Beyond prompting: Making Pre-trained Language Models Better Zero-shot Learners by Clustering Representations

Yu Fei[1], **Zhao Meng**[*1], **Ping Nie**[*2], **Roger Wattenhofer**[1], **Mrinmaya Sachan**[1]
[1]ETH Zurich, [2]Peking University
feiyuwalter@gmail.com, {zhmeng, wattenhofer}@ethz.ch,
ping.nie@pku.edu.cn, mrinmaya.sachan@inf.ethz.ch

Abstract

Recent work has demonstrated that pre-trained language models (PLMs) are zero-shot learners. However, most existing zero-shot methods involve heavy human engineering or complicated self-training pipelines, hindering their application to new situations. In this work, we show that zero-shot text classification can be improved simply by clustering texts in the embedding spaces of PLMs. Specifically, we fit the unlabeled texts with a Bayesian Gaussian Mixture Model after initializing cluster positions and shapes using class names. Despite its simplicity, this approach achieves superior or comparable performance on both topic and sentiment classification datasets and outperforms prior works significantly on unbalanced datasets. We further explore the applicability of our clustering approach by evaluating it on 14 datasets with more diverse topics, text lengths, and numbers of classes. Our approach achieves an average of 20% absolute improvement over prompt-based zero-shot learning. Finally, we compare different PLM embedding spaces and find that texts are well-clustered by topics even if the PLM is not explicitly pre-trained to generate meaningful sentence embeddings. This work indicates that PLM embeddings can categorize texts without task-specific fine-tuning, thus providing a new way to analyze and utilize their knowledge and zero-shot learning ability[1].

1 Introduction

Recent developments in large pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020a) open up the possibility of classifying texts without massive in-task data annotation. Such a zero-shot setting is receiving increasing attention as it is a good way to evaluate the generalizability of knowledge in PLMs. Currently, most existing methods either utilize keywords for self-training (Chang et al., 2008; Meng et al., 2018; Wang et al., 2021) or reformulate the classification task into a cloze task using prompts (Brown et al., 2020; Schick and Schütze, 2021; Gao et al., 2021a). Keyword-based methods usually train multiple modules sequentially (Meng et al., 2020b), while prompting methods depend heavily on human engineering (Liu et al., 2021) or external knowledge (Hu et al., 2021). Such task-specific training or engineering is inefficient and usually does not generalize well to new applications.

In this work, we show that we can better elicit the zero-shot text classification abilities of PLMs simply by clustering texts in their embedding spaces. We draw inspiration from recent findings (Aharoni and Goldberg, 2020) that texts in the same domain (e.g., legal or medical texts) tend to be clustered together in the PLM embedding spaces. This indicates that PLMs already have the knowledge to distinguish texts with different meanings. Following this idea, we propose `SimPTC`: A **Sim**ple **P**robabilistic **T**ext **C**lassification framework building upon state-of-the-art sentence embeddings SimCSE (Gao et al., 2021b). Given an unlabeled dataset and the corresponding class names, `SimPTC` models the texts in each class with a Gaussian distribution and fits the text embeddings with a Bayesian Gaussian Mixture Model (BGMM). To initialize the clusters, we first use the class names to generate class-related anchor sentences. Then the initial cluster assignment of a text is determined according to its similarity to the class anchors in the embedding space.

Despite the simplicity of `SimPTC`, it achieves state-of-the-art performance while avoiding many previously mentioned drawbacks of existing methods: 1) Without self-training of the PLM, `SimPTC` achieves superior or comparable performance on both topic and sentiment classification datasets; 2) Unlike prompt-based methods, `SimPTC` works well

[*]Equal contribution.
[1]Code and datasets available at: https://github.com/fywalter/simptc

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8560 - 8579
December 7-11, 2022

without human engineering or access to external knowledge; 3) SimPTC outperforms previous methods when the dataset is unbalanced. Finally, once we obtain the sentence embeddings, we no longer use the PLM, and SimPTC clusters the embeddings in a fixed dimensional space. Thus, one can easily apply SimPTC to new and large datasets.

To explore the applications and limitations of SimPTC, we compare it with prompt-based zero-shot learning (Schick and Schütze, 2021) on 14 datasets with more diverse topics, text lengths, and numbers of classes. SimPTC gives consistently better performance with a 20% absolute improvement in macro-F1 score on average. We find that SimPTC handles domain-specific rare class names and large class numbers better, while both the prompt-based method and SimPTC suffer when the class names are abstract concepts, e.g., subjective v.s. objective.

Finally, we analyze the embedding spaces of different PLMs using SimPTC. Surprisingly, although RoBERTa$_{\texttt{large}}$ (Liu et al., 2019) is not explicitly pre-trained to generate meaningful sentence embeddings, texts of the same topic are clustered with state-of-the-art zero-shot accuracy. A Larger PLM like T5 (Raffel et al., 2020b) is able to achieve better zero-shot results, even matching the fully supervised performance of BERT (Devlin et al., 2019) on some datasets. On the other hand, SimCSE embeddings separate topics better, and texts of sub-topics can form sub-clusters. On some datasets, we can even observe a linear semantic structure.

To conclude, the strong performance of such a simple clustering-based algorithm suggests that the zero-shot learning ability of PLMs is still under-explored. With SimPTC, we provide a new starting point to utilize and analyze the implicit knowledge and zero-shot learning ability of PLMs.

2 Related Work

In this section, we review three types of zero-shot text classification approaches. Zero-shot text classification aims at classifying texts without any annotated data. This is also referred to as *weakly-supervised* text classification as it can use various weak supervision signals, such as the names or descriptions of the classes, to make predictions.

Keyword-driven methods The most common supervision signal is keywords (Chang et al., 2008; Mekala and Shang, 2020). Meng et al. (2018, 2020b) use iterative self-training on unlabeled in-task data to refine the model or keyword sets. Wang et al. (2021) learn document representations that align with the classes. Zhang et al. (2021b) build a keyword graph to take the connections between keywords into account. Unlike these approaches, SimPTC contains no model training or keyword refinement process and depends solely on the sentence embedding spaces of PLMs.

Clustering-based methods Early clustering-based methods work with discrete text representations such as TF-IDF (Zeng et al., 2003) or bag-of-words (Kyriakopoulou and Kalamboukis, 2006). Recently, ULR (Chu et al., 2021) has explored clustering-based text classification with contextualized sentence embeddings. However, ULR requires fine-tuning the PLM on extra task-related data and uses a heuristic regularization. The K-Means-based approach also places a strong spherical assumption on the cluster shapes. In this work, we show that neither the task-relevant pre-training nor the heuristic designs are necessary. The original embedding spaces of PLMs are sufficient to give strong results with a more flexible clustering algorithm. Nevertheless, it is possible to utilize unsupervised learning to further improve the clustering quality of text representations like in Gupta et al. (2022) and Zhang et al. (2021a). We leave this as a future direction.

Prompt-based methods Prompt-based methods perform zero-shot learning in a natural way by mimicking human behaviors when solving NLP tasks (Brown et al., 2020). Many existing works on prompts focus on text classification, where a template is used to transform the classification task into a cloze task, and a verbalizer maps the predicted words into classification labels (Schick and Schütze, 2021). With carefully designed templates and verbalizers, prompt-based methods can perform comparably to supervised methods in text classification. Various methods have been explored for designing templates (Gao et al., 2021a; Qin and Eisner, 2021) and verbalizers (Cui et al., 2022). Other researchers leverage external knowledge. Hu et al. (2021) expand label names with knowledge bases, and Chen et al. (2022) re-train PLMs by adaptively retrieving extra data.

