# Diverse Distributions of Self-Supervised Tasks for Meta-Learning in NLP

**Trapit Bansal**[♠*†] and **Karthick Gunasekaran**[♠] and **Tong Wang**[♦] and **Tsendsuren Munkhdalai**[♣†] and **Andrew McCallum**[♠]
[♠] University of Massachusetts Amherst
[♦] Microsoft Research, Montréal, Canada
[♣] Google Research

## Abstract

Meta-learning considers the problem of learning an efficient learning process that can leverage its past experience to accurately solve new tasks. However, the efficacy of meta-learning crucially depends on the distribution of tasks available for training, and this is often assumed to be known a priori or constructed from limited supervised datasets. In this work, we aim to provide task distributions for meta-learning by considering self-supervised tasks automatically proposed from unlabeled text, to enable large-scale meta-learning in NLP. We design multiple distributions of self-supervised tasks by considering important aspects of task diversity, difficulty, type, domain, and curriculum, and investigate how they affect meta-learning performance. Our analysis shows that all these factors meaningfully alter the task distribution, some inducing significant improvements in downstream few-shot accuracy of the meta-learned models. Empirically, results on 20 downstream tasks show significant improvements in few-shot learning – adding up to $+4.2\%$ absolute accuracy (on average) to the previous unsupervised meta-learning method, and perform comparably to supervised methods on the FewRel 2.0 benchmark.

## 1 Introduction

Humans show a remarkable capability to accurately solve a wide range of problems efficiently – utilizing a limited amount of computation and experience. Deep learning models, by stark contrast, can be trained to be highly accurate on a narrow task while being highly inefficient in terms of the amount of compute and data required to reach that accuracy. Within natural language processing (NLP), recent breakthroughs in unsupervised pre-training have enabled reusable models that can be applied to many NLP tasks, however, learning of new tasks is still inefficient (Yogatama et al., 2019; Bansal et al., 2020a; Linzen, 2020). Meta-learning (Schmidhuber, 1987; Bengio et al., 1992; Thrun and Pratt, 2012) treats the learning process itself as a learning problem from data, with the goal of learning systems that can generalize to new tasks efficiently. This has the potential to produce few-shot learners that can accurately solve a wide range of new tasks. However, meta-learning requires a distribution over tasks with relevant labeled data that can be difficult to obtain, severely limiting the practical utility of meta-learning methods.

In the supervised setting, in particular, meta-learning task distribution is often defined by sub-sampling from the classes in a classification problem over a fixed dataset (Vinyals et al., 2016). This not only limits the applicability of meta-learning to the underlying classification problem, but also requires a diverse set of supervised datasets with a large number of classes to enable learning. Self-supervised meta-learning, on the other hand, seeks to propose tasks from unlabelled data (Hsu et al., 2019; Bansal et al., 2020b), and has great potential to enable numerous important applications (Hospedales et al., 2020) such as neural architecture search, continual learning, hyper-parameter optimization, learning in low-resource settings, etc. Existing work in meta-learning for NLP, however, defaults to task distributions that tend to be overly simplistic, e.g. using existing supervised datasets (Han et al., 2018; Dou et al., 2019; Bansal et al., 2020a) or unsupervised cloze-style tasks with uniform selection of words from the vocabulary (Bansal et al., 2020b). Given the lack of exploration on this critical component, we propose to devise and evaluate various task distributions in the context of unsupervised meta-learning for NLP.

Specifically, we explore a diverse set of approaches to create task distributions that are inductive to better meta-training efficacy. We provide empirical evidence that existing definitions of

* Correspondence: tbansal@cs.umass.edu.
† Part of the work was done at Microsoft Research.



*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5812–5824
November 7–11, 2021. 

task distributions are prone to producing tasks that might not be challenging enough for the underlying model to learn useful representations, which in turn translates into poor downstream task performance. We therefore propose several new approaches that instead consider important features of the task distribution including task diversity, difficulty, resemblance to the downstream tasks, and the curriculum or the order in which tasks are presented during training. When evaluated on a suite of 20 NLP classification tasks, our best unsupervised meta-learning method leads to an absolute increase of up to $+4.2\%$ in average few-shot accuracy over *unsupervised* baseline results; and it even outperforms *supervised* meta-learning methods on FewRel 2.0 benchmark (Gao et al., 2019) on 5-shot evaluation.

