A Benchmark on Extremely Weakly Supervised Text Classification: Reconcile Seed Matching and Prompting Approaches

Zihan Wang[1*] **Tianle Wang**[2*] **Dheeraj Mekala**[1] **Jingbo Shang**[1†]

[1] University of California, San Diego

[2] Shanghai Jiao Tong University

{ziw224, dmekala, jshang}@ucsd.edu wtl666wtl@sjtu.edu.cn

Abstract

EXtremely Weakly Supervised Text Classification (XWS-TC) refers to text classification based on minimal high-level human guidance, such as a few label-indicative seed words or classification instructions. There are two mainstream approaches for XWS-TC, however, never being rigorously compared: (1) training classifiers based on pseudo-labels generated by *(softly) matching seed words* (SEED) and (2) *prompting (and calibrating)* language models using classification instruction (and raw texts) to decode label words (PROMPT). This paper presents the first XWS-TC benchmark to compare the two approaches on fair grounds, where the datasets, supervisions, and hyperparameter choices are standardized across methods. Our benchmarking results suggest that (1) Both SEED and PROMPT approaches are competitive and there is no clear winner; (2) SEED is empirically more tolerant than PROMPT to human guidance (e.g., seed words, classification instructions, and label words) changes; (3) SEED is empirically more selective than PROMPT to the pre-trained language models; (4) Recent SEED and PROMPT methods have close connections and a clustering post-processing step based on raw in-domain texts is a strong performance booster to both. We hope this benchmark serves as a guideline in selecting XWS-TC methods in different scenarios and stimulate interest in developing guidance- and model-robust XWS-TC methods[1].

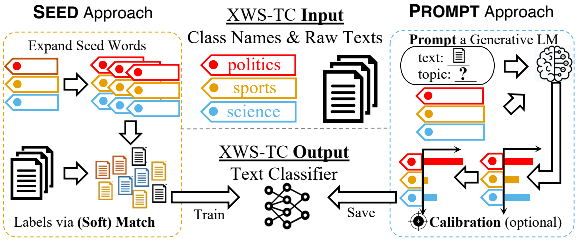

Figure 1: Illustrations of the XWS-TC problem and the SEED and PROMPT approaches.

1 Introduction

Recently there has been a significant advancement in the text classification with the emergence of Extremely Weakly Supervised Text Classification (XWS-TC) methods (Meng et al., 2020b; Wang et al., 2021; Zhang et al., 2021b; Zhao et al., 2022; Park and Lee, 2022), which requires no human-annotated datasets. Instead, these methods rely on minimal human guidance, such as the names of the classes or instructions describing the classification task. There are two main approaches to XWS-TC: one based on matching seed words (SEED), and the other on prompting a language model (LM) with instructions (PROMPT). We give a brief introduction in the following paragraphs, and a more thorough review is in Section 3.

SEED methods for XWS-TC rely on a user-specified list of *seed words* for each class, as well as an unlabeled in-domain corpus. These seed words are then expanded into a larger set of *related words* for the class through statistical methods (Mekala and Shang, 2020), embedding similarity (Wang et al., 2021), or masked language model predictions (Meng et al., 2020b). These related words are used to assign a pseudo-class to each text in the unlabeled corpus through some matching strategy (e.g., assign a text to a class if it contains the related words for that class). The pseudo labels are then used to train a classifier through standard fully-supervised fine-tuning.

On the other hand, PROMPT methods for XWS-TC, rely on reformulating text using an instruction template and prompting the language model to generate the likelihoods for each label in the classification task (Brown et al., 2020). For example, in a sentiment classification task, using an

[*] Equal Contribution.

[†] Corresponding Author.

[1] Github repo at https://github.com/ZihanWangKi/x-TC

Findings of the Association for Computational Linguistics: ACL 2023, pages 3944–3962
July 9-14, 2023 ©2023 Association for Computational Linguistics

instruction template of `<text>. sentiment:`, the model generating "happy" or "sad" will help classifiy the sentiment of the text. Naive zero-shot prompting considers the highest likelihood label as the answer and recent improvements for more accurate likelihoods include calibration of likelihood scores (Holtzman et al., 2021; Zhao et al., 2021; Han et al., 2022) and verbalizers that find more label words to better represent the class (Schick and Schütze, 2021; Ma et al., 2023; Hu et al., 2022).

Both SEED and PROMPT methods have demonstrated strong performance in XWS-TC. However, there has been a lack of comprehensive comparison between these two approaches. This is due to the perception that the approaches are unrelated and the lack of standardization in datasets, supervision, and hyperparameter choices across methods.

We are motivated to construct a benchmark that fairly evaluates the performance of XWS-TC methods. The benchmark consists of 11 datasets covering four domains along with their fine-grained variants and different numbers of classes. In addition, we make an effort to use the same hyperparameters across datasets for the methods, as there should not be a development set to tune the hyperparameters in the XWS setting (Perez et al., 2021).

Our benchmarking results suggest that both SEED and PROMPT approaches are competitive, with no clear winner. SEED tends to perform better when both approaches use a similar-sized pre-trained model and is more robust and tolerant to changes in human guidance (such as seed words, classification instructions, and label words). On the other hand, PROMPT methods have the ability to handle more general types of human guidance (such as descriptions of class names, rather than specific words) and do not have a strict requirement for an unlabeled corpus. When the underlying pre-trained language model changes, PROMPT is more robust and scales better with the language model than SEED. We also examine two specific methods from each approach, X-Class (Wang et al., 2021) and ProtoCal (Han et al., 2022), which independently proposed a post-processing approach to calibrate the class predictions through clustering on an unlabeled in-domain corpus to improve classification performance. Our results show that this subroutine can be a universal booster for both SEED and PROMPT approaches.

Through this benchmark, we aim to advance the study of XWS-TC methods and call for the development of methods that are robust to different human guidance and language models. We firmly believe that this paper will serve as a guide for selecting the appropriate method in different scenarios and contribute to the advancement of the field.

2 Related Work

2.1 Different Types of Weak Supervision

Extremely Weak Supervision is a setting that assumes access to only high-level human inputs, such as names of classes or instructions about classification criteria. We briefly discuss different types of minimal supervision in the following paragraphs.