SimPTC shares the idea of utilizing natural language templates and class names. Nevertheless, instead of reformulating the classification task, SimPTC uses natural language templates and class names to construct class-related texts, which are used to compute initial cluster positions and shapes

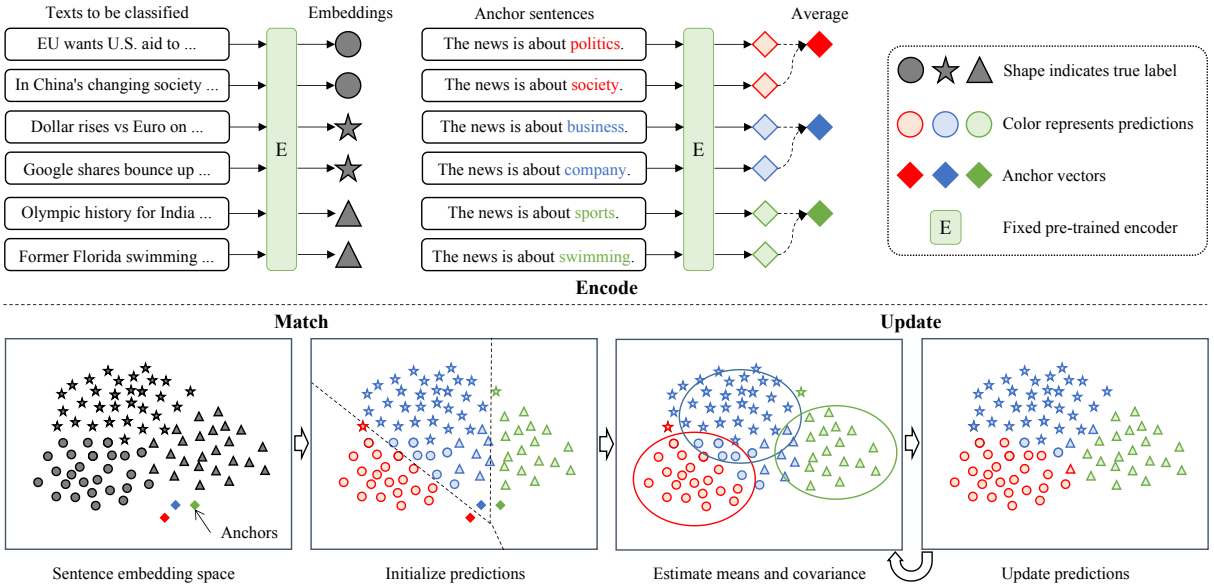


Figure 1: An overview of SimPTC. **Top**: In the Encode step, all unlabeled texts and anchor sentences of each class are encoded using a PLM. Anchor sentences are constructed by combining a template with class names. The anchor sentence embeddings of the same class are averaged to get the final anchor vector. **Bottom left**: In the Match step, the initial cluster assignments are determined based on the cosine similarity between text embeddings and anchor vectors. **Bottom right**: In the update step, we fit the unlabeled data with a BGMM starting from the initial clusters.

for the subsequent probabilistic clustering step.

3 SimPTC

As illustrated in Figure 1, SimPTC formalizes a zero-shot text classification task into a clustering problem and solves it in three steps: Encode, Match, and Update. We start by modeling each class with a Gaussian cluster in the embedding space. Next, the Encode step and Match step provide a coarse initialization of the cluster means and covariances using the class names. Finally, starting from the initialization, we fit the unlabeled data with a BGMM. We elaborate on the three steps of SimPTC below.

3.1 Encode

The first step of SimPTC is to construct class anchor sentences by filling the class names expanded based on external knowledge bases into natural language templates. Then we encode both the unlabeled texts and the class anchor sentences into the PLM embedding space (Figure 1 top).

Expanding class names To make the anchor sentences more class-indicative and less dependent on the exact textual forms of the class names, we expand the class names using external knowledge bases. Specifically, we use ConceptNet Numberbatch (Speer et al., 2017), a set of word embeddings with semi-structured, common sense knowledge from ConceptNet (Speer et al., 2017) combining word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). To extract M related words given a class name s_i, we simply choose the words whose embeddings have top-M largest inner products with the embedding of s_i:

$$S_i = \underset{x \in \mathcal{V}}{\text{top-}M}(\mathbf{x}^\top \mathbf{s}_i),$$

where S_i is the expanded class name set of s_i; $\mathcal{V}$ is the vocabulary; bold font denotes word embeddings. Words that appeared in multiple S_i's are deleted. If $m > 1$ class names are given for one class, for each name we extract M/m words. See Appendix A for extracted word examples.

Constructing anchor sentences We take the idea of using natural language templates from prompt-based methods (Schick and Schütze, 2021) to construct anchor sentences. A *template* is a piece of text containing one or multiple special tokens to be filled in, such as "The text is about $\langle mask \rangle$.". By replacing the $\langle mask \rangle$ token with the expanded class names $s_i \in S_i$, we get a set of class-related sentences. Unlike prompt-based methods, we allow class names with multiple tokens. The anchor embeddings of the same class are averaged to give the final anchor vector (Figure 1 top middle).

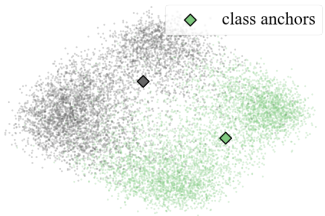


Figure 2: 2D PCA visualization of Amazon dataset in the SimCSE embedding space. **The texts of different classes are well-clustered, and the class anchors from the Encode step reflect the relative positions of text clusters. (Figure 1 bottom left).**

Template	Acc
`Vanilla Prompting`	
$\langle text \rangle$ A $\langle mask \rangle$ news .	31.5
$\langle text \rangle$ [class: $\langle mask \rangle$]	70.3
$\langle text \rangle$ This text is about $\langle mask \rangle$.	68.7
`Encode&Match`	
A $\langle mask \rangle$ news.	**78.9**
[class: $\langle mask \rangle$]	**76.8**
This text is about $\langle mask \rangle$.	**78.2**

Table 1: Accuracy of `Vanilla Prompting` and `Encode&Match` with different templates on AG's News test set. **`Encode&Match` depends less on the choice of template and gives better performance.**

3.2 Match

Let $\{\mathbf{x}_i, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be the embeddings of a set of unlabeled texts of size N, and $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K\}$ be the averaged anchor vectors for K classes. The pseudo-label $\hat{y}_i$ of $\mathbf{x}_i$ are determined by:

$$\hat{y}_i = \arg\max_{j \in [K]} \text{cos-sim}(\mathbf{x}_i, \mathbf{a}_j). \quad (1)$$

$\{\hat{y}_i\}$ are then used to compute the initial cluster means and covariances. We call this pseudo-label-generating process `Encode&Match (E&M)`.

Figure 2 illustrates the encoded anchor vectors $\mathbf{a}_i$'s and example vectors $\mathbf{x}_i$'s after performing PCA. The anchor vectors $\mathbf{a}_i$'s indeed reflect the relative positions of the clusters. To provide more insights, we conduct a pilot experiment on AG's News (Zhang et al., 2015) dataset . We show the zero-shot performance of `E&M` and `Vanilla Prompting`, the vanilla prompt-based zero-shot text classification method used in Schick and Schütze (2021), in Table 1[2]. For a fair comparison, we use the original class names directly to construct anchor sentences for `E&M`. `E&M` provides a competitive initialization and is more stable across different choices of natural language templates.

3.3 Update

Model classes with Gaussian clusters To capture the position and shape characteristics of text clusters, we model the texts of the same class with a Gaussian in the embedding space and define a Gaussian Mixture Model (GMM). Then the likelihood of the dataset is given by

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

where $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the model parameters; $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$, and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K\}$ are the priors, means, and covariances of each component respectively. We can further require all components to share the same covariance matrix to add extra regularization when the data is sparse, or we have additional prior knowledge that the clusters have similar shapes.

Variational update Clustering in a high dimensional space can be challenging, for instance, when the data is limited, or the initialization is poor. One simple solution is to inject prior knowledge, such as assuming a uniform prior on the classes as in several prompt-based methods (Zhao et al., 2021; Hu et al., 2021). However, debiasing model explicitly can be harmful when the prior is incorrect. To balance injecting prior knowledge and fitting the data, we turn to the Bayesian approaches and introduce prior distributions on model parameters. We choose a Dirichlet distribution as the prior for mixture weights $\boldsymbol{\pi}$ to favor balanced weights:

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\alpha_0) = C(\alpha_0) \prod_k \pi_k^{\alpha_0 - 1}$$

where $C(\alpha_0)$ is a normalizing constant, and α_0 can be interpreted as the prior number of observations associated with each class. We simply choose $\alpha_0 = N/K$. For the means and covariances, we choose a non-informative Gaussian-Wishart prior (see Appendix C for details). Then we update the model with the standard variational optimization (Bishop and Nasrabadi, 2006). As BGMM is guaranteed to converge (Boyd et al., 2004), we stop updating when the model predictions stop changing or the maximum number of iterations is reached.

The overall `SimPTC` algorithm is summarized in Algorithm 1. **Note that, in general, we can**

[2] `Vanilla Prompting` results are based on $\text{BERT}_{\texttt{large}}$, and `E&M` uses SimCSE supervised $\text{BERT}_{\texttt{large}}$

replace `Encode&Match` with any initialization method and Bayesian GMM with any clustering algorithm (see discussion in §4.2).