The paper is organized as follows. We start by providing some relevant background (2) on meta-learning and the unsupervised task generation approach in SMLMT. Next, we introduce (3) new approaches to improve the task distribution. We then analyze (4.2) the different unsupervised distributions and how they relate to each other. Finally, we evaluate (4.3, 4.4) the different unsupervised methods on a wide range of NLP tasks including sentiment classification, entity typing, text classification, sentence-pair classification and relation classification.

## 2 Background

### 2.1 Meta-Learning

In this work, we focus on Model Agnostic Meta-Learning (MAML) (Finn et al., 2017), which is an optimization-based meta-learning method. To efficiently adapt to a task training data, MAML jointly optimizes the initial point of a neural network model and a gradient-descent based optimizer. This is framed as a bi-level optimization consisting of an inner loop for task-specific learning and outer loop for fast adaptation across tasks:

$$\text{Inner: } \theta_i' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_i(\mathcal{D}^{tr}, \theta)$$

$$\text{Outer: } \Theta \leftarrow \Theta - \beta \, \nabla_\Theta \mathbb{E}_{T_i \sim \mathcal{P}(\mathcal{T})} \left[ L_i(\mathcal{D}^{val}, \theta_i') \right]$$

where $\theta$ are the parameters of the model, $\alpha$ is the (learnable) inner loop learning rate, $\Theta := \{\theta, \alpha\}$, $\mathcal{L}_i$ is the loss for task $T_i$, $(\mathcal{D}^{tr}, \mathcal{D}^{val}) \sim T_i$ are support and validation data for the task $T_i$, and $\beta$ is the outer loop learning rate which is a hyper-parameter. Typically, multiple steps of gradient descent are performed in the inner loop. Training such methods proceeds in an episodic framework (Vinyals et al., 2016), where in each episode a mini-batch of tasks are sampled along with their support and validation sets, and the model parameters are optimized as above.

### 2.2 Task Distribution for Meta-Learning

Meta-learning assumes access to a distribution $\mathcal{P}(\mathcal{T})$ over tasks. The goal is to utilize tasks $T_i \sim \mathcal{P}(\mathcal{T})$ sampled from this distribution to train a learning procedure that generalizes to unseen tasks $T' \sim \mathcal{P}(\mathcal{T})$ from the distribution. Supervised meta-learning often utilizes a fixed task dataset to create $\mathcal{P}(\mathcal{T})$ by sub-sampling from all class labels (Vinyals et al., 2016). Bansal et al. (2020b) sought to provide an unsupervised approach that proposes tasks from unlabelled data. The resulting Subset Masked Language Modeling Tasks (SMLMT) approach proposes self-supervised tasks to enable meta-learning and improves few-shot learning across a diverse set of classification tasks.

Sampling an $N$-way task from SMLMT requires first sampling a size-$N$ subset of the vocabulary, which are subsequently mapped to consecutive integer ids and serve as labels for the task. Then to sample examples for each label, sentences containing that word are sampled and the occurrences of the word are masked out. Note that a task in SMLMT is a sentence classification task where each input sentence consists of exactly one word type that is masked throughout the sentence and the label for the sentence is the underlying word type that was masked. This enables sampling combinatorially many classification tasks for meta-learning.

## 3 Exploring Unsupervised Task Distribution for Meta-Learning

Sampling tasks in SMLMT depends on sampling of words, which serve as labels, and sampling of sentences containing that word. The original formulation used uniform sampling for both steps. This can lead to several limitations on the quality of the resulting task distribution including task diversity and difficulty. The single-sentence classification tasks also lack cross-sentence reasoning capacities, leading to a severe train-test mismatch for downstream tasks involving sentence pairs. To remedy these problems, we consider alternative distributions that are inductive to more diverse and challenging tasks for meta-training. We also describe an automatic curriculum over tasks that seeks to

continuously find challenging tasks for the model during training.

### 3.1 Sampling labels in SMLMT

**Frequency-based sampling:** Word distribution in natural language is characterized by an exponential distribution with a long tail of rare words (Baayen, 2002). Uniform sampling of words in SMLMT puts a disproportionately high weight on the long tail, leading to inefficient use of the training corpora since the low frequency words occur in only a small proportion of the sentences. On the other hand, simple frequency-based sampling can be highly skewed towards a handful of high frequency words. We thus propose to simply sample words in proportion to their log-frequency instead.