Few-shot Supervision Few-shot supervision is the setting where there are only a small number of labeled examples for each of the classes. An intuitive way is to directly train the classifier on few-shot data, but usually that yields subpar performance. Another popular way is called *in-context learning*, where the few-shot supervision is used as *context* to prompt LM for the answer (Brown et al., 2020). Various methods have been proposed to improve it by searching for better label words (Schick and Schütze, 2021; Ma et al., 2023), stabilizing the output (Lu et al., 2022), and efficient fine-tuning (Gao et al., 2021).

Distant Supervision Distant supervision includes supervision from external resources such as encyclopedias or gazetteers. There have been efforts to incorporate external knowledge into prompting (Hu et al., 2022), phrase mining (Shang et al., 2018), and named entity recognition (Liang et al., 2020). External models can also be used to help with extremely weak supervision. A line of research is on leveraging models trained on natural language inference data to suggest better-related words (Park and Lee, 2022) or directly classify the text (Yin et al., 2019; Gera et al., 2022).

No Supervision Unsupervised methods fall into this category where they require no supervision. These methods typically take one of the two following approaches: (1) clustering (Aharoni and Goldberg, 2020), (2) topic modeling (Blei et al., 2003). However, both of these approaches lack control over the clusters/topics generated i.e. classes. For example, a text corpus can be categorized on several basis including topic, location, and sentiment. An unsupervised method cannot handle such scenarios. It would be beneficial to be able to retrieve all possible classifications of a corpus in an unsupervised manner, but as far as we are aware, there are no methods with this ability.

2.2 Weak Supervision Benchmarks

We introduce two other Weak Supervision Benchmarks and talk about differences with this work.

Wrench (Zhang et al., 2021a) is a benchmark that explored various types of weak supervision labeling functions (i.e., rules used to label the text). They synthesize the performance of different labeling functions, ways to combine them, and the fine-tuning process to learn the pseudo-training data. In our benchmark, we analyze extremely weak text classifiers that go beyond the labeling functions and compare their performance and robustness with zero-shot prompting.

AutoWS-Bench-101 (Roberts et al., 2022) is another benchmark that analyzes how labeling functions help text classification along with additional few-shot supervision. They conclude that pre-trained models are strong baselines for in-domain settings and should be considered integrating with weak supervision methods. In this work, we focus on extremely weak supervision methods without any labeled data. The SEED and PROMPT methods compared in this benchmark are all based on pre-trained language models.

2.3 Verbalizers

Verbalizers are a type of PROMPT method that find a larger set of label words so that the class choices are accurately represented. We did not consider Verbalizer methods in this benchmark since they mostly rely on additional supervision, such as few-shot (Schick and Schütze, 2021; Ma et al., 2023) or an external knowledge base (Hu et al., 2022).

3 Background

Extremely Weak Supervision in Text Classification refers to a few high-level human guidance as supervision. This guidance typically is in the form of seed words that describe each class, or an instruction paired with label words that define the task. There are two main approaches for XWS-TC: matching seed words (SEED) and prompting language models (PROMPT).

3.1 Seed Matching Methods

SEED approaches are provided with a few class-indicative seed words and unlabeled documents as input. These methods typically involve seed word expansion where more words related to provided seed words are identified in the unlabeled corpus through several statistics-based (Salton and Buckley, 1988; Mekala and Shang, 2020) or deep learning-based strategies (Meng et al., 2020b; Wang et al., 2021; Zhang et al., 2021b). Using these expanded seed words, each unlabeled document is pseudo-labeled. Different heuristics have been explored for pseudo-labeling such as string-matching (Meng et al., 2018). Recently, the matching approach has also evolved into softer manners such as embedding-based matching (Wang et al., 2021), and graph-based matching (Zhang et al., 2021b), that can address conflicts in a principled manner during pseudo-labeling.

We introduce 4 strong-performing SEED methods to include in our benchmark.

LotClass (Meng et al., 2020b) obtains related words through predicting masked tokens in a masked language modeling trained model (Devlin et al., 2019), over an unlabelled corpus. They match the text to related words by fine-tuning a model to predict the related words given a text.

XClass (Wang et al., 2021) obtains related words by finding words that have similar representations. They construct class-oriented representations for text. and match the text to related words by representation similarity. They also showed that the performance can be improved significantly by matching based on clusters from text representations.

ClassKG (Zhang et al., 2021b) models the dependence of related words as an annotating problem on the keyword graph.

NPPrompt (Zhao et al., 2022) obtains related words through embedding similarity from a pre-trained LM. The related words are used as label words to prompt a generative LM for predictions, which are then aggregated as the matching result. To some extent, NPPrompt belongs to an intersection of PROMPT and SEED methods.

3.2 Prompt Methods

Prompting language models is another approach to extremely weak supervision in text classification. This approach involves prompting a generative language model with an instructive text and extracting the *likelihoods* of different label words. This approach does not require an unlabeled in-domain corpus and can be used to predict text in an online fashion. However, language models have been known to be biased towards text sequences more common in pre-training data, leading to instability in zero-shot & few-shot settings. Recently proposed post-processing methods (Holtzman et al., 2021; Han et al., 2022) have attempted to address this by calibrating the predicted probabilities using estimates of the model's bias towards each verbalized label. We describe 2 calibration methods.

DC-PMI (Holtzman et al., 2021) considers a null prompt to obtain the raw likelihoods of language model to predict each label. Then, for each text, they modify the likelihood of the predicted label by marginalizing the raw ones.

ProtoCal (Han et al., 2022) considers an unlabelled corpus and obtains the predicted likelihoods on the corpus. The likelihood vectors are then clustered to better obtain the prediction boundary for each class. Instead of maximum likelihood, this prediction boundary is used to predict the class.

Some more SEED and PROMPT methods are described in Appendix A.

4 Benchmark

In order to establish a benchmark that can accurately evaluate various XWS-TC methods, it is essential to consider a range of factors: Dataset choices, Instructions, Label words, Hyperparameter control, use of Pre-trained Language Models, Metrics and ensure their consistency across all experiments. We will discuss each of these factors in detail in the following sections.