Algorithm 1: SimPTC

Input: unlabeled texts U; test texts U^{test}; class names S; sentence encoder E; max iteration T
Output: The prediction of U^{test}
$\mathbf{X} \leftarrow E(U)$;
$\mathbf{X}^{test} \leftarrow E(U^{test})$;
$\{\hat{y}_i\} \leftarrow$ `Encode&Match` (§3.1 and §3.2);
$M \leftarrow$ BayesianGMM(
  initial predictions $\leftarrow \{\hat{y}_i\}$,
  weight prior $\alpha_0 \leftarrow |U|/|S|$,
  mean & cov prior $\leftarrow$ Eq. (2) in App. C,
  max iter $\leftarrow T$,
);
Fit M with $\mathbf{X}$;
$\{y_i^{test}\} \leftarrow$ prediction of M on $\mathbf{X}^{test}$;
Return $\{y_i^{test}\}$

4 Experiments

We conduct extensive experiments to understand `SimPTC`. We compare `SimPTC` with state-of-the-art zero-shot text classification methods in §4.1, study the effect of its components in §4.2, and explore its applications and limitations on a wide range of tasks in §4.3. For all experiments, we use the SimCSE supervised RoBERTa$_{\texttt{large}}$ embeddings, which are in $\mathbb{R}^{1024}$ and trained using NLI datasets via contrastive learning starting from the original RoBERTa$_{\texttt{large}}$ model. We discuss and analyze other PLMs, such as T5 in §4.4.

4.1 Comparison with State-of-the-art

We evaluate the zero-shot text classification performance of `SimPTC` on five benchmark datasets.

Datasets We use three topic datasets: AG's News (Zhang et al., 2015), DBpedia (Lehmann et al., 2015), and Yahoo (Zhang et al., 2015), and two sentiment datasets: IMDb (Maas et al., 2011) and Amazon (McAuley and Leskovec, 2013). The full dataset statistics can be found in Appendix D.

Implementations Following Hu et al. (2021), we manually design four templates (Appendix B) for every dataset. The number of extracted class-related words for each class is 1000. We fit the BGMM with both the unlabeled train and test data. For topic datasets, each Gaussian has its individual covariance. For sentiment datasets, all Gaussians share the same covariance to provide extra regularization as the data is relatively sparse. The maximum iterations are set empirically based on the size of unlabeled data. See Appendix D for details.

Baselines We compare `SimPTC` with the following methods. `Vanilla Prompting` is the vanilla prompt-based zero-shot text classification without self-training used in Schick and Schütze (2021). We use the original class names and templates designed by Hu et al. (2021) for predicting. `ULR` (Chu et al., 2021) performs zero-shot text classification by clustering data using K-Means with a heuristic regularization. Since `ULR` originally uses an encoder pre-trained with extra in-domain data, we evaluate `ULR` with the same embeddings used by `SimPTC`. `LOTClass` (Meng et al., 2020b) is a state-of-the-art keyword-based method that involves training multiple models with multiple tasks sequentially. `KPT` (Hu et al., 2021) is the state-of-the-art prompt-based method that utilizes external knowledge bases and contextualized calibration to produce stable zero-shot predictions.

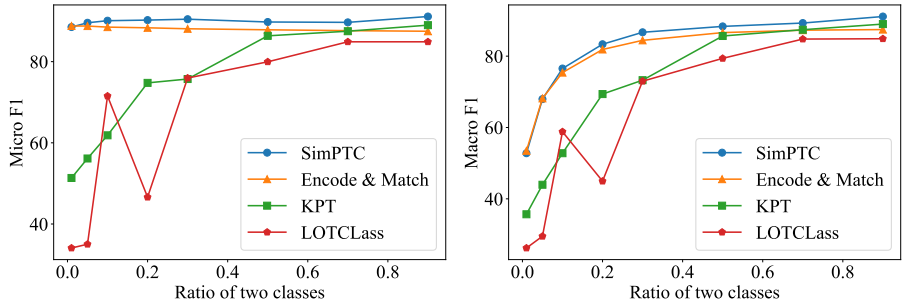
Experimental Design We conduct experiments to evaluate the following three claims:

C1: `SimPTC` achieves superior or comparable performance on both topic and sentiment datasets. Table 2 reports the accuracy on the test sets. We report the average scores with standard deviations for methods using multiple natural language templates. Without fine-tuning the PLM, `SimPTC` presents superior or comparable performance on all datasets. On IMDb, `KPT` gets slightly better results (91.6 v.s. 91.0) but has a much larger standard deviation (2.7 v.s. 0). Moreover, `KPT` improperly poses a balanced dataset assumption (Appendix H), which hurts model performance when the dataset is unbalanced (see **C3**).

C2: `SimPTC` gives stable predictions across different templates. Compared to `Vanilla Prompting`, `E&M` gives a better or comparable performance on all datasets with much lower standard deviations across different natural language templates (Table 2). The observation holds even when we compare `E&M` with the prompt-based method enhanced with external knowledge (`KPT`). `SimPTC` further reduces the standard deviations and improves performance.

Method	AG's News	DBPedia	Yahoo	Amazon	IMDb
Vanilla Prompting	72.1 ± 10.4	80.9 ± 2.3	40.4 ± 3.1	79.7 ± 10.8	81.5 ± 4.1
ULR (Chu et al., 2021)	80.1	79.8	59.6	92.6	82.4
LOTClass† (Meng et al., 2020b)	86.4	91.1	fail	91.6	86.5
KPT† (Hu et al., 2021)	84.8 ± 1.2	82.2 ± 5.4	61.6 ± 2.2	92.8 ± 1.2	$\mathbf{91.6 \pm 2.7}$
Encode&Match (E&M)	78.2 ± 0.3	74.4 ± 1.6	58.3 ± 0.1	91.2 ± 0.1	85.6 ± 0.4
SimPTC	$\mathbf{86.9 \pm 0.3}$	$\mathbf{93.2 \pm 1.0}$	$\mathbf{63.9 \pm 0.1}$	$\mathbf{93.9 \pm 0.0}$	$\mathbf{91.0 \pm 0.0}$
-class name expansion	87.6 ± 0.5	92.9 ± 0.1	63.7 ± 0.1	93.9 ± 0.0	91.0 ± 0.0
-manual templates	86.5	93.3	62.9	93.9	91.1

Table 2: Zero-shot test accuracy on five benchmark datasets. †: We use the number reported in the original papers. Indentation means the configuration is modified based on the up-level indentation. The keyword-extracting module of LOTClass fails on Yahoo.



(a) Micro-F1 score (b) Macro-F1 score

Figure 3: The micro-F1 and macro-F1 score plots of different methods on unbalanced IMDb datasets with different class ratios. **When the dataset is unbalanced, SimPTC consistently performs better, and the gain is substantial in extreme cases.**

C3: SimPTC consistently outperforms prior work when the classes in the dataset are unbalanced. Currently, most benchmark datasets are balanced. Overfitting to this balanced bias reduces the generalizability of the method. To illustrate this problem, we conduct the following experiment on IMDb. We keep the texts of one class with a ratio varying from 0.01 to 0.9 to generate different unbalanced settings, and we compare SimPTC with KPT and LOTClass. KPT injects a balanced dataset assumption directly into its design (Appendix H), and LOTClass is a self-training keyword-based method without an explicit balanced assumption. As shown in Figure 4, the performance of KPT and LOTClass drops significantly as the dataset becomes more unbalanced, whereas SimPTC achieves consistently better performance. As the class ratio approaches zero, the micro-F1 score of KPT goes to 50 since the balanced prior forces the model to make a balance prediction. Although LOTClass is purely data-driven, the data imbalance still dramatically affects its self-training process. On the other hand, E&M provides a strong starting point for SimPTC, and SimPTC further improves its performance.

We discuss the convergence of SimPTC, the effect of unlabeled dataset size, and sharing covariance matrix in Appendix E, F and G respectively.

4.2 Ablations

We try to understand what contributes to the competitive performance of SimPTC by studying the importance of 1) the choice of natural language template and class names, 2) the initialization method, and 3) the clustering algorithm.

4.2.1 Templates and Class Names

SimPTC gives state-of-the-art results even without carefully designed templates or class names extracted using external knowledge. We first evaluate SimPTC using only the original class names for constructing class anchor sentences (-class name expansion in Table 2). SimPTC still gives a comparable performance on all datasets. Then we further test SimPTC with the naive template "$\langle mask \rangle$" (-manual templates in Table 2). The performance is again only slightly affected. Unlike prompt-based methods, which are sensitive to the quality of class names and templates,

Method	AG	DB	YH	AM	IM
VP	72.1	80.9	40.4	79.7	81.5
E&M	78.2	74.4	58.3	91.2	85.6
SimPTC+VP	86.7	92.7	63.4	**93.9**	**91.0**
SimPTC+E&M	**86.9**	**93.2**	**63.9**	**93.9**	**91.0**

Table 3: Comparison of different initialization methods. **SimPTC is fairly robust to the quality of initialization.**

Clustering Algo.	AG	DB	YH	AM	IM
K-Means	75.3	90.5	61.7	92.1	88.3
GMM	76.4	82.9	51.6	**93.9**	89.4
BGMM	**86.9**	**93.2**	**63.9**	**93.9**	**91.0**

Table 4: Comparison of different clustering algorithms. **BGMM outperforms K-Means, while GMM fails to work on many-class tasks like DBpedia and Yahoo**.

