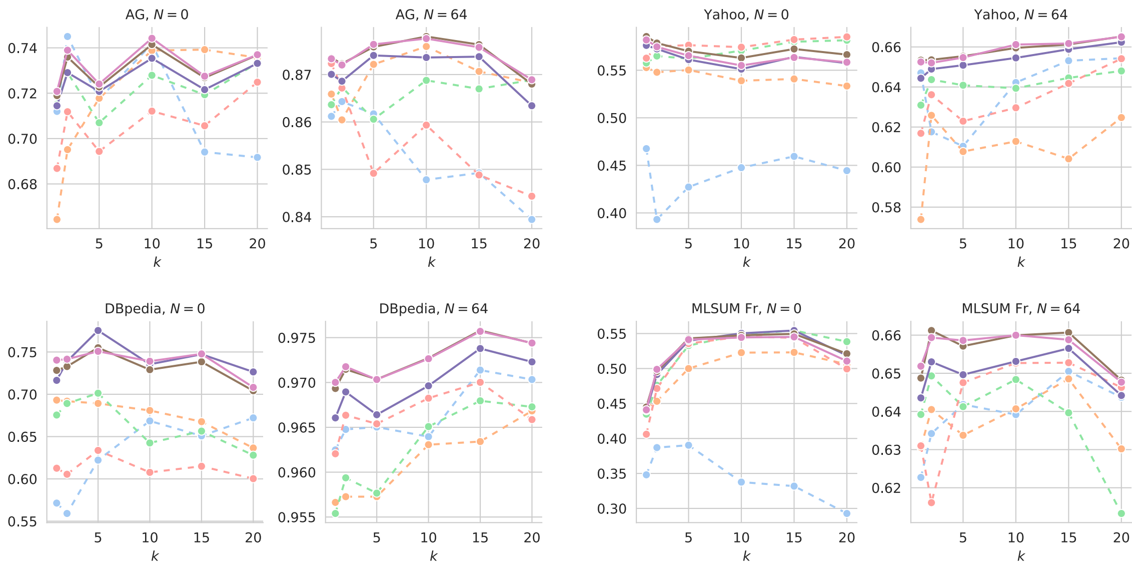

Figure 1: Accuracy of MaVEN by number of label words, on three datasets for $N \in \{0, 64\}$. Dashed colored lines represent templates T : 0, 1, 2, 3. Solid colored lines each represent the ensemble methods: vote, proba, logit.

MaVEN manual verbalizer declines. As more and more training data is provided, the LM learns to attribute the probability mass of a certain class only to the core word and thus, neighbor words become less useful for prediction.

In summary, MaVEN consistently achieves the highest average score across five datasets all few-shot learning contexts. It shows an improvement of 2.9% in average over the manual verbalizer for $N = 32$. For the zero-shot case, it slightly under-performs the manual verbalizer.

The automatic verbalizer performs poorly for cases with extremely low data amounts, such as N. With N increasing, 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). The main reason for this evolution is that automatic label word searching needs supervised training data to mine for label words. With very few labeled data, the choice of label words based on the evaluation of word probabilities may result in errors. Notably, on AG and MLSUM Fr, the automatic verbalizer exceeds the manual verbalizer and MaVEN, which suggests that initial words given by the manual verbalizer of these datasets are biased and less accurate than words extracted from the data, at least from the point of view of the LM.

N	Method	AG	Yahoo
0	PET	69.5	44.0
	Manual	72.14	**58.91**
	MaVEN	**72.75**	57.43
50	PET	86.3	66.2
	Manual	87.26 ± 0.82	66.25 ± 0.37
	PETAL	84.2	62.9
	Auto	**87.54 ± 0.90**	65.89 ± 1.15
	MaVEN	86.35 ± 1.01	**66.59 ± 0.78**
1000	PET	86.9	72.7
	Manual	90.96 ± 0.37	73.07 ± 0.47
	MaVEN	**91.08 ± 0.22**	**74.99 ± 0.28**

Table 3: Comparaison of ours verbalizers to extracted results of PET (manual + unlabeled data + distillation) and PETAL (auto + unlabeled data + distillation).