4.1 Dataset

We consider datasets from prior evaluations (Holtzman et al., 2021; Wang et al., 2021; Meng et al., 2020b) that contain data from diverse domains. To facilitate the evaluation process, the size of the evaluation set for each dataset has been controlled to a few thousand instances. Additionally, as many XWS-TC methods require the use of an unlabelled in-domain corpus, a similar-sized sample has been sampled from the training split to serve this purpose, with the evaluation set and unlabelled corpus being disjoint. The datasets have been uniformly sampled without altering the distribution of labels, thus preserving the imbalance ratio, which is defined as the ratio between the size of the largest class and the smallest class. The statistics of the datasets are presented in Table 1. Details of the sources of the datasets are in Appendix B.

4.2 Instructions and Label/Seed Words

To fairly compare SEED and PROMPT methods, we need to provide equal amounts of human supervision. That means, for SEED methods, we should only allow a single word for each class, matching the amount used for label words. For instructions, we consider simple ones that hint at the classification criteria (Holtzman et al., 2021). Details choices can be found in Appendix C.

4.3 Metrics

For evaluation metrics, we consider the macro F_1 score on a dataset-by-dataset basis, which values each class within a dataset equally. To understand the performance of a method on all datasets, we employ two metrics: the average of the macro F_1 scores, and a ranking-based metric that combines the ranking of methods on each dataset to obtain a scale-prone value (Colombo et al., 2022).

4.4 Hyperparameters

Another crucial aspect of the benchmark is the number of hyperparameters utilized by each method. In the context of extremely weak supervision, we argue that it is unrealistic to use different hyperparameters for different datasets, as doing so would necessitate the use of a separate development set, thereby defeating the purpose of using only high-level human supervision (Perez et al., 2021). Therefore, we slightly tune the hyperparameters on one of the datasets to rule out failing scenarios and then stick with a single choice of hyperparameters throughout all datasets. Under this hyperparameter enforcement, the ideal method should exhibit consistent performance across all datasets.

4.5 Pre-trained Language Models

PROMPT methods use generative language models such as GPT while SEED methods use representation encoding language models such as BERT. To fairly compare methods between these two approaches on XWS-TC, we have to consider the ability of language models as a factor. We use the number of parameters of the pre-trained language model as an approximation of the power of the language model. Since all language models use the transformer as the backbone, this implies that the number of layers and size of hidden states is controlled. A further discussion is in Appendix D.

Name	Domain	# Classes	||Unlabelled||	||Eval||	Imbalance
IMDB	Reviews/Sentiment	2	5000	5000	1.0
Yelp-2	Reviews/Sentiment	2	5600	3800	1.1
Yelp-5	Reviews/Sentiment	5	6500	5000	1.1
AGNews	News/Topic	4	6000	7600	1.0
20News	News/Topic	5	6254	5362	1.9
20News-Fine	News/Topic	17	5589	4792	1.3
NYT-S	News/Topic	5	4578	3925	17.1
NYT-S-Fine	News/Topic	26	4034	3459	96.3
NYT	News/Topic	9	5119	6400	30.7
NYT-Loc	News/Location	10	5119	6400	17.1
DBpedia	Wikipedia/Ontology	14	5600	7000	1.3

Table 1: Dataset statistics in our benchmark.

4.6 Large Language Models

This benchmark specifically excludes the evaluation of (multi-task) fine-tuned language models such as T0 (Sanh et al., 2022), large language models (LLMs) such as GPT3, and human feedback-trained language models like InstructGPT (Ouyang et al., 2022) and ChatGPT because there are no equivalent representation encoding language models for the SEED approaches. We discuss this in more details and include an evaluation of ChatGPT on a single dataset as a reference in Appendix E.

5 Benchmark Experiments

5.1 Main Results

In Table 2 we show the performances of all SEED and PROMPT methods considered in the benchmark across the 11 datasets and report the average macro F_1 performance and the rank score.

Performance of PROMPT Methods We note that the performance of the standalone PROMPT method is about 20 points lower than its counterparts with calibration methods. The use of additional instance independent instructions (DCPMI) or an additional clustering based on unlabelled text (ProtoCal) is crucial for PROMPT methods to work well in XWS (zero-shot) text classification.

Performance of SEED Methods All the SEED methods exhibit strong performance, with X-Class performing stably well across all datasets, and ClassKG performing the best on several datasets, but losing on certain fine-grained datasets.

Comparing PROMPT and SEED Methods First, on the absolute performances, we can see that SEED methods have overall better performance than PROMPT methods, even when appropriate calibration is added for PROMPT methods. However, we can also observe that a larger pre-trained GPT model increases the performance of PROMPT methods quite significantly, while SEED methods have a lower performance improvement when a larger pre-trained language model is used. This effect is further studied in Section 5.2.3.

5.2 Robustness

Through this benchmark, we hope to not only decide which method performs the best, but also analyze under dynamic circumstances, which method is more robust to changes. Different choices of label words/seed words, instructions, and pre-trained language models can happen in real life. Therefore, the robustness of methods when these ingredients are reasonably varied would indicate how stable the method is under variating circumstances. Due to the complexity of multiple runs of each method, we focus on 4 datasets pertaining to different domains, imbalance ratios, and number of classes: Yelp, AGNews, NYT-S, and DBpedia. We leave out two methods, LoT-Class and NPPrompt to save computational resources.

5.2.1 Different Seed/Label words

In Table 3 we explore the effect when a different choice of label words and seed words are used. For example, for Yelp-2, we chose negative/positive, terrible/great bad/good, awful/find, and nasty/nice as the variants. We report the performance of the methods on each of the five choices, and also the aggregated performance over the 4 aforementioned datasets. We notice that PROMPT methods in general have a high instability. While DCPMI and Pro-