SimPTC gives strong performance even without external knowledge or human engineering.

4.2.2 Initialization Method

SimPTC is robust to the quality of initialization. We use E&M to initialize the clusters mainly because E&M works directly with the text embeddings computed for later clustering, adding only minimal additional computations. In general, SimPTC works with any initialization method (see Algorithm 1). As a comparison, we test using Vanilla Prompting(VP) as the initialization. We report the results averaged over four templates on five benchmarks in Table 3. Although VP gives a slightly worse initialization performance, SimPTC achieves a similar performance after clustering, showing the robustness of SimPTC to the initialization method.

4.2.3 Clustering Algorithm

In this section, we aim to show what makes a good choice of clustering algorithm for SimPTC by comparing BGMM with K-Means and GMM.

Modeling cluster shapes is beneficial. As shown in Table 4, BGMM outperforms K-Means on all five balanced benchmark datasets. This shows that putting a strong assumption on the cluster shapes like K-Means limits the clustering step's performance. Since the SimCSE embedding space is rather well-structured, we further test SimPTC + K-Means with the original RoBERTa$_{\texttt{large}}$ embeddings. The performance on IMDb drops from 92.3 to 54.1, indicating that BGMM is a more robust choice for clustering PLM embeddings in general.

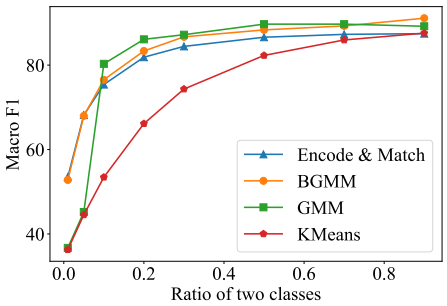


Figure 4: The macro-F1 score plots of different clustering algorithms on unbalanced IMDb datasets with different class ratios. **K-Means cannot handle unbalanced datasets. BGMM and GMM perform better by allowing the cluster weights to adapt to the data, but GMM is less stable in extreme cases.**

Adding prior on cluster weights helps on many-class tasks. Following the previous observation, GMM outperforms K-Means on AG News, IMDb, and Amazon by allowing to model the cluster shapes using data. However, GMM fails on many-class tasks like DBpedia and Yahoo (Table 4), showing the benefits of adding prior on cluster weights as extra regularization.

Learnable cluster weights handle class imbalance. The learnable mixing weights of BGMM (and GMM) model the proportion of classes and therefore handle unbalanced clusters. To test this, we again compare three clustering algorithms on IMDb dataset with different class ratios. Figure 4 shows that K-means fails completely when the dataset is unbalanced. BGMM and GMM perform better by allowing the cluster weights to adapt to the data, but GMM is less stable in extreme cases.

4.3 TC14 Datasets

To further study the potential applications and limitations of SimPTC, we collect 14 publicly available text classification datasets with various topics, text lengths, and numbers of classes (Table 5). For simplicity, we refer to these datasets as TC14. For more dataset information, see Appendix I.1.

Setup To simulate the most basic scenario, we evaluate SimPTC with the naive template "$\langle mask \rangle$" and the original class names without expansion. We choose Vanilla Prompting as the baseline since it is the most widely used zero-shot prompt-based method. For a fair comparison, we do not engineer templates or verbalizers and use the original class names with templates adopted from Hu et al. (2021) (see Appendix I.2 for implementation details).

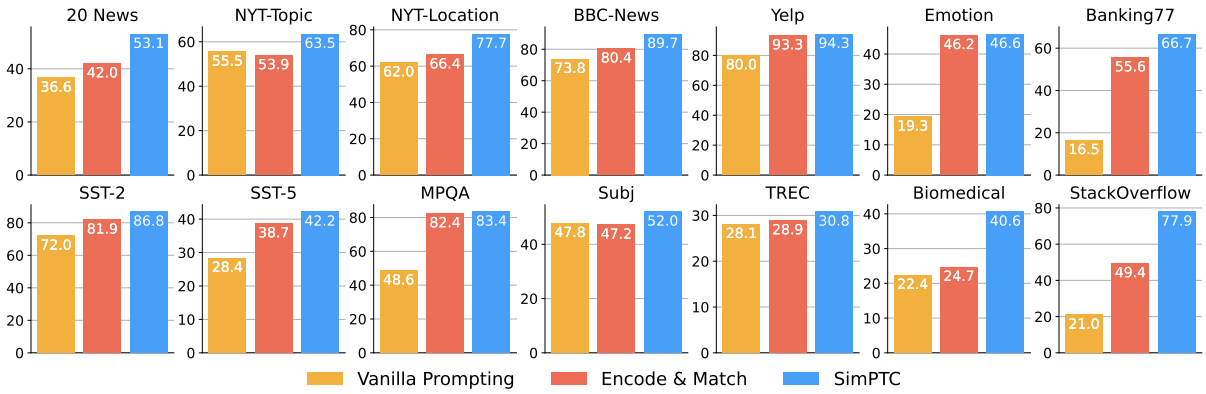


Figure 5: Macro-F1 scores on TC14. **SimPTC outperforms Vanilla Prompting on all 14 datasets.**

Datasets	# Texts	# Cls.	Ave. Len.	Unb.
20 News	18391	20	186	✓
NYT-Topic	31997	9	783	✓
NYT-Location	31997	10	783	✓
BBC News	2225	5	390	✓
Yelp	38000	2	132	✗
Emotion	20000	6	19	✓
Banking77	13083	77	12	✓
SST-2	9613	2	19	✓
SST-5	11855	5	19	✓
MPQA	10606	2	3	✓
Subj	10000	2	23	✗
TREC	5952	6	10	✓
Biomedical	20000	20	13	✗
StackOverflow	20000	20	8	✗

Table 5: TC14 datasets (Cls.: class, Unb.: unbalanced).

Results We report the macro-F1 scores on TC14 in Figure 5 and put micro-F1 scores in Appendix I.3. E&M outperforms Vanilla Prompting on 12 out of 14 datasets. SimPTC further boosts the performance and gives a superior performance on all 14 datasets, showing the strong generalizability of our approach. SimPTC achieves the most gain when 1) the class names contain multiple tokens (e.g., Banking77); 2) the number of classes is large (e.g., StackOverflow); 3) the class names contain rare or domain-specific words (e.g., Biomedical).

When does SimPTC not work very well? Both Vanilla Prompting and E&M suffer when the class names are abstract concepts, e.g., subjective and objective in the Subj dataset. This suggests that prompting and current text embeddings are still poor at linking texts to class names describing abstract properties. But interestingly, the two classes of Subj separate well in the SimCSE embedding space (Figure 6), indicating the ability of PLM embedding spaces to capture abstract semantic concepts. Additionally, both methods underperform

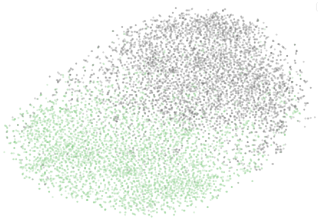


Figure 6: 2D t-SNE visualization of Subj dataset in the SimCSE embedding space. **Although E&M cannot provide a meaningful initial prediction given the abstract class names: subjective and objective, the two classes are well separated in the embedding space.**

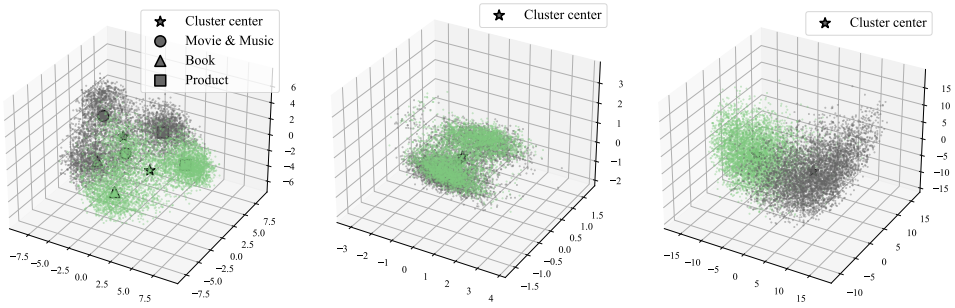
self-training keyword-based methods in long document tasks (see Appendix I.4 for more details).