4.4 Comparison to PET and PETAL

Table 3 compares our implementation of the manual and automatic verbalizer to PET (Schick and Schütze, 2021a) and PETAL (Schick et al., 2020). Note that in addition to prompting and ensemble models, PET and PETAL further introduce self-training with a large amount of unsupervised data (up to 20,000), as a way of knowledge distillation from ensembles to sequence classifiers. Here we omit these elements from the pipeline and instead use part of $\mathcal{D}$ for early stopping[3]. Since we could not find details on the unlabeled data used for PET and PETAL, we extract results shown in (Schick

[3]For $N = 1000$, we split $|\mathcal{D}_{\text{train}}| = 900$, $|\mathcal{D}_{\text{valid}}| = 100$

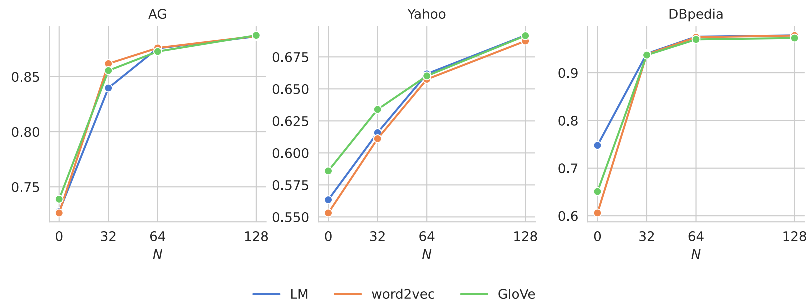

Figure 2: MaVEN accuracy using different embedding spaces (LM, word2vec, GloVe) with varying data amount N.

and Schütze, 2021a; Schick et al., 2020), and only make the comparison for AG and Yahoo. The results show that our implementation allows achieving a competitive level of performance while using significantly less data.

We hypothesize that the improvement of our manual to PET arises from the usage of a part of supervised data for early stopping. Furthermore, in the case of our auto versus PETAL, we also use a larger value of k and experimentally show that this helps raise the accuracy of automatic verbalizers (see section 4.5 and appendix D). Overall, table 3 shows that MaVEN achieves similar or better performance than manual and automatic verbalizers (including PET, PETAL, and our implementation).

4.5 Impact of the Neighborhood Size k

Motivated by remarks in appendix C, in this section, we inspect the impact of the parameter k for the automatic verbalizer and our MaVEN.

Figure 1 shows the dependence of prediction accuracy on k of both individual models and assembled models. For zero-shot prediction, the performance depends significantly on k, fluctuating within a range of 10% for MLSUM Fr and less than 5% for other datasets. With the presence of supervised data, fine-tuned models become more robust with k, where the variation is confined within a margin of about 2% globally, and in particular around 0.6% for DBpedia. In practice, a fixed value between 10 and 15 guarantees a decent level of performance. We also observe that the dependence on k is minor compared to the dependence on the textual template.

For the automatic verbalizers, our analysis in appendix D shows that larger k produce stronger verbalizers and raises the accuracy, which contradicts the conclusion in (Schick et al., 2020) that negates the impact of k. In some cases, using more label words compensates for annotating more data.

4.6 Effectiveness of Ensemble Models

We compare results using individual templates and by assembling the following three methods in figure 1. As observed in most cases, assembled models produce more accurate predictions, even better than the most efficient template. Ensembles also enhance stability and ease the dependence on prompt selection, usually done by large validation sets (Perez et al., 2021), particularly when different templates perform significantly differently. One other significant advantage is that ensemble models tend to be less sensitive to the variation of the neighborhood size k, as discussed in section 4.5.

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

4.7 Effect of the Embedding Space E

In this section, we analyze the importance of the embedding space E in MaVEN. The embedding space intervenes in two major manners: the choice of the neighborhood $\mathcal{N}_k(w_0)$ and the initialization of weights q_w^y via $s(w_0, w)$ (section 3). The vanilla MaVEN utilizes the same embedding layer as the token embedding layer of the fine-tuned LM (`RoBERTa-large` to be precise). Figure 2 demonstrates the performance of MaVEN using different embedding spaces: LM's embedding layer, Google word2vec[5] (Mikolov et al., 2013b,a) and GloVe[6] pre-trained on Wikipedia and Gigaword (Pennington et al., 2014).