Method	Model	IMDB	Yelp-2	Yelp-5	AGNews	20News	20News-Fine	NYT-S	NYT-S-Fine	NYT	NYT-Loc	DBpedia	Average	Rank Score
							PROMPT							
Prompt	GPT2-small	56.42	47.36	7.62	38.42	36.32	28.76	22.45	38.90	33.44	60.32	13.93	34.90	0
	GPT2-medium	35.80	33.57	25.87	69.36	55.16	46.03	54.08	46.14	24.92	79.00	24.52	44.95	1
Prompt + DCPMI	GPT2-small	70.13	65.34	23.01	72.67	61.64	37.45	73.93	63.19	55.20	70.40	51.10	58.55	4
	GPT2-medium	63.24	87.00	11.34	74.13	61.15	52.74	79.80	67.66	58.44	87.35	57.30	63.65	8
Prompt + ProtoCal	GPT2-small	70.35	65.89	23.77	72.66	58.62	36.77	53.69	29.82	55.15	65.80	51.97	53.14	2
	GPT2-medium	70.58	88.60	36.62	75.26	62.58	48.55	51.97	46.85	59.04	72.45	66.46	61.54	9
							SEED							
LoT-Class	BERT-base	58.56	67.96	24.92	73.94	70.57	9.40	61.36	23.05	48.59	67.13	57.98	51.2	3
	BERT-large	81.03	77.03	25.17	68.25	65.71	45.51	44.00	37.11	43.08	80.55	58.04	56.86	5
X-Class	BERT-base	82.89	85.44	28.80	81.81	76.98	58.78	91.94	61.06	67.19	86.38	89.50	73.71	10
	BERT-large	82.05	90.39	31.02	85.91	77.52	59.98	87.53	68.40	68.73	85.77	87.91	75.02	12
ClassKG	BERT-base	88.08	92.21	32.33	88.10	81.72	52.29	84.12	49.59	60.79	92.81	94.75	74.25	13
	BERT-large	90.96	93.10	39.41	87.30	83.84	51.62	80.95	59.95	56.31	91.03	72.74	73.38	11
NPPrompt	Roberta-base	85.19	81.17	14.20	80.42	68.92	48.64	77.76	55.23	64.46	53.85	60.36	62.75	7
	Roberta-large	85.67	93.58	23.45	83.62	69.82	43.33	77.93	35.91	59.96	65.83	47.11	62.38	6

Table 2: Performance of PROMPT and SEED methods on the benchmark with standard models, prompt instructions, label words, and seed word choices. All scores are higher the better.

Method	Model	Yelp-2							Averaged over Datasets		
		default	alt. 1	alt. 2	alt. 3	alt. 4	Median	Average (std)	Median	Average	std
						PROMPT					
Prompt	GPT2-small	47.36	49.34	32.84	58.19	32.24	47.36	43.99 (10.04)	32.88	31.01	6.37
	GPT2-medium	33.57	32.89	32.84	55.10	32.78	32.89	37.44 (8.84)	39.39	40.70	8.77
Prompt + DCPMI	GPT2-small	65.34	57.19	72.80	45.12	56.98	57.19	59.49 (9.27)	61.81	62.46	5.13
	GPT2-medium	87.00	66.65	36.53	75.31	39.23	66.65	60.94 (19.93)	68.56	66.54	7.26
Prompt + ProtoCal	GPT2-small	65.89	54.59	70.43	58.03	63.72	63.72	62.53 (5.63)	64.62	64.03	6.17
	GPT2-medium	88.60	87.31	90.53	80.53	68.59	87.21	83.11 (8.00)	72.17	70.74	8.76
						SEED					
X-Class	BERT-base	85.44	88.01	85.69	62.24	84.33	85.44	81.14 (9.53)	86.18	83.83	5.70
	BERT-large	90.39	89.71	88.70	84.75	85.49	88.70	87.81 (2.27)	83.77	83.36	4.47
ClassKG	BERT-base	92.21	91.71	87.78	91.18	92.47	91.71	91.07 (1.70)	87.71	85.88	4.45
	BERT-large	93.10	93.16	94.13	93.89	92.01	93.16	93.26 (0.74)	84.93	85.40	3.74

Table 3: Performance of PROMPT and SEED methods when the label word/seed word are changed to similar meaning alternatives. We show the performance on 5 choices of label words on Yelp-2 (4 alternatives + 1 default), its median, average, and standard deviation, and the averaged metrics across all datasets.

toCal can remedy the variance a bit, SEED methods are still more robust to changes of seed words.

5.2.2 Different Instructions

A high variance is also observed when the instructions are changed for the PROMPT methods, as in Table 4. A noticeable trend is that when the pre-trained model is larger, while the performance increases, the variance brought by instructions or label words also increases. This could be alarming for PROMPT methods.

5.2.3 Different Pre-trained Language Models

In Table 5 we analyze how changes in pre-trained language models would affect the performance of SEED and PROMPT methods (See Appendix H for the full table). Although SEED performs better than PROMPT, PROMPT methods has a strong increasing trend as the size of the pre-trained language model (e.g., changing from BERT-base to BERT-large). Also, X-Class and NPPrompt fail on RoBERTa and BERT respectively, which we hypothesize is that assumptions made in the methods are not general to all pre-trained language models; for example, the distribution of similarities of representations generated by a language model might be different by models. This scaling trend is a factor that should be taken into selecting methods to use for XWS-TC, when the language model size is different than evaluated in this benchmark.

Method	Model	Yelp-2							Averaged over Datasets		
		default	alt. 1	alt. 2	alt. 3	alt. 4	Median	Average (std)	Median	Average	std
Prompt	GPT2-small	47.36	32.89	37.31	73.11	39.01	39.01	45.94 (14.37)	31.06	32.32	8.40
	GPT2-medium	33.57	33.18	56.77	78.41	42.34	42.34	48.85 (17.08)	38.34	39.11	11.73
Prompt	GPT2-small	65.34	76.96	50.14	48.83	39.53	50.14	56.16 (13.29)	60.00	61.48	6.45
+ DMCPMI	GPT2-medium	87.00	88.03	48.56	79.67	67.76	79.67	74.20 (14.72)	65.26	61.54	14.18
Prompt	GPT2-small	65.89	83.87	60.54	71.23	72.25	72.25	70.76 (7.78)	65.54	64.80	6.23
+ ProtoCal	GPT2-medium	88.60	87.40	57.85	80.13	82.73	82.73	79.34 (11.18)	62.59	62.07	10.85

Table 4: Performance of PROMPT methods when the instructions are changed to similar meaning alternatives. We show the performance on 5 choices of instructions on Yelp-2 (4 alternatives + 1 default), its median, average, and standard deviation, and the averaged metrics across all datasets.