**Cluster-based sampling:** Given two words randomly sampled from a large vocabulary, it is likely to be rather trivial to distinguish their corresponding contexts. This can lead to overly simple tasks in the SMLMT task distribution. To avoid this problem, we consider clustering words based on pre-trained word embeddings and grouping words into semantically-related clusters. Diverse and difficult instances of tasks in SMLMT can then be sampled by selecting all words in a task from either (1) the same cluster (*intra-cluster* sampling), or (2) different clusters (*inter-cluster* sampling). Words co-occurring in the same cluster are semantically or topically related and hence occur in similar contexts, leading to harder to classify sentences as we see in our analysis (Sec 4.2). Moreover, choosing different clusters to sample words across tasks provides a natural diversity over topics in the training tasks. On the other hand, picking words from different clusters (*inter-cluster* sampling) can still lead to tasks where the sentences are easy to classify due to easily distinguishable contexts.

Specifically, clustering of pre-trained word embeddings using $k$-means has been proven effective in generating topical clusters rivaling topic models (Sia et al., 2020). We use the FastText (Joulin et al., 2017) embeddings as word representations. We choose FastText as it is fast, incorporates subword information, can generate embeddings for out-of-vocabulary words, and has been found to yield topical clusters (Sia et al., 2020).

Since cluster sizes can be imbalanced, we pick clusters proportional to the number of words in the cluster. Thus, assuming $\{C_1, \ldots, C_m\}$ to be the $m$ clusters of the word vocabulary, we replace the uniform sampling over words in SMLMT as:

$$
\begin{aligned}
p_i &= \frac{|C_i|}{\sum_{t=1}^{m} |C_t|} \\
i &\sim \text{Cat}(p_1, \ldots, p_m) \\
w|i &\sim \text{Uniform}(\{w|w \in C_i\})
\end{aligned}
$$

where $\text{Cat}(p_1, \ldots, p_m)$ is a categorical distribution over $m$ categories with probabilities $\{p_1, \ldots, p_m\}$.

### 3.2 Dynamic Curriculum over Self-Supervised Tasks

The methods discussed so far use a static task distribution for learning with tasks sampled i.i.d from this distribution for training. Curriculum learning (Bengio et al., 2009; Graves et al., 2017) instead posits that choosing the order in which instances are presented, with gradually increasing complexity, can enable faster learning and better generalization. We explore whether a curriculum in task sampling is beneficial for meta-learning by proposing a method to sample increasingly difficult tasks during training. To enable this we need a method to propose difficult tasks based on the current state of the model during training.

Since words act as labels in SMLMT, words that are closer in the representational space of the neural model will be more difficult to distinguish, leading to more difficult tasks. On the other hand, nearest-neighbors can be too difficult to induce effective learning for a model. This is related to findings in negative sampling in metric learning literature (Schroff et al., 2015; Suh et al., 2019) where using "too hard" negatives typically hurts performance.

To alleviate this problem, we cluster representations computed from the model and uniformly sample words within the same cluster to create difficult but not impossible tasks (similar to the "static" clustering approach). Secondly, we adopt an easy-to-hard curriculum by controlling the ratio between the harder tasks from the dynamic distribution $\mathcal{D}_t$ and the easier ones from the static distribution $\mathcal{S}$, consisting of tasks sampled i.i.d from uniform random word sampling or fixed word-clustering. At step $t$, let $\lambda_t$ be the probability of sampling a task from $\mathcal{D}_t$ and $1 - \lambda_t$ from $\mathcal{S}$. Then the dynamic curriculum is defined by sampling tasks from the following mixture distribution with $\lambda_t$ linearly annealed over the training epochs from 0 to 1:

$$
T \sim \lambda_t \mathcal{D}_t + (1 - \lambda_t)\mathcal{S}
$$

To construct $\mathcal{D}_t$, we consider the following word (i.e. label) representation for clustering, obtained by the average representation under the model of the masked sentences corresponding to a word:

$$\hat{w}_i^{(t)} = \mathbb{E}_{x \sim S(w_i)} \left[ f_{\theta_t}(x) \right]$$

where $S(w_i)$ is the set of all sentences containing the word $w_i$ with the word $w_i$ masked out (as defined in SMLMT), $f_{\theta_t}(.)$ is the representation from the neural model for instance $x$ that is fed into the softmax classification layer, and $\theta_t$ are the model parameters at step $t$.