In zero-shot context, we observe a significant difference in model performance for the three embeddings. We remark that the range of variation is positively correlated to the number of classes for

[5] https://code.google.com/archive/p/word2vec/
[6] https://nlp.stanford.edu/projects/glove/

Embedding	LM[4]		word2vec		GloVe	
sports	_Sports	0.7727	sport	0.6915	sport	0.7274
	_sport	0.7537	sporting	0.6360	sporting	0.5801
	_sporting	0.6824	Sports	0.6295	basketball	0.5788
	_athletics	0.6536	DeVillers_reports	0.6123	soccer	0.5734
	_sports	0.6527	athletics	0.6093	baseball	0.5572
	Sports	0.6479	football	0.5927	football	0.5556
	Sport	0.6198	sporting_events	0.5816	espn	0.5110
	_athletic	0.6132	soccer	0.5805	athletics	0.5071
	_athletes	0.6090	al_Sunaidy	0.5768	athletic	0.5070
	_SPORTS	0.6086	baseball	0.5658	entertainment	0.5062
	_football	0.6076	limited edition_MGTF	0.5636	hockey	0.4972
	_soccer	0.5956	OSAA_oversees	0.5610	news	0.4953
	_basketball	0.5938	motorsports	0.5515	athletes	0.4897
	_tennis	0.5873	athletic	0.5434	golf	0.4781
	_baseball	0.5846	writers_Jim_Vertuno	0.5395	tennis	0.4762
science	_Science	0.8053	faith_Jezierski	0.6965	sciences	0.6844
	_scientific	0.7044	sciences	0.6821	physics	0.6518
	_sciences	0.7001	biology	0.6776	scientific	0.6487
	science	0.6901	scientific	0.6535	biology	0.6283
	_scientists	0.6895	mathematics	0.6301	mathematics	0.6216
	_scientist	0.6889	Hilal_Khashan_professor	0.6153	research	0.6128
	_physics	0.6700	impeach_USADA	0.6149	technology	0.6056
	Science	0.6638	professor_Kent_Redfield	0.6144	fiction	0.5882
	_biology	0.6482	physics_astronomy	0.6105	professor	0.5873
	_neuroscience	0.6223	bionic_prosthetic_fingers	0.6083	chemistry	0.5856
	_astronomy	0.6094	professor_Burdett_Loomis	0.6065	university	0.5850
	_mathematics	0.5957	Board_BONU_specialty	0.6063	engineering	0.5757
	_scientifically	0.5897	Science	0.6052	psychology	0.5684
	_Sciences	0.5796	portal_EurekAlert	0.5958	institute	0.5678
	_chemistry	0.5720	Shlomo_Avineri_professor	0.5942	literature	0.5656

Table 4: 15 nearest neighbors of "sports" and "science" constructed from three word embeddings: LM, word2vec, and GLoVe, with their respective similarities to the corresponding core words.

the considered problem. For example, the magnitude of this range of variation is 1% for AG with 4 classes, 3% for Yahoo with 10 classes and up to 15% for DBpedia with 14 classes. We remark that using the LM embedding surpasses word2vec and GloVe by a large margin on DBpedia, and works similarly to others in other cases.

When supervised data is available for few-shot fine-tuning, we observe a convergent trend for the three embeddings. As the amount of data increases, the difference in performance of models built from different embeddings reduces. For $N = 128$, the variation due to embedding space of MaVEN is less than 0.5% The role of the embedding space is minimized with the quantity of supervised data.

Table 4 presents the neighborhood of 15 nearest tokens provided by three embedding spaces for two example core words "sports" and "science". For the LM embeddings, most extracted neighbor tokens are spelling variants (e.g. "Sport" vs "Sports"), case-intensitive variants (e.g. "_Sports" vs "_sports") or morphological variants (e.g. "_sports" vs "_sport") of the given core words. In some other cases, the neighborhood also includes tokens deriving from the same origin (e.g. "science", "scientific" and "scientist"). This phenomenon is observed partly in GloVe and even less in word2vec. Tokens extracted from GloVe space are semantically related to the core words, providing global coverage of the topic of the considered class. Meanwhile, some tokens extracted by word2vec are rare combinations of words, proper nouns, etc., that are less meaningful to the considered class. This could be a potential explanation for the poor performance of word2vec in many cases in figure 2.