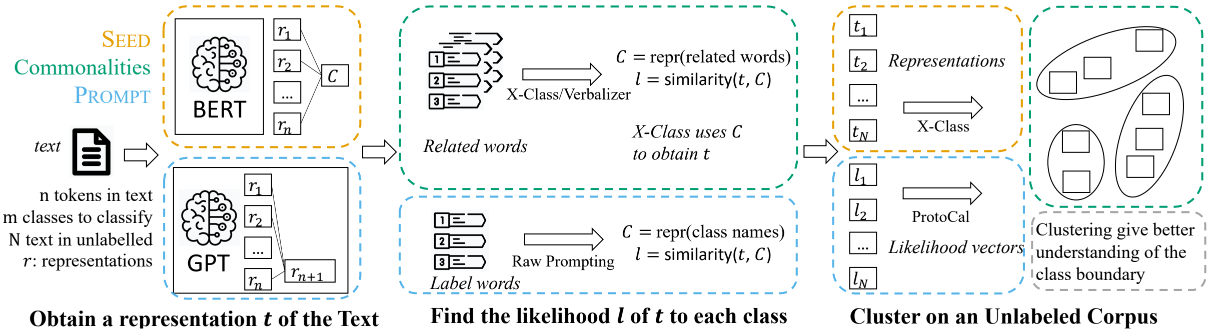

Figure 2: We highlight similarities (green) between a SEED method X-Class (orange) and two PROMPT methods Verbalizers and ProtoCal (blue).

6 Connections between Recent SEED and PROMPT Methods

While PROMPT is introduced by the seminal GPT-3 paper (Brown et al., 2020) not too long ago, SEED has a longer history and can be traced back to early tf-idf retrieval methods (Salton and Buckley, 1988). In recent years, SEED methods and PROMPT methods are exploring similar ideas. SEED methods have been leveraging pre-trained language models to better understand the semantics of seed words; for example, by asking the language model to fill in masks (Meng et al., 2020b) or through means of representation similarities (Wang et al., 2021; Zhao et al., 2022). PROMPT methods have been exploring calibration and verbalizers to improve and stabilize its predictions. Verbalizer includes a step of finding more label words that better represent the class, which is a similar approach used in SEED. We show that a recent representative SEED method X-Class and two PROMPT methods, Verbalizers and ProtoCal have higher similarities and deeper connections in their design. This is particularly interesting as both directions have been developing independently. In Figure 2, we provide a pipeline of the methods and highlight the similarities.

6.1 Obtaining Text Representations

X-Class matches text to classes by learning class-oriented text representations from an encoder-based language model. X-Class views class representations as the union of representations describing the words. The text representation in X-Class is defined as a weighted average of individual token representations where the weights are based on their respective similarity to the class representations. On the other hand, general prompting relies on a decoder-based language model to produce a next token representation. In the penultimate layer of the decoder, the last token representation is computed by an attention mechanism over all other tokens, which essentially produces a weighted average of all the token representations.

In both methods, the text representation is obtained using an attention-like weighted average of tokens in the text. The attention is guided such that the output representation is indicative of the class. X-Class uses signals from class names to guide the attention while prompting relies on the understanding of the instruction.

6.2 Obtaining Predicted Likelihoods

PROMPT methods obtain likelihoods of the class by comparing the similarity of the next token rep-

Method	Model	Average	Rank Score
	PROMPT		
Prompt	GPT2-small	30.54	1
	GPT2-medium	45.38	8
	BERT-base	43.04	7
	BERT-large	51.84	15
	RoBERTa-base	45.71	6
	RoBERTa-large	59.85	22
Prompt + DCPMI	GPT2-small	65.76	24
	GPT2-medium	74.56	31
	BERT-base	60.52	23
	BERT-large	55.88	14
	RoBERTa-base	47.14	5
	RoBERTa-large	55.86	18
Prompt + ProtoCal	GPT2-small	61.05	21
	GPT2-medium	70.07	30
	BERT-base	55.74	11
	BERT-large	70.16	25
	RoBERTa-base	61.07	20
	RoBERTa-large	66.09	28
	SEED		
X-Class	BERT-base	87.17	37
	BERT-large	87.94	39
	RoBERTa-base	60.18	19
	RoBERTa-large	46.78	13
ClassKG	BERT-base	89.80	40
	BERT-large	83.52	38
	RoBERTa-base	86.94	36
	RoBERTa-large	93.17	41
NPPrompt	BERT-base	32.46	0
	BERT-large	31.45	2
	RoBERTa-base	74.93	32
	RoBERTa-large	75.56	33

Table 5: Performance of PROMPT and SEED methods when the choice of the pre-trained model is alternated.

resentation to representations of the label words. A recent line of research on improving prompting for classification is to enlarge the set of label words to capture more diverse meanings of the classes, known as verbalizers, such as PET (Schick and Schütze, 2021), ProtoVerb (Ma et al., 2023), and KPT (Schick and Schütze, 2021; Ma et al., 2023; Hu et al., 2022). The notion of verbalizers is very similar to seed-words expansion in SEED methods. For example, X-Class and verbalizers both obtain a list of related words and use it to aggregate a class representation to replace the naive usage of label/seed word representation. Notably, the verbalizer methods require external supervision to find the related words, such as few-shot data (Schick and Schütze, 2021; Ma et al., 2023) or a knowledge base (Hu et al., 2022) to obtain the related word list, while SEED methods detect related words through an unlabelled corpus. Both approaches could be useful under different input settings.

Method	Model	Average	Rank Score
Prompt	GPT2-small	34.90	0
Prompt + clustering	GPT2-small	53.14	1
Prompt + DCPMI	GPT2-small	58.55	2
Prompt + + DCPMI + clustering	GPT2-small	59.70	3
XClass (w/o clustering)	BERT-base	67.40	6
XClass (w clustering)	BERT-base	73.71	8
NPPrompt	roberta-base	62.75	4
NPPrompt + clustering	roberta-base	64.54	5
ClassKG	BERT-base	74.25	7
ClassKG + clustering	BERT-base	75.16	9

Table 6: Performance of PROMPT and SEED methods with and without the clustering post-processing.