To make the computation of $\hat{w}_i^{(t)}$ tractable, we first approximate the quantity by the expectation over a subset of $S(w_i)$. Moreover, since computing the representations $\{\hat{w}_i^{(t)}\}$ for all vocabulary words and clustering at every step $t$ of the training will be computationally infeasible, we consider doing this after $m$ steps of meta-training. This also allows the model to train on the current distribution for sometime before abruptly changing the distribution. Finally, while the model is being updated between time step $t$ and $t+m$, we use the model snapshot at $t$ to create the word clusters asynchronously for the model at $t+m$, which allows the task generation to run in parallel to the model training.

### 3.3 Task proposal using sentence clustering

SMLMT uses a data-augmentation strategy to automatically assign labels to unlabelled sentences by consistently masking out the *same* word type in a set of sentences. The masked out word then serves as the label for the sentences. This cloze-style approach to creating tasks was inspired by the success of masked language modeling (Devlin et al., 2018) in learning useful representations. While this leads to significant improvements in sentence-level classification on a range of real downstream tasks (Bansal et al., 2020b), it is unclear whether a word masking approach is the most efficient to learning useful sentence representations. To probe this question further, we explore an alternative to SMLMT that directly assigns semantic labels to sentences without any augmentation.

Specifically, we consider pre-trained sentence representations for proposing tasks, which have been proven useful for improving semi-supervised learning (Du et al., 2020). We use a pre-trained sentence embedding model (Du et al., 2020; Wenzek et al., 2020) to embed all sentences in a corpus and cluster them. To propose an $N$-way task, we first randomly sample $N$ cluster-ids and remap them to random consecutive integers $\{1, \ldots, N\}$. Then examples for each label are sampled from the corresponding cluster, creating a classification task for classifying the sentences into their underlying cluster labels. Note that the step of remapping the cluster-ids ensures that the model cannot memorize the sentence to cluster mapping, which would lead to meta over-fitting (Hsu et al., 2019).

### 3.4 Contrastive learning over sentence pairs

SMLMT proposes sentence-level tasks and thus lacks cross-sentence reasoning. This is confirmed by the poor downstream few-shot performance of models trained on SMLMT (see Sec. 4.3). Since models trained on SMLMT have never seen pairs of sentences as input, it leads to a train-test mismatch for sentence-pair classification tasks. To remedy this, we introduce a simple but effective contrastive learning task over sentence-pairs that bridges this gap. Contrastive learning has been used to learn effective sentence representations (Logeswaran and Lee, 2018). Next sentence prediction, a sentence-pair task, was used in the training of BERT (Devlin et al., 2018) which was later found to be not effective (Liu et al., 2019b). BERT considered segments instead of full sentences, however the downstream tasks often require reasoning over complete sentences. Thus, we consider classifying whether two sentences come from the same document as opposed to different documents, as a sentence-pair task to enable cross-sentence reasoning. This simple objective was found to be quite effective in our experiments. Note that during meta-training, this can be treated as an additional task in the task distribution. Since the SMLMT task distribution consists of an exponential number of tasks, we sample the sentence-pair task in an episode with a fixed probability $\alpha$, which is hyper-parameter.

## 4 Experiments

We evaluate various self-supervised task distributions for their utility in meta-learning for few-shot classification. We first describe the experimental setting, then we perform evaluations to understand how the different self-supervised tasks relate to each other, and finally show performance on a large set of 20 real classification datasets. These datasets cover a wide range of tasks: sentiment classification, entity typing, text classification, sentence pair classification and relation classification.

Our proposed approach shows significant improvements over previous few-shot classification results (Bansal et al., 2020b; Gao et al., 2019).

### 4.1 Experimental Setup

We consider the challenging few-shot setting where models are trained on unlabelled corpora and then evaluated on target tasks with only $k$ examples per label ($k \leq 32$) to allow fine-tuning of the models on the target task. Since our focus is on unsupervised meta-learning, we closely follow the experimental setup of Bansal et al. (2020b).