5 Conclusion

In this paper, we propose MaVEN, a novel method to extend the manual verbalizer that is effective for few-shot learning via prompt-based fine-tuning of PLMs. By leveraging the neighborhood relationship in the embedding space of PLMs, MaVEN was able to identify words related to the topic title to construct verbalizers without the need for data or external knowledge. Experiments show that MaVEN outperforms other constructions of verbalizer for extremely few-shot contexts.

6 Limitations

As an extension of the manual verbalizer, MaVEN requires some initial core words that contain the semantics meaning of the class. Our method, therefore is not applicable if class names are not meaningful description of the classes.

The formulation and construction of verbalizers studied in this work focus on masked LMs, which are exploited only in encoder mode. Meanwhile, recent released PLMs (GPT Brown et al., 2020, LLaMA Touvron et al., 2023, Falcon Almazrouei et al., 2023, etc.) are auto-regressive models that are more powerful on a variety of benchmarks. This leaves the potential to adapt these verbalizer constructions for generative fine-tuning, to exploit these models in decode mode, with the intention to exploit fully the rich knowledge incorporated in these large LMs.

Our work includes datasets and verbalizers in English and French only. It is not sure how well our conclusions generalize to other languages. Other works may need to be carried out in other languages, or more research on verbalizers with multilingual models can be explored.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

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. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. OpenPrompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland. Association for Computational Linguistics.

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.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification.

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.

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.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

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.

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.

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.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2023. Pre-trained language models can be fully zero-shot learners. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15590–15606, Toronto, Canada. Association for Computational Linguistics.

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.

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, Huhhot, China. Chinese Information Processing Society of China.

A Hyperparameters

For simplicity, most choices of hyperparameters are based on existing works and practical considerations. However, these choices could have been done using the validation set.

Parameter	Value
Optimizer	AdamW
Learning rate[7]	1×10^{-5}
Training epochs	10
Batch size	4
Weight decay	0.01
β_1	0.9
β_2	0.999
Gradient accumulation	1

Table 5: Hyperparameters for fine-tuning.

[7] The learning rate increases linearly from 0 to its maximal value for the first 10% steps, then decreases linearly to 0.

B Manual Verbalizers

Here, we specify the label words used for the manual verbalizers of each dataset in table 6 and table 7.

C Preliminary experiments on FrN

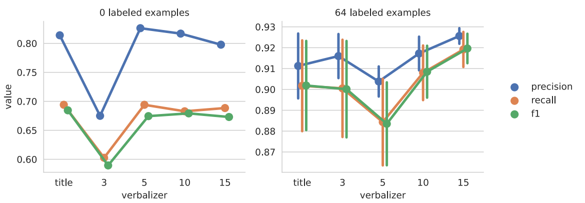

Figure 3: Study of different sizes for the manual verbalizer on the FrN dataset. `title` means using words in class names as label words.

We examine the FrN dataset in zero-shot and in few-shot context with $N = 64$, with the manual verbalizer provided by our collaborators of 15 words per class. By retaining the k most important words (see table 7), we observe the influence of the number of label words. Figure 3 shows a clear improvement from 5 label words for zero-shot and 10 for few-shot. Moreover, few-shot models 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 inspect this phenomenon for an automatic search algorithm, such as PETAL or MaVEN.

D Effect of Verbalizer Size k on Automatic Verbalizers

Figure 4 illustrates the performance of the automatic verbalizer while varying the number k for label word searching. In general, increasing k produces more efficient verbalizers and raises the accuracy for limited data, where 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.

In some cases, notice that using more label words may compensate for annotating more data as a cheaper alternative strategy, from a pragmatic perspective. 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 $N = 64$.

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
MLSUM Fr	
Economie	économie
Opinion	opinion
Politique	politique
Societe	société
Culture	culture
Sport	sport
Environement	environement
Technologie	technologie
Education	éducation
Justice	justice

Table 6: Manual verbalizers of AG, DBPedia, Yahoo, and MLSUM Fr.

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 7: Manual verbalizer of FrN, provided by our private company collaborator. **Bold** words indicates in `title` figure 3.