6.3 Unlabeled Corpus Clustering

Finally, a SEED method X-Class and a PROMPT method ProtoCal independently introduced a post-processing step by clustering on an unlabelled corpus, with the goal of obtaining a better decision boundary. X-Class clusters the text representations and initializes the clusters with the prior text-class similarity so that the clusters and classes are aligned. Protocal clusters the predicted likelihoods and align the clusters to classes by post-matching the cluster centers to the classes. We further explore the effect of the two clustering ideas, a summary is in Table 6 (Full table in Appendix I). We show that adding such a post-clustering process to various methods can almost freely (apart from an unlabeled corpus) improve the performance of different methods consistently for five different methods.

6.4 Implications

Given these connections between SEED and PROMPT methods and previous analysis on robustness, a natural extension is to analyze the cause of the stability issues on label/seed words and model differences. We presented one empirical analysis of the clustering step in X-Class and ProtoCal and show that this step can improve performance for various different methods talked about in the benchmark (Section 6.3). Further analysis on other components is left as future work. For example, one could reason that the introduction of related words makes the model less sensitive to the given label/seed words. This would require an exploration of the quality of the related words found by different SEED and verbalizer methods, and whether the related words between methods can be used interchangeably.

7 Conclusions and Future Work

In this work, we introduce a benchmark to qualitatively evaluate different SEED and PROMPT approaches for extremely weakly supervised text classification. Through the benchmark, we raise awareness of the existence of SEED approaches, that are strong competitors to the more well-known zero-shot prompting (with calibrations). We also experiment on the robustness of these two approaches, and show that SEED are more tolerant to the given human guidance changes, however also being more selective to the pre-trained language models. We also analyzed the connections of SEED and PROMPT approaches through the lens of a few representative methods of the two approaches and showed that the methodologies are converging more recently. Finally, we also include a study on clustering as a calibration technique that was independently proposed for both approaches , and show that it can be a good performance booster.

We envision future work in two directions. The first one would be to understand the source of robustness difference and design a method that can take the best of both worlds (see Section 6.4). The other would be to scale up the experiments and test if the conclusions still hold for larger pre-trained language models.

Limitations

Limitation of Model Scale The benchmark only included the evaluation of moderate-size language models and did not experiment on large language models. We justify our reasons in Section 4.6 and Appendix E and include an evaluation of ChatGPT in Appendix E, showing that even human feedback fine-tuned large language models is far from perfect on XWS-TC. However, we acknowledge that the current state of extremely weak supervision would be better understood and assessed if complete evaluations on state-of-the-art large language models, such as Instruct-GPT (Ouyang et al., 2022), PaLM (Chowdhery et al., 2022), and ChatGPT exist. While we lack the computational resources to perform such an evaluation, we hope this work can stimulate interest in XWS-TC and complete the study.

Limitation of Text Classification Another limitation is the scope of Text Classification. While PROMPT and SEED methods have shown strong performances on text classification, this performance does not extend to other general classification tasks, such as natural language inference/entailment (Zhao et al., 2022).

Ethics Statement

This paper establishes a benchmark for extremely weakly supervised text classification frameworks. We provide empirical results on various SEED and PROMPT methods, test their robustness, and analyze their connections. We give intuitions and insights on what method one should use for XWS-TC in different circumstances. We believe that we are on the ethical side and do not find any ethical concerns in this work.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7747–7763. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stéphan Clémençon. 2022. What are the best systems? new perspectives on NLP benchmarking. *CoRR*, abs/2202.03799.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. *CoRR*, abs/2210.17541.

Zhixiong Han, Yaru Hao, Li Dong, and Furu Wei. 2022. Prototypical calibration for few-shot learning of language models. *CoRR*, abs/2205.10183.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7038–7051. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2225–2240. Association for Computational Linguistics.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *ICML*, pages 331–339. Morgan Kaufmann.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. BOND: bert-assisted open-domain named entity recognition with distant supervision. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1054–1064. ACM.

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. *CoRR*, abs/1907.11692.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8086–8098. Association for Computational Linguistics.

Ting Ma, Mingming Li, Shangwen Lv, Fuqing Zhu, Longtao Huang, and Songlin Hu. 2023. Conte: contextualized knowledge graph embedding for circular relations. *Data Min. Knowl. Discov.*, 37(1):110–135.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333, Online. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020a. Discriminative topic mining via category-name guided text embedding. In *WWW*, pages 2121–2132. ACM / IW3C2.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 983–992. ACM.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020b. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 9006–9017. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5316–5330. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Seongmin Park and Jihwa Lee. 2022. LIME: weakly-supervised text classification without seeds. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1083–1088. International Committee on Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11054–11070.

Nicholas Carl Roberts, Xintong Li, Tzu-Heng Huang, Dyah Adila, Spencer Schoenberg, Cheng-Yu Liu, Lauren Pick, Haotian Ma, Aws Albarghouthi, and Frederic Sala. 2022. Autows-bench-101: Benchmarking automated weak supervision with 100 labels. *CoRR*, abs/2208.14362.

Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, 24(5):513–523.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Timo Schick and Hinrich Schütze. 2021. 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, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Trans. Knowl. Data Eng.*, 30(10):1825–1837.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3043–3053. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3912–3921. Association for Computational Linguistics.

Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021a. WRENCH: A comprehensive benchmark for weak supervision. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021b. Weakly-supervised text classification based on keyword graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2803–2813. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*, pages 649–657.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2022. Pre-trained language models can be fully zero-shot learners. *arXiv preprint arXiv:2212.06950*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Other SEED and PROMPT methods

More SEED methods. There are also other SEED methods that we will briefly describe here. WeST-Class (Meng et al., 2018) is one of the earlier weakly supervised methods that utilizes seed words to train a classifier by generating pseudo-documents instead of generating pseudo-labels. Conwea (Mekala and Shang, 2020) explores the multi-sense of words and proposes to view seed words of different meanings as different words. Lime (Park and Lee, 2022) uses a fine-tuned model on a natural language inference dataset to suggest the seed words.

More PROMPT methods. There are also other post/pre-processing techniques that we will briefly describe here. ContextualCal (Zhao et al., 2021) and PromptOrder (Lu et al., 2022) work for in-context learning (in the few-shot scenario), and addresses the stability issue of the few-shot context in prompts. NosiyChannel (Min et al., 2022) considers the likelihood of generating the document based on the label, rather than generating the label based on the document.