**Meta-learning Model** We use the same model as in Bansal et al. (2020b) for our results to be comparable[1]. The model is a BERT transformer encoder coupled with a parameter generator, a 2-layer MLP, that generates the initial point for classification layer for a task conditioned on the support examples. The model is meta-trained using the MAML algorithm (Finn et al., 2017), with learned per-layer learning rates, on the self-supervised task distributions. All model hyper-parameters are kept the same so that any change in performance can be attributed to differences in the task distribution. See Supplementary for all hyper-parameters.

**Methods Evaluated** We consider all the different approaches to self-supervised task distributions described in Sec 3 and the baseline approach of SMLMT: (1) *Uniform*: this is the SMLMT approach of Bansal et al. (2020b) which use uniform random sampling over word-types; (2) *Frequency*: SMLMT with a sampling proportional to log-frequency (see 3.1); (3) *Cluster*: SMLMT where labels are picked from same word cluster (see 3.1); (4) *Dynamic*: curriculum-based task sampling with Cluster as the static distribution (see 3.2); (5) *Cluster-ccnet*: same as Cluster but using ccnet (Wenzek et al., 2020) as the corpora, which consists of web crawled data; (6) *SentCluster*: alternative to SMLMT which proposes tasks from subsets of sentence clustering (see 3.3); (7) *SentPair*: the sentence-pair tasks (see 3.4). All methods, except SentCluster and Cluster-ccnet, have Wikipedia as the text corpora. The sentence embeddings for SentCluster task distribution were obtained from Du et al. (2020), and consist of embeddings of about 1 billion sentences from ccnet (Wenzek et al., 2020). For this reason, we also report Cluster-ccnet that uses this same set of sentences. We found it beneficial to include 25% *Frequency* tasks in the *Cluster* task distribution and *SentPair* tasks are included in all other task distributions unless otherwise noted. Note that we only consider completely unsupervised meta-learning methods for fair evaluation. However our results improve over Bansal et al. (2020b) which showed improvements over BERT and multi-task BERT baselines. As we utilize the same dataset splits released in their work, our results can be directly compared.

[1]Code and datasets available at: https://github.com/thetb/meta_tasks

### 4.2 Analyzing task distributions

We start by a quantitative exploration of the various self-supervised task proposals without resorting to full fine-tuning on downstream tasks. Our goal here is to understand properties of these task distributions and how they relate to each other. To do this, we consider models meta-trained on a specific type of task proposal (rows in Table 1) and evaluate their performance in the few-shot setting on tasks sampled from all of the other task proposal methods (columns therein). We use r$i$ (or c$j$) below to refer to row $i$ (or column $j$) in the table.

We consider the following task proposal methods: Frequency (FREQ, c1): using the frequency-based word sampling in SMLMT; Inter-Cluster (X-C, c2): using the word-clustering approach explained in sec 3.1 but sampling all labels of task from different clusters; Intra-Cluster (I-C, c3&4): using the word-clustering approach explained in sec 3.1 which samples all labels of task from the same cluster; Sentence Cluster (S-C, c5): this is the sentence clustering approach to task proposal presented in sec 3.3. For evaluation, we consider 4-way tasks sampled from the above methods and evaluate average accuracy over 5000 tasks. We consider a BERT model (r1) which is not trained on the SMLMT distribution but is trained on the related masked language modeling (MLM) task. To enable evaluation of this model, we use it as a prototypical network model (Snell et al., 2017). We also consider meta-trained models trained on the SMLMT distribution with uniform sampling (Bansal et al., 2020b) (r2), frequency-based sampling (r3), and intra-cluster sampling (r4). Note that all models are trained on Wikipedia corpus.