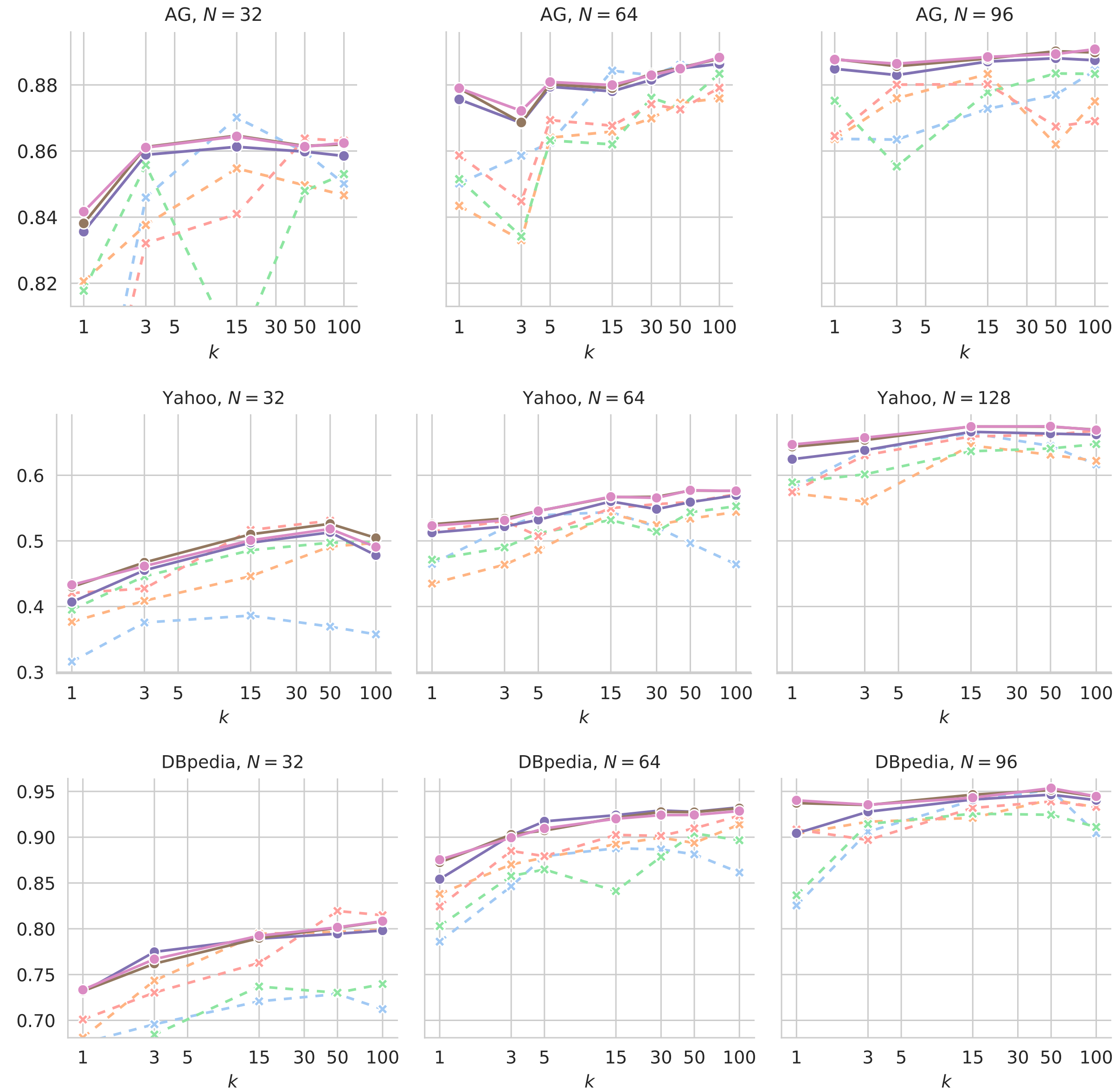

Figure 4: Accuracy of automatic verbalizers by number of label words, on three datasets for $N \in \{0, 64\}$. Dashed colored dashed lines represent templates T : 0, 1, 2, 3. Solid colored lines each represent the ensemble methods: vote, proba, logit.