B Dataset Sources

The datasets are first introduced in the following papers:

- **IMDB** (Maas et al., 2011).
- **Yelp-2, Yelp-5, AGNews,DBpedia** Zhang et al. (2015)
- **20News, 20News-Fine** Lang (1995)[2]
- **NYT-S, NYT-S-Fine,NYT, NYT-Loc** Meng et al. (2020a)

C Detailed instructions and Label/Seed Words

We provide Table 7 showing the instructions and label words used in the main experiment of the benchmark.

D Comparing Pre-trained Language Models

We are aware that a similar number of parameters in language models do not directly imply similar abilities. We notice that the GPT-family LMs do tend to have a lower fine-tuning performance on natural language understanding tasks (Wang et al., 2019) when compared with BERT/RoBERTa. However, we also notice that similar-sized GPT models do have a similar performance on zero-shot prompting as RoBERTa as observed in Table 8. Since we are comparing under an XWS setting, instead of fully supervised fine-tuning, we believe it is fair to compare similar-size GPT models and RoBERTa models. We do acknowledge that BERT might be at a disadvantage since RoBERTa is better than BERT at both fully supervised fine-tuning (Liu et al., 2019) and zero-shot prompting (Table 8). However, as we note in Section 5.2.3, certain SEED methods that work well on BERT might not be easily transferable to RoBERTa.

E Excluding Large Language Models

We did not include large language models in this benchmark. Here, we elaborate on two specific reasons.

From the design purpose of the benchmark, the focus of the benchmark is to understand the strengths of different SEED and PROMPT methods, which would be fruitful for moderate businesses or individual persons to make decisions on which method to use for XWS-TC. Therefore, the analyses and comparisons on moderate-sized language models (100M - 300M parameters in the benchmark) would be more meaningful.

From a fair evaluation principle, all the models mentioned above are only developed for generative language models, which are not typically used for SEED approaches. Using a more powerful language model for one approach would defeat the purpose of a fair comparison between models. Further, fine-tuned language models have already seen many classification tasks same as or very similar to the datasets in this benchmark. Therefore, it would be hard to access the true performance of the methods, as the similarity of the fine-tuned tasks to the evaluation tasks becomes another factor.

We also include an evaluation of ChatGPT on the benchmark. It is hard to fairly evaluate such a model, since (1) we do not know how it is trained and whether it saw the datasets in the benchmark, and (2) there is no easy way to do large-scale evaluation. We decide to evaluate it on the dataset NYT-S-Fine since we believe it is unlikely it is trained on such a fine-grained dataset. We pick 4 examples from each class resulting in total 104 examples. Since we can not retrieve the likelihoods, we embed the choice of classes in the prompt as follows: `<instruction> <text> Answer:`, where

[2] http://qwone.com/~jason/20Newsgroups/

Dataset	Instruction	Label Words/Seed Words
IMDB	review: <text> sentiment: <label>	positive; negative
Yelp-2	review: <text> sentiment: <label>	positive; negative
Yelp-5	review: <text> sentiment: <label>	excellent; good; average; bad; awful
AGNews	text: <text> topic: <label>	politics; sports; business; technology
20News	text: <text> topic: <label>	computer; sports; science; politics; religion
20News-Fine	text: <text> topic: <label>	atheism; graphics; Microsoft; IBM; Mac; motif; autos; motorcycles; baseball; hockey; encryption; electronics; medicine; space; Christian; guns; Arab
NYT-S	text: <text> topic: <label>	politics; art; business; science; sport
NYT-S-Fine	text: <text> topic: <label>	budget; gun; laws; gay; energy; environment; immigration; military; cosmos; insurance; stocks; bank; abortion; music; baseball; economy; television; golf; tennis; hockey; football; dance; movies; soccer; surveillance; basketball
NYT	text: <text> topic: <label>	business; politics; sports; health; education; estate; arts; science; technology
NYT-Loc	text: <text> location: <label>	America; Iraq; Japan; China; Britain; Russia; Germany; Canada; France; Italy
DBpedia	text: <text> topic: <label>	company; education; artist; athlete; politician; transportation; place; nature; village; species; plant; album; movie; book;

Table 7: Instructions, Label words, and Seed Words.

Method	Model	IMDB	Yelp-2	Yelp-5	AGNews	20News	20News-Fine	NYT-S	NYT-S-Fine	NYT	NYT-Loc	DBpedia	Average	Rank Score
Prompt	GPT2-small	56.42	47.36	7.62	38.42	36.32	28.76	22.45	38.90	33.44	60.32	13.93	34.90	1
	BERT-base	42.16	35.48	7.59	68.89	50.35	3.78	49.94	39.96	37.88	38.49	17.71	35.67	0
	RoBERTa-base	40.51	54.01	15.27	66.94	46.87	12.45	33.27	19.80	38.88	43.92	28.60	36.41	2
	GPT2-medium	35.80	33.57	25.87	69.36	55.16	46.03	54.08	46.14	24.92	79.00	24.52	44.95	4
	BERT-large	46.64	40.91	13.71	71.45	50.20	8.67	38.84	21.12	37.58	37.56	56.17	38.39	3
	RoBERTa-large	86.87	90.54	25.75	76.72	44.89	5.21	33.09	16.29	44.89	59.95	39.03	47.57	5

Table 8: Performance of PROMPT methods with different pre-trained language models.

`<instruction>` is "Choose exactly one of the following classes that best describes the text. Just give the class name as answer, no explanations, nothing more." followed by the list of all class names.

ChatGPT is able to suggest a single-word answer within the set of 26 class names in 91 out of 104 questions; we were able to correct 3 of the 13 out-of-scope answers since they do contain the correct class name. After the correction, ChatGPT is correct on 71 out of 104 questions, making it a model with 68.27% prediction accuracy. The results of X-Class on the same 104 questions is 57.69%. This indicates that while ChatGPT is performing pretty well, there is still much room to improve, given that it is using a much larger language model than X-Class is.

F Method Implementations

We use the public source implementation of different methods.

X-Class `https://github.com/ZihanWangKi/XClass`.

LoTClass `https://github.com/yumeng5/LOTClass`.

ClassKG `https://github.com/zhanglu-cst/ClassKG`.