Results are in Table 1. First, since BERT wasn't trained on any of the task distributions, we find low accuracy on all these tasks on r1, indicating that they contain information different than

| Model | FREQ | X-C | I-C | I-C (ccnet) | S-C (ccnet) |
|---|---|---|---|---|---|
| BERT | 43.1 | 43.3 | 37.7 | 38.7 | 66.3 |
| SMLMT (uniform) | 96.2 | 96.5 | **78.4** | **68.5** | 91.7 |
| SMLMT (frequency) | 96.8 | 96.9 | **79.6** | **70.0** | 91.0 |
| SMLMT (clustering) | 96.9 | 97.0 | 96.9 | **75.2** | 94.7 |
| Sentence Cluster | 69.2 | 71.2 | **53.0** | **45.0** | 98.9 |

Table 1: Analysis of task proposals. The columns are the different task proposal methods and rows are models trained on unsupervised task distributions. Low accuracy on a task distribution indicates harder to classify tasks or missing information in the training distribution (see Sec 4.2 for details).

what is learned from MLM. Moreover, the highest accuracy of this model is on Sentence Cluster tasks (r1c5; random baseline is 25%), even though the domain of this task is quite different than the training data of BERT. Next, lets consider the vanilla SMLMT model which uses uniformly random word sampling to create the meta-learning task distribution. Interestingly, we find that it gives high accuracy on frequency-sampled tasks (r2c1). Similarly, accuracy is high on the inter-cluster tasks (r2c2), even though the model wasn't meta-trained directly on this distribution. More importantly, performance drops significantly ($\approx 18\%$) on the tasks sampled using the intra-cluster approach (r2c3). This performance drops even further ($\approx 10\%$; r2c4) when the tasks are sampled from a different domain (common crawl) than the training domain of the model (Wiki). Accuracy on Sentence Cluster is also very high (r2c5), without training on this distribution. Models trained on frequency-based sampling perform similarly (r3). We also show the performance of a model trained on tasks sampled using the intra-cluster approach. Note that this model was trained on Wikipedia corpus, and even though it was trained on intra-cluster tasks, we still see a significant performance drop on intra-cluster tasks on a different domain (r4c4 vs r4c3). Finally, consider models trained on the sentence clustering tasks. These perform poorly on all of the tasks proposed by SMLMT (r5c1–4), indicating that this task distribution does not contain the same amount of information as SMLMT.

In summary, these results indicate that: (1) the intra-cluster tasks are more difficult than frequency-based sampling, and inter-cluster tasks are as easy as uniform-sampling (r2c2) (2) sentence cluster tasks are the easiest among all task proposals (c5), and training on this task distribution leads to poor performance on the SMLMT distributions (r5c1–4; but not vice versa), indicating lack of information in this distribution as compared to SMLMT. From this analysis we expect intra-cluster task distribution to be richer as compared to the other alternatives and models meta-trained on these should improve downstream performance over the others. As we will see in the next section, the downstream performance improvements are highly correlated with these unsupervised evaluations.

### 4.3 Evaluation on diverse downstream classification tasks

**Datasets** We consider all 17 downstream tasks in Bansal et al. (2020b) and 2 additional sentence-pair tasks. We group performance on datasets by the type of the task: (1) *Sentiment classification*: 4 domains (Books, DVD, Kitchen, Electronics) of Amazon review binary sentiment datasets (Blitzer et al., 2007); (2) *Rating classification*: 4 domain of 3-way classification based on ratings of reviews from the above Amazon datasets, 1 dataset on 3-way classification of tweets about sentiment towards Airlines; (3) *Entity typing*: CoNLL-2003 (Sang and De Meulder, 2003) entity mention classification into 4 coarse types, MIT-Restaurant (Liu et al., 2013) task on classifying mentions in user queries about restaurants into 8 types; (4) *Sentence-pair classification*: Scitail, a scientific natural language inference dataset (Khot et al., 2018), RTE task on textual entailment and MRPC task on paraphrase classification from the GLUE benchmark (Wang et al., 2018). (5) *Other text classification*: multiple social-media datasets on classifying tweets into (a) 2-way: political audience, bias or mention of a disaster, (b) 9-way: classifying based on political message, (c) 13-way: classifying emotion.

**Evaluation Protocol** We meta-train separate models on the self-supervised task distributions, without any access to the downstream supervised tasks. The models are then fine-tuned on the downstream task training sets which consist of $k = 8, 16, 32$ examples per class. Note that tasks can have different number of classes. Following Bansal et al. (2020b), we use the development set of Scitail and Amazon-Electronics to select the number of steps of fine-tuning for all models, all other hyper-parameters are kept the same as meta-training. Since few-shot performance is sensitive to the few examples in training, each model is fine-tuned on 10 sets for each task and the average test performance is reported with standard deviation.