4.4 Different Encoders

In this section, we utilize SimPTC to analyze different PLM embedding spaces. Specifically, we ask two questions: 1) **Are the texts also clustered by topics in the embedding spaces of PLMs that are not explicitly trained to generate meaningful embeddings?** 2) **Are the embeddings of larger PLMs more informative?** To answer these questions, we compare RoBERTa$_{\texttt{large}}$ (RL) (Liu et al., 2019), Sentence RoBERTa$_{\texttt{large}}$ (SRL) (Reimers and Gurevych, 2019), SimCSE supervised RoBERTa$_{\texttt{large}}$ (SimCSE), and T5-3B (Raffel et al., 2020b) embedding spaces. For sentence embeddings, we average the embeddings of all tokens in a text for RL, and use the embeddings of the last hidden states from the encoder for T5-3B.

4.4.1 Quantitative Results

PLM embeddings can categorize text without task-specific fine-tuning. As RL is not pretrained to generate meaningful sentence embed-



(a) SimCSE-RoBERTa$_{\texttt{large}}$. (b) RoBERTa$_{\texttt{large}}$. (c) Sentence-RoBERTa$_{\texttt{large}}$.

Figure 7: 3D PCA visualization of Amazon dataset in different sentence embedding spaces. (a) Two classes are clearly separated, and we can even find sub-topics by clustering texts in the same class. We can even observe a clear linear semantic structure. (b) Data sub-structures are somewhat kept, but the two sentiment classes are not separated distinctly. (c) The two classes are better distinguished, but the detailed data structures are lost.

Encoder	Size	AG	DB	YH	AM	IM
SimCSE	350M	**86.9**	93.2	**63.9**	93.9	91.0
RL†	350M	86.1	96.0	54.2	93.5	92.3
SRL†	350M	85.8	93.4	55.4	92.9	90.9
T5-3B†	3B	86.7	**96.7**	55.1	**95.3**	**94.5**
BERT(sup.)	110M	94.4	99.4	75.0	97.2	94.5

Table 6: Comparison of different encoders. †: Clusters are initialized using Vanilla Prompting (§4.4.1).

dings, E&M does not work with RL. So we initialize SimPTC using Vanilla Prompting. We do the same for SRL as it provides a better initialization. We share the covariance matrices to offer extra regularization. Surprisingly, as shown in Table 6, the original RL achieves comparable performance to SimCSE and outperforms the more sophisticated sentence encoder SRL on 4 out of 5 datasets.

Larger PLMs tend to have more informative embedding spaces. With a larger model T5-3B, SimPTC gives even better results. Initialized using VP, T5-3B achieves comparable or better performance on 4 out of 5 datasets than the state-of-the-art sentence encoder SimCSE, matching even the supervised BERT performance on IMDb. This indicates that embedding spaces of larger PLMs might have even better clustering properties, which agrees with their stronger zero-/few-shot learning ability.

4.4.2 Qualitative Analysis

To explain the first finding in §4.4.1, we analyze the 3D PCA visualization of the Amazon dataset in three embedding spaces (Figure 7). We observe that: 1) RL preserves the dataset sub-structures, but the two sentiment clusters do not separate very well. 2) SRL pushes semantically close texts together by introducing an extra training objective, which leads to more separable clusters, but the detailed structures of data are lost. 3) The SimCSE embeddings separate the two classes distinctively, and the texts are further clustered together by sub-topics, such as books or products. Very interestingly, a clear linear semantic sub-structure can be observed:

$$\bar{\mathbf{v}}^{book}_{pos} - \bar{\mathbf{v}}^{book}_{neg} \approx \bar{\mathbf{v}}^{prod}_{pos} - \bar{\mathbf{v}}^{prod}_{neg} \approx \bar{\mathbf{v}}_{pos} - \bar{\mathbf{v}}_{neg},$$

where $\bar{\mathbf{v}}^{prod}_{neg}$ is the cluster center vector of all negative product reviews; $\bar{\mathbf{v}}_{pos}$ and $\bar{\mathbf{v}}_{neg}$ are the centers of two sentiment classes. Therefore RL outperforms SRL possibly because it is more descriptive of texts. With a good separability of topics and the ability to capture data sub-structures, SimCSE achieves the best overall zero-shot classification performance.

5 Conclusion

In this work, we show that a simple clustering-based approach, SimPTC, can achieve state-of-the-art zero-shot text classification performance on a wide range of tasks. With extensive experiments, we identify the keys to cluster texts in the PLM embedding spaces and also the limitations of SimPTC. Further analysis of different PLMs shows that PLMs can categorize texts in their embedding spaces without being trained to derive semantically meaningful sentence embeddings, and Larger PLMs tend to have more informative embeddings. We hope our exploration into the embedding spaces of PLMs can provide insights into understanding and developing new methods to elicit the zero-/few-shot learning ability of PLMs.

Limitations

We identify three limitations of SimPTC as well as this work: 1) Due to the nature of clustering and sentence embeddings, SimPTC still suffers at many-class tasks with long documents and tasks with abstract class names (e.g., subjective v.s. objective); 2) Currently how to apply SimPTCto other NLP tasks like NLI is not straightforward. 3) Due to computational resource constraints, our analysis is limited to PLMs with parameters up to 3 Billion. It would be interesting to see if our observations generalize to the largest models like GPT-3 (175B) (Brown et al., 2020) and PaLM (540B) (Chowdhery et al., 2022), which show the strongest zero-/few-shot ability.

Ethics Statement

This work aims to analyze how to use PLM knowledge in their embedding spaces to categorize texts on different topics. Unlike many other deep-learning-based models, SimPTC involves no large neural model pre-training, re-training, or fine-tuning throughout the entire development of the method. Once we get the embeddings of the unlabeled texts, the PLMs are not used anymore. Thus developing and applying our approach requires only minimal computational resources and cause fewer carbon emissions than methods that require dataset-specific fine-tuning or engineering. Besides, we do not anticipate any significant ethical issues introduced by our approach. We use only off-the-shelf PLMs, and the datasets involved are all publicly available topic or sentiment classification datasets. Nevertheless, we urge anyone to evaluate the robustness of the method before using SimPTC in sensitive contexts such as healthcare or legal scenarios.

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*, pages 7747–7763, Online. Association for Computational Linguistics.

Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C Jess Riedel, Emmie Hine, Carolyn Ashurst, Paul Sedille, Alexis Carlier, et al. 2021. Raft: A real-world few-shot text classification benchmark. *arXiv preprint arXiv:2109.14076*.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835.

Yulong Chen, Yang Liu, Li Dong, Shuohang Wang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2022. Adaprompt: Adaptive model training for prompt-based nlp. *arXiv preprint arXiv:2202.04824*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Zewei Chu, Karl Stratos, and Kevin Gimpel. 2021. Unsupervised label refinement improves dataless text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4165–4178, Online. Association for Computational Linguistics.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. *arXiv preprint arXiv:2203.09770*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. 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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 377–384, New York, NY, USA. Association for Computing Machinery.

Vikram Gupta, Haoyue Shi, Kevin Gimpel, and Mrinmaya Sachan. 2022. Deep clustering of text representations for supervision-free probing of syntax. In *Association for the Advancement of Artificial Intelligence*.

Ismail Harrando and Raphaël Troncy. 2021. Explainable zero-shot topic extraction using a common-sense knowledge graph. In *LDK 2021, 3rd Conference on Language, Data and Knowledge*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.

Antonia Kyriakopoulou and Theodore Kalamboukis. 2006. Text classification using clustering. In *Proceedings of the Discovery Challenge Workshop at ECML/PKDD 2006*, pages 28–38.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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

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.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA. Association for Computing Machinery.

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*, page 2121–2132. Association for Computing Machinery, New York, NY, USA.

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*, pages 983–992.

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)*, pages 9006–9017, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Kevin P Murphy. 2007. Conjugate bayesian analysis of the gaussian distribution. *def*, 1(2σ2):16.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

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

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 255–269, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Roman Vershynin. 2012. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

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*, pages 3043–3053, Online. Association for Computational Linguistics.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31.

Hua-Jun Zeng, Xuan-Hui Wang, Zheng Chen, Hongjun Lu, and Wei-Ying Ma. 2003. Cbc: Clustering based text classification requiring minimal labeled data. In *Third IEEE International Conference on Data Mining*, pages 443–450. IEEE.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shangwen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew Arnold, and Bing Xiang. 2021a. Supporting clustering with contrastive learning. *arXiv preprint arXiv:2103.12953*.

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*, pages 2803–2813, Online and Punta Cana, Dominican Republic. 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.

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*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Expanded Class Name Examples for All Datasets

Some examples of the original and extracted class names are shown in Table 11 - 14.

B Templates Used for All Datasets

AG's News:

> The news is about $\langle mask \rangle$.
> The news is related to $\langle mask \rangle$.
> $\langle mask \rangle$ is the topic of the news.
> This week's news is about $\langle mask \rangle$.