NPPrompt `https://anonymous.4open.science/r/NPPrompt`.

DCPMI `https://github.com/peterwestuw/surface-form-competition`.

ProtoCal We implemented it ourselves.

G Computation Costs

We ran experiments on A6000 and A5000 GPUs. The total estimated GPU hours is 600.

H Full version of Table 5

We show Table 9, the detailed version of Table 5 that includes performances on individual datasets.

I Full version of Table 6

We show Table 10, the detailed version of Table 6 that includes performances on individual datasets.

Method	Model	Yelp-2	AGNews	NYT-S	DBpedia	Average	Rank Score
PROMPT							
Prompt	GPT2-small	47.36	38.42	22.45	13.93	30.54	1
	GPT2-medium	33.57	69.36	54.08	24.52	45.38	8
	BERT-base	35.58	68.89	49.94	17.71	43.04	7
	BERT-large	40.91	71.45	38.84	56.17	51.84	15
	RoBERTa-base	54.01	66.94	33.27	28.60	45.71	6
	RoBERTa-large	90.54	76.72	33.09	39.03	59.85	22
	BART-base	68.93	52.02	36.11	16.61	43.42	4
	BART-large	89.02	70.89	34.35	27.82	55.52	16
Prompt + DCPMI	GPT2-small	65.34	72.67	73.93	51.10	65.76	24
	GPT2-medium	87.00	74.13	79.80	57.30	74.56	31
	BERT-base	78.46	75.53	51.44	36.63	60.52	23
	BERT-large	78.02	64.38	21.09	60.02	55.88	14
	RoBERTa-base	67.73	59.61	30.96	30.24	47.14	5
	RoBERTa-large	69.42	74.91	39.94	39.16	55.86	18
	BART-base	34.83	45.53	49.68	14.66	36.18	3
	BART-large	55.16	75.13	36.24	41.16	51.92	17
Prompt + ProtoCal	GPT2-small	65.89	72.66	53.69	51.97	61.05	21
	GPT2-medium	88.60	75.26	51.97	64.46	70.07	30
	BERT-base	75.91	65.72	44.65	36.68	55.74	11
	BERT-large	78.18	66.45	57.51	78.52	70.16	25
	RoBERTa-base	82.76	71.34	39.01	51.16	61.07	20
	RoBERTa-large	92.13	78.95	43.29	49.97	66.09	28
	BART-base	86.78	52.94	47.51	23.51	52.68	10
	BART-large	92.18	73.89	50.73	50.83	66.91	27
SEED							
X-Class	BERT-base	85.44	81.81	91.94	89.50	87.17	37
	BERT-large	90.39	85.91	87.53	87.91	87.94	39
	RoBERTa-base	55.06	32.66	61.17	91.85	60.18	19
	RoBERTa-large	38.58	23.91	50.72	73.89	46.78	13
ClassKG	BERT-base	92.21	88.10	84.12	94.75	89.80	40
	BERT-large	93.10	87.30	80.95	72.74	83.52	38
	RoBERTa-base	79.04	88.84	82.98	96.89	86.94	36
	RoBERTa-large	97.13	88.20	91.30	96.04	93.17	41
NPPrompt	BERT-base	37.20	33.89	32.11	11.42	32.46	0
	BERT-large	37.20	33.89	13.49	41.20	31.45	2
	RoBERTa-base	81.17	80.42	77.76	60.36	74.93	32
	RoBERTa-large	93.58	83.62	77.93	47.11	75.56	33

Table 9: This is the full version of Table 5, that includes the performance of PROMPT and SEED methods when the choice of the pre-trained model is alternated. PROMPT methods are evaluated on GPT2, BERT, BART, and RoBERTa, while SEED methods are evaluated on BERT and RoBERTa.

Method	**Model**	**IMDB**	**Yelp-2**	**Yelp-5**	**AGNews**	**20News**	**20News-Fine**	**NYT-S**	**NYT-S-Fine**	**NYT**	**NYT-Loc**	**DBpedia**	**Average**	**Rank Score**
Prompt	GPT2-small	56.42	47.36	7.62	38.42	36.32	28.76	22.45	38.90	33.44	60.32	13.93	34.90	0
Prompt + clustering	GPT2-small	70.35	65.89	23.77	72.66	58.62	36.77	53.69	29.82	55.15	65.80	51.97	53.14	1
Prompt + DCPMI	GPT2-small	70.13	65.34	23.01	72.67	61.64	37.45	73.93	63.19	55.20	70.40	51.10	58.55	2
Prompt + DCPMI + clustering	GPT2-small	70.38	65.84	27.58	78.08	62.40	41.94	82.21	36.88	58.74	63.97	68.64	59.70	3
XClass (w/o clustering)	BERT-base	73.79	83.49	27.48	72.05	74.09	55.35	85.76	55.93	68.57	82.37	62.48	67.40	6
XClass (w clustering)	BERT-base	82.89	85.44	28.80	81.81	76.98	58.78	91.94	61.06	67.19	86.38	89.50	73.71	8
NPPrompt	RoBERTa-base	85.19	81.17	14.20	80.42	68.92	48.64	77.76	55.23	64.46	53.85	60.36	62.75	4
NPPrompt + clustering	RoBERTa-base	84.84	82.99	14.48	83.12	70.42	50.44	91.84	44.10	62.22	54.17	71.32	64.54	5
ClassKG	BERT-base	88.08	92.21	32.33	88.10	81.72	52.29*	84.12	49.59*	60.79	92.81	94.75	74.25	7
ClassKG + clustering	BERT-base	88.86	92.65	40.59	87.19	80.95	54.51*	85.71	52.87*	56.75	91.44	95.20	75.16	9

Table 10: This is the full version of Table 6 that contains the performance of PROMPT and SEED methods with and without the clustering post-processing.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Last page

☑ A2. Did you discuss any potential risks of your work?
Last page

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Yes, first page

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Sec 4.1

☑ B1. Did you cite the creators of artifacts you used?
Appendix B

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
They are open-soruced

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
They are open-sourced for text classification evaluation.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The dataset are not collected by us and is open-sourced.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Sec 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Sec 4.1

C ☑ Did you run computational experiments?

Sec 5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix F,G.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sec 4.4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sec 4.2

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.