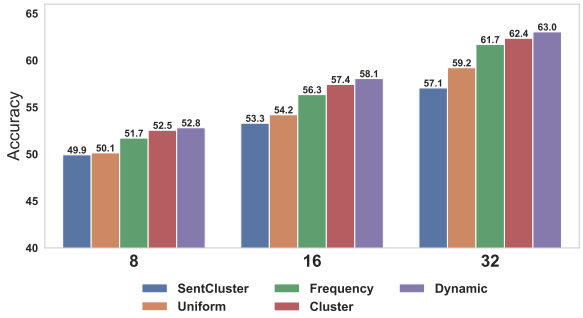


Figure 1: Overall average across 19 downstream tasks for the different task distributions proposed in this work. Cluster tasks and Dynamic curriculum lead to the best overall accuracy.

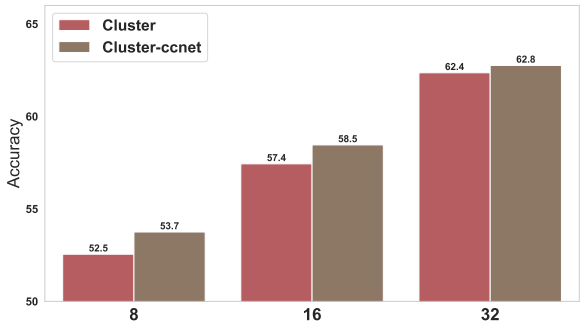


Figure 2: Changing domain of tasks from Wikipedia to CommonCrawl (ccnet) while keeping size of data, compute and model fixed. Overall average across 19 downstream tasks is shown. More diverse domain (ccnet) in training leads to improved down-stream accuracy.

**Results.** Table 2 shows the performance of all methods on different types of downstream tasks. We group datasets based on task type as described above and report the average performance over all the datasets in each group. First, note that the SentCluster approach is always inferior to any of the cloze-style approach, except on sentiment and rating classification where it is slightly better than SMLMT with Uniform sampling but worse than the other methods proposed here. Interestingly, replacing Uniform sampling with the simple Frequency sampling already leads to significant improvements throughout. Comparing the Cluster approach, we observe that this is better than Frequency on sentence-level tasks (like sentiment, rating, others), while slightly worse or comparable on sentence-pair tasks and phrase-level classification tasks (entity typing). Overall, the word-clustering approach to sampling labels for SMLMT are more preferable as they are often among the two highest performing on any task group or close to the highest performance. Note that our unsupervised

| Task Group | Model | $k$-shot | | |
|---|---|---|---|---|
| | | 8 | 16 | 32 |
| Sentiment Classification | Uniform | 59.1 ± 5.2 | 62.6 ± 5.3 | 69.6 ± 5.1 |
| | Frequency | 60.2 ± 4.8 | 65.2 ± 5.1 | 74.0 ± 5.4 |
| | Cluster | 62.2 ± 5.3 | 67.3 ± 5.9 | 75.9 ± 4.0 |
| | Dynamic | **63.6** ± 6.0 | **69.3** ± 6.3 | **77.3** ± 4.6 |
| | SentCluster | 61.1 ± 5.8 | 64.2 ± 5.7 | 70.4 ± 4.7 |
| | Cluster-ccnet | **64.7** ± 6.6 | **69.9** ± 7.1 | **76.2** ± 6.3 |
| Rating Classification | Uniform | 41.9 ± 7.2 | 47.3 ± 7.2 | 52.9 ± 7.6 |
| | Frequency | 42.6 ± 6.9 | 49.2 ± 7.2 | 55.1 ± 7.7 |
| | Cluster | 45.2 ± 7.7 | 51.9 ± 6.6 | 56.5 ± 7.1 |
| | Dynamic | **46.3** ± 8.1 | **53.5** ± 7.0 | **57.9** ± 7.3 |
| | SentCluster | 45.1 ± 8.8 | 48.7 ± 9.2 | 50.9 ± 9.0 |
| | Cluster-ccnet | **45.2** ± 7.5 | **52.1** ± 7.3 | **57.1** ± 7.8 |
| Entity Typing | Uniform | 61.4 ± 2.6 | 72.5 ± 4.8 | 81.4 ± 3.0 |
| | Frequency | 64.0 ± 3