DBpedia:

> The object is about $\langle mask \rangle$.
> The object is related to $\langle mask \rangle$.
> $\langle mask \rangle$ is the topic of the object.
> $\langle mask \rangle$ is the subject of the object.

Yahoo:

> The answer is about $\langle mask \rangle$.
> The answer is related to $\langle mask \rangle$.
> $\langle mask \rangle$ is the topic of the answer.
> $\langle mask \rangle$ is involved in the answer.

Amazon:

> A $\langle mask \rangle$ product review.
> The product review is $\langle mask \rangle$.
> The reviewer found the product $\langle mask \rangle$.
> The product is $\langle mask \rangle$.

IMDb:

> A $\langle mask \rangle$ movie review.
> The movie review is $\langle mask \rangle$.
> The reviewer found the movie $\langle mask \rangle$.
> The movie is $\langle mask \rangle$.

C Math foundation of the `SimPTCUpdate` Step

Bayesian approaches inject prior knowledge by introducing prior distribution on model parameters while still allowing the model to fit the data. In this section we first discuss the prior distributions we choose. Then we show how these choices affect the model prediction by analyzing the maximum a posteriori probability (MAP) solution of model parameters.

Prior distributions Following Bishop and Nasrabadi (2006), we choose a Dirichlet distribution as the prior for mixture weights $\boldsymbol{\pi}$, and a Gaussian-Wishart prior for the mean and precisions, i.e., the inverse of covariance $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$:

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\alpha_0) = C(\alpha_0) \prod_k \pi_k^{\alpha_0 - 1}$$

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= p(\boldsymbol{\mu}|\boldsymbol{\Lambda}) p(\boldsymbol{\Lambda}) \\ &= \prod_k \mathcal{N}(\boldsymbol{\mu}_k | \mathbf{m}_0, (\beta_0 \boldsymbol{\Lambda}_k)^{-1}) \cdot \\ &\quad \mathcal{W}(\boldsymbol{\Lambda}_k | \mathbf{W}_0, \nu_0), \end{aligned}$$

where $C(\alpha_0)$ is a normalizing constant, and α_0 can be interpreted as the prior number of observations associated with each mixture. We simply choose $\alpha_0 = \frac{N}{K}$ to favor balanced weights. For the means and covariances, we offer the model maximum freedom to fit the data by choosing a non-informative prior (Murphy, 2007). Specifically, we set:

$$\mathbf{m}_0 = 0,\ \beta_0 \to 0,\ \mathbf{W}_0 = \frac{1}{d} \boldsymbol{\Sigma}_{init}^{-1},\ \nu_0 = d, \quad (2)$$

where $\boldsymbol{\Sigma}_{init}$ is some initial guess of the covariance matrix, which can be set as the empirical covariance of the data. Then we update the model with the standard variational optimization (Bishop and Nasrabadi, 2006) for Bayesian GMM.

MAP solution Here, we show the MAP solution after one update step to give some intuition about how our choice of prior model parameters (2) influences the update of model parameters. As the standard EM update of maximum likelihood methods, the variational update also contains two steps. In the variational E step, we evaluate the responsibilities using the current variational distribution parameters:

$$r_{nk} := \mathbb{E}_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}}[z_{nk}],$$

where z_{nk} is the binary latent variable indicating whether data x_n belongs to cluster k; and in the

Name	Type	# Class	Training Size	Test Size	Max Iter	Covariance Setting
AG's News	Topic	4	120000	7600	50	Full
DBpedia	Topic	14	560000	70000	40	Full
Yahoo	Topic	10	1400000	60000	20	Full
Amazon	Sentiment	2	200000	10000	50	Tied
IMDb	Sentiment	2	25000	25000	150	Tied

Table 7: Statistics of datasets used to compare with state-of-the-art methods in §4.1 and extra model settings. SimPTC stops when the model prediction stops changing, or the maximum number of iteration is achieved. Full: each Gaussian mixture has its own covariance $\mathbf{\Sigma}_k$. Tied: all Gaussians share the same covariance $\mathbf{\Sigma}$.

variational M step, we update the variational distribution parameters. For simplicity, we introduce the following statistics:

$$N_k = \sum_n r_{nk}$$
$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_n r_{nk} \mathbf{x}_n$$
$$\mathbf{S}_k = \frac{1}{N_k} \sum_n r_{nk} (\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^\top.$$

Then the MAP solution of $\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Sigma}$ given the responsibilities r_{nk}'s after a variational M steps is given by

$$\begin{aligned} \pi_k^* &= \frac{\alpha_0 - 1 + N_k}{K(\alpha_0 - 1) + N} \\ \boldsymbol{\mu}_k^* &= \bar{\mathbf{x}}_k \\ \mathbf{\Sigma}_k^* &= \frac{d\mathbf{\Sigma}_{init} + N_k \mathbf{S}_k}{N_k - 1}, \end{aligned} \tag{3}$$

where d is the number of feature dimensions and K is the number of classes. We can see that by choosing non-informative prior (2) of $(\boldsymbol{\mu}, \mathbf{\Sigma})$, we allow the model to fit the data with maximum freedom. By choosing a large α_0, we can push the mixing weights towards uniform but still allow the model to fit the data.

D Datasets Statistics and Model Settings

The statistics of the five datasets used in §4.1 and max iteration numbers can be found in Table 7. For Amazon, we use the same test set sampled by Hu et al. (2021) and randomly sample 200,000 texts from the original training set for the unlabeled training data. Since SimCSE only handles texts with a maximum length of 512, we crop texts with lengths exceeding 512. We choose the maximum number of iterations empirically according to the size of the unlabeled data which is equal to the training set size plus the test set size. For topic datasets, each Gaussian has its individual covariance matrix. For sentiment datasets, all Gaussians share the same covariance matrix to provide extra regularization as the data is relatively sparse. The effect of sharing the covariance matrix is discussed in Appendix G.

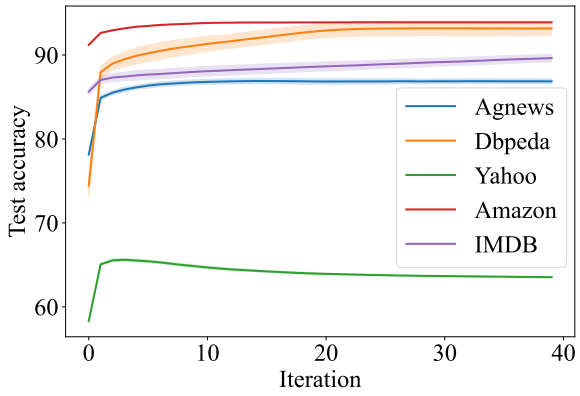


Figure 8: The performance v.s. update iteration plot of SimPTC on all five datasets. The solid line shows the average accuracy at each iteration, whereas the blurred area indicates the standard deviation of using different templates. **SimPTC converges to a good-quality prediction as the clustering process converges.**

E Convergence Analysis

Although SimPTC is guaranteed to converge, it is unclear whether it will converge to a good solution when the algorithm stops. Therefore we study how the model performance varies as the updating process proceeds. We plot the test accuracy of intermediate update steps on all datasets in Figure 8, where the standard deviations caused by using different templates are illustrated with blurred areas. We observe that the performance gradually improves and converge in all dataset except Yahoo, where SimPTC still converges to a result much better than the initialization. Also, as shown in the blurred areas in Figure 8, the update step is sta-

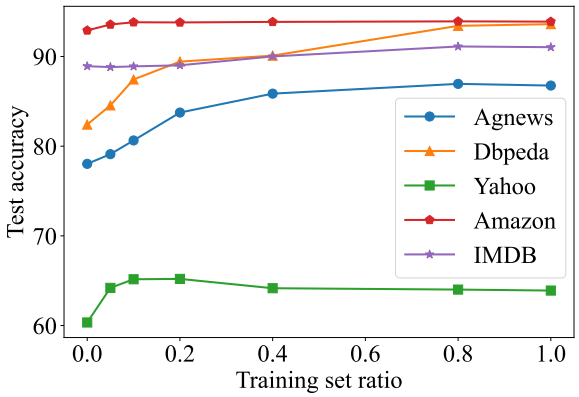


Figure 9: The performance v.s. unlabeled dataset size plot of SimPTC on all five datasets. More unlabeled data, in general, tends to improve the prediction of SimPTC.

ble when different templates are used. Moreover, SimPTC almost converges on all five datasets under our setting of the maximum number of iterations.

F Effect of Unlabeled Dataset Size

In the standard setting, we use both the train and test set for fitting the Bayesian GMM. To study the effect of the unbalanced dataset size, we keep the unlabeled test data and use the training data with ratios varying from 0 to 1. As illustrated in Figure 9, on almost all datasets, more unlabeled data brings more improvement.

One possible explanation is: to model the shape of all clusters with a certain error threshold, one needs samples of a number at least linear to the number of dimensions (Vershynin, 2012) and linear to the number of classes. Therefore a large unlabeled dataset helps the model to fit data with many classes in a high-dimensional space better (for RoBERTa$_{\texttt{large}}$, the number is 1024). By sharing the covariance matrix (Amazon and IMDb), we reduce the number of model parameters. Thus SimPTC works better than fitting individual covariance for each cluster (Agnews, Dbpedia, and Yahoo) when the data is sparse. Since for many tasks collecting unlabeled data is considered to be much easier than collecting annotated data, we can improve the performance of SimPTC in real-world applications at a low cost.

G Effect of Sharing Covariance Matrix

We explore two covariance settings in SimPTC. *Full*: each Gaussian mixture has its own covariance $\mathbf{\Sigma}_k$, and *tied*: all Gaussians share the same covariance $\mathbf{\Sigma}$. Note that the sharing the covariance matrices

Setting	Topic			Sentiment	
	AG	DB	YH	AM	IM
E&M	78.2	74.4	58.3	91.2	85.6
Full	**86.9**↑	**93.2**↑	**63.9**↑	92.4↑	86.2↑
Tied	86.5↑	90.8↑	56.9↓	**93.9**↑	**91.0**↑

Table 8: Average test accuracy of all templates with different covariance settings. Tied: all Gaussians share the same covariance matrix. Full: every Gaussian has its own covariance matrix.

(the *full* and *tied* setting) is a standard hyperparameter of GMM. The *full* setting is more flexible, and as Table 8 shows it improves the initial E&M predictions on all datasets. By sharing the covariance matrices (the *tied* setting) we 1) reduce model parameters to provide extra regularization and 2) add stronger assumptions on the cluster shapes. Therefore it is useful when

- the data is relatively sparse (e.g., IMDb in Table 8 and TC14 datasets in §4.3),
- the embedding space of PLM is less structured (T5 and RoBERTa$_{\texttt{large}}$ embeddings (§4.4)),
- the texts of different classes describe similar objects (e.g., sentiment tasks).

Otherwise, we recommend allowing clusters to have different covariances.

H Implicit Balanced Assumption of KPT

Hu et al. (2021) proposed a data-dependent Contextualized Calibration (CC). They motivate CC by observing that some label words are less likely to be predicted than others, regardless of the label of input sentences. To solve the problem, CC works in the following steps: First, to estimate a contextualized prior distribution of label words using some sampled unlabeled data:

$$\begin{aligned} P_{\mathcal{D}}(v) &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} P_{\mathcal{M}}([MASK]=v|\mathbf{x}) \\ &\approx \frac{1}{|C|}\sum_{\mathbf{x}\in C} P_{\mathcal{M}}([MASK]=v|\mathbf{x}), \end{aligned} \tag{4}$$

where v stands for a particular label word, $\mathcal{D}$ is the data distribution, $P_{\mathcal{M}}$ is the model prediction, C is a sampled subset of the dataset. Then they use the contextualized prior of label words to calibrate the predicted distribution:

$$\tilde{P}_{\mathcal{M}}([MASK]=v|\mathbf{x}) \propto \frac{P_{\mathcal{M}}([MASK]=v|\mathbf{x})}{P_{\mathcal{D}}(v)}. \tag{5}$$

Name	Type	Class name examples	Template for prompting
20 News	Topic	comp.graphics; sci.space	[Category : $\langle mask \rangle$] $\langle text \rangle$
NYT-Topic	Topic	business; politics; sports	[Category : $\langle mask \rangle$] $\langle text \rangle$
NYT-Location	location	united_states; iraq; japan	[Category : $\langle mask \rangle$] $\langle text \rangle$
BBC News	Topic	sport; business; entertainment	[Category : $\langle mask \rangle$] $\langle text \rangle$
Emotion	Emotion	sad; joy; anger	[Category : $\langle mask \rangle$] $\langle text \rangle$
Banking77	Intent	activate_my_card; age_limit	[Category : $\langle mask \rangle$] $\langle text \rangle$
TREC	Question	abbr.; entity; description	[Category : $\langle mask \rangle$] $\langle text \rangle$
Biomedical	Paper title	aging; chemistry; erythrocytes	[Category : $\langle mask \rangle$] $\langle text \rangle$
StackOverflow	Question	svn; oracle; bash	[Category : $\langle mask \rangle$] $\langle text \rangle$
Yelp	Sentiment	positive; negative	It is $\langle mask \rangle$. $\langle text \rangle$
SST-2	Sentiment	positive; negative	It is $\langle mask \rangle$. $\langle text \rangle$
SST-5	Sentiment	very positive; positive; negative	It is $\langle mask \rangle$. $\langle text \rangle$
MPQA	Opinion polarity	positive; negative	It is $\langle mask \rangle$. $\langle text \rangle$
Subj	Subjectivity	subjective; objective	It is $\langle mask \rangle$. $\langle text \rangle$

Table 9: Extra information about TC14 datasets. Template for prompting is the template we used to perform prompt-based zero-shot learning, i.e., `Vanilla Prompting`. We use the same template for all sentiment tasks and another for all other datasets.

The final probability is normalized to 1.

The contextualized prior can be interpreted as a marginal distribution. Consider we have one label word for each class. The contextualized prior measures the portion of each class in the dataset based on the model's predictions. Then `CC` penalizes the probability of predicting one class if the model thinks it assigns too many samples to this class ($P_{\mathcal{D}}(v)$ is large). Intuitively this is to force the model to assign equal numbers of samples to each class, which is to force a uniform marginal distribution. The underlying implicit assumption is that the dataset is balanced. Although `CC` improves the zero-shot performance of `KPT`, we argue that this is because the evaluation datasets happen to be balanced, and `CC` becomes problematic when the dataset is unbalanced (see C2 in §4.1).

I TC14 Datasets

To study the applications and limitations of `SimPTC`, we collect the following 14 datasets with diverse topics, text lengths, and class numbers. Specifically, we did a literature search in zero-shot text classification and collected datasets that best fit our text classification setting with label names that have class-info. We first introduce the details of the TC14 datasets (§I.1). Then we discuss the implementation details in §I.2. We show the full results in §I.3 and provide extra analysis in §I.4.

I.1 Dataset Information

The datasets we used are:

- **20 News** (Lang, 1995) is a news classification dataset. It has a relatively long average text length and many classes.
- **NYT-Topic** (Meng et al., 2020a) is a long document topic classification dataset that is very unbalanced.
- **NYT-Location** (Meng et al., 2020a) uses the same corpus as NYT-Topic but categorizes the texts according to locations. The dataset is very unbalanced.
- **BBC News** (Greene and Cunningham, 2006) is a news dataset containing 2225 articles.
- **Yelp** (Zhang et al., 2015) is a review sentiment dataset.
- **Emotion** (Saravia et al., 2018) is a dataset of English Twitter messages with six basic emotions, and the dataset is very unbalanced.
- **Banking77** (Casanueva et al., 2020) is a dataset composed of online banking queries annotated with their corresponding intents. It has a very fine-grained set of intents in the banking domain. 13,083 customer service queries are categorized into 77 intents.
- **SST-2** (Socher et al., 2013) is a sentence sentiment classification dataset.
- **SST-5** (Socher et al., 2013) is a fine-grained sentiment classification dataset. Texts are classified into five sentiment classes: very negative, nega-

Method	20News	NYT-T	NYT-L	BBC	Yelp	Emotion	Banking77
VP	41.0/36.6	**72.1**/55.5	66.3/62.0	75.8/73.8	80.6/80.0	21.7/19.3	21.2/16.5
Encode&Match	42.9/42.0	59.6/53.9	65.9/66.4	80.6/80.4	93.3/93.3	**52.3**/46.2	57.0/55.6
SimPTC	**51.2/53.1**	66.0/**63.5**	**72.1/77.7**	**89.5/89.7**	**94.3/94.3**	51.0/**46.6**	**66.6/66.7**
Zero-shot SOTA	78.6/77.8[a]	79.0/68.6[a]	91.8/92.0[a]	84.0 (acc)[b]	90.0/90.0[a]	-/-	-/33.2[c]
	SST-2	**SST-5**	**MPQA**	**Subj**	**TREC**	**Biomed.**	**StackOF**
VP	73.7/72.0	32.4/28.4	49.0/48.6	**56.2**/47.8	**37.7**/28.1	25.0/22.4	26.6/21.0
Encode&Match	82.0/81.9	42.7/38.7	83.8/82.4	51.7/47.2	35.4/28.9	26.8/24.7	49.0/49.4
SimPTC	**86.8/86.8**	**46.2/42.2**	**84.8/83.4**	53.9/**52.0**	37.3/**30.8**	**38.4/40.6**	**74.2/77.9**
Zero-shot SOTA	83.6 (acc)[d]	35.0 (acc)[d]	67.6 (acc)[d]	51.4 (acc)[d]	32.0 (acc)[d]	46.2 (acc)[e]	75.5 (acc)[e]

Table 10: Zero-shot micro-/macro-F1 scores on other datasets. VP: vanilla prompting (§4). We collect publicly available zero-shot state-of-the-art (SOTA) method performance as a reference. a: X-Class, (Wang et al., 2021) a SOTA keyword-based method. b: (Harrando and Troncy, 2021). c: Crowdsourced human performance from Alex et al. (2021) (they used a selected portion of Banking77). d: zero-shot prompt-based zero-shot learning provided by Gao et al. (2021a). e: SCCL, a contrastive-learning-based unsupervised text clustering method by Zhang et al. (2021a). SCCL forces on clustering texts of different topics. When calculating accuracy, the labels of clusters are determined by solving a min-cost perfect matching problem based on the predicting accuracy.

tive, neutral, positive, and very positive.

- **MPQA** (Wiebe et al., 2005) is an opinion polarity analysis dataset.
- **Subj** (Pang and Lee, 2004) is a subjectivity analysis dataset.
- **TREC** (Voorhees and Tice, 2000) is an unbalanced question classification dataset.
- **Biomedical** (Xu et al., 2017) is a paper title classification dataset, where 20,000 titles are categorized into 20 groups.
- **StackOverflow** (Xu et al., 2017) is a dataset containing 20,000 questions with 20 classes.

Since we are evaluating zero-shot methods, we report scores on the full datasets (dataset sizes are shown in Table 5).

I.2 Additional Implementation Details

We compare with Vanilla Prompting rather than KPT because KPT has an improper balanced dataset assumption (§4.1 C3), and KPT cannot handle class names containing multiple words.

For the 20 News dataset, we use class names from Mekala and Shang (2020) as the original class names are not complete English. We implement Vanilla Prompting using OpenPrompt (Ding et al., 2021). When a class name contains multiple words, we use the average probability of predicting each word as implemented in OpenPrompt. BBC News contains only 2225 texts and is too small to fit a 1024-by-1024 covariance matrix even if we share the covariance matrices of clusters. Banking77 has too many classes compared with the dataset size, and as a result, Encode&Match assing zero samples to some classes. To fix these two problems, we perform a PCA to reduce the feature dimension such that the reconstruction error is 3% before Encode&Match.

I.3 Full Results

We report the micro-macro F1 scores on TC14 in Table 10. For comparison, we also collect publicly available state-of-the-art results on these datasets. Some papers only report the accuracy of their models, and we report these numbers instead.

I.4 Additional Analysis

As discussed in §4.3, both prompting and E&M suffer on the Subj dataset where the class names are abstract concepts (subjective v.s. objective). As a result, SimPTC also does not go very far from random guessing (50%). However, despite E&M failing to link the texts correctly with the abstract class names, the texts themselves are well-separated in the embedding space (Figure 6). This suggests that texts with abstract classes can also be clustered together in the PLM embedding spaces. A 10-shot setting (averaged over 5 seeds) improves SimPTC from 52.0 to 89.2 on Subj, outperforming GPT-3 175B in-context learning (76.4).

In terms of limitations, another important observation is that: on long document classification tasks (20 News, NYT-Topic, NYT-Location), both SimPTC and Vanilla Prompting underperform the state-of-the-art keyword-based method X-Class (Wang et al., 2021), showing an information loss when PLMs encodes long documents into the em-

bedding spaces. This indicates that in terms of extracting information from long documents, self-training keyword-based approaches still perform better than zero-shot our clustering-based approach and prompting methods.

Class Name	Expanded Class Names
politics	alt rightist, social fascism, psychopolitical, leader of opposition, junior minister, whipped vote, political, regressive leftism, policy making, dollar democracy, ...
sports	professional baseball, game set match, banana ball, empty bench, first touch, football, sportsman, visiting team, athletic, exhibition game, super cup, ...
business	account name, commerciality, making money, sprinkler strategy, web company, consumer good, business economics, maintained markup, commercial enterprise, ...
technology	cryoengineering, aeronautical engineering, geotechnology, cwm silicon, nuclearism, digital technology, cryotechnology, xenotechnology, applied science, deepfake, ...

Table 11: Original class names and expanded class names of AG's News.

Class Name	Expanded Class Names
company	hook stock, private corporation, large company, big company, business organization, furniture company, companies, sprinkler strategy, corp, livery company, ...
school	elementary schooler, undergraduates, university student, dual school, antiuniversity, schoolless, overschooled, secondary modern, science room, state school, ...
artist	arte povera, ernstian, art show, da vincian, polystylist, gallery opening, pricasso, artworks, artistdom, superrealist, artists, clean brushes, post impressionist ...
athlete	olga korbut, athleticism, pull muscle, walking sports event, pancratical,nongymnast, sportswomen, athletic contest, weightlifter, winter olympics competition, ...
politics	alt rightist, social fascism, psychopolitical, leader of opposition, junior minister, whipped vote, political, regressive leftism, policy making, dollar democracy, ...
transportation	antirail, air freight logistics, delivered ex ship, road rail, transmodal, water bailage, transportive, cargon, vecturist, multiride, transfer to hospital, ...
building	tower block, nonbuilding, inbond, interior door, interiorscaper, split level, electrical wiring, seismic retrofit, house raising, sevenplex, office complex, ...
river	mountainlike, talav, mountainside, mount sharp, river, lake albert nyanza, subapennine, khabur, transmountain, longs peak, riverling, land form, monticulus, ...
village	koprivnica, khutor, intown, b road, mini mall, oppidan, cybervillage, gaothan, lawley, shillingstone, shakespeare play, claygate, goosnargh, hamlets, northcott, ...
animal	gambian pouched rat, cattle beast, wild game, cymothoa exigua, farm animal, bestiarian, stylophora, brazilian wandering spider, western black rhinoceros, ...
plant	anthoxanthum odoratum, harpulla, calochortus amabilis, brazilian pepper tree, tree roots, cuphea, lespedeza bicolor, phoenix tree, akeake, rauli beech, nontree,...
album	studio album, lyrics, space cakes, guitar drums, song, chiodos, american life, dance pop, keys of kingdom, record deal, rock opera, songsheet, songcraft, ...
film	star actor, filmically, company men, moving pictures, stfilm, getting acquainted, sound film, photographic film, collage film, cinematology, filmize, ...
book	megabook, pilgrim's progress, neophiliac, forebook, young adult fiction, clipsheet, novels, novel, book, novelle, reading material, booklessness, e novel, ...

Table 12: Original class names and expanded class names of DBpedia.

Class Name	Expanded Class Names
society, culture	crowd elevator, cybersociety, macroculture, intersocietal, islandness, desocialize, cultureshed, overculture, preculture, ghost skin, antisociety, ...
science, mathematics	inequality sign,ur science, odd function, common antilog, hydroscience, known quantity, find out truth, science, commutative law, aetherometry, ...
health	being well, dietetist, hale and hearty, healthcare delivery, healthful, health, country doctor, geomedical ,health centre, nutritionwise, patient contact,...
education, reference	postsecondary school, uneducation, special educator, secondary education, cross index, tertiary education, forward reference, exophora,...
computers, internet	allows null sessions, dynamic ip address, friendly url, data processor, laptops, deadlink, web diving, dictionary attacker, nt account system, ...
sports	professional baseball, game set match, banana ball, empty bench, football, sportsman, visiting team, athletic, exhibition game, super cup, ...
business, finance	adhocratic, net operating loss, business organization, capital structure, systematic risk, manufacturers rep, web company, garmento, ...
entertainment, music	bigophonic, good fun, entertaintment, natabhairavi, eating popcorn, allegro non troppo, semihemidemisemiquaver, musicaholic, ...
family, relationships	mother father, enicocephalid, profamily, close friendship, salpidae, visual proximity, relations, lac scale, sexual relationship, ...
politics, government	governmentalise, ruling party, westminster system, antiindependence, leader of opposition, cryptarchy, macropolitical, antipopulist,...

Table 13: Original class names and expanded class names on AG's News.

Class Name	Expanded Class Names
bad	overawful, crappy, uglysome, not good, suck balls, do badder, blow chunks, shitly, godawful, sucktastic, worsts, horridsome, fucky, god awful, terrible, ...
good	correct answer, have good day, better job, clean apartment, double plus good, nice, talk with friends, goodish, supernice, like million bucks, healthy environment, ...

Table 14: Original class names and expanded class names on AG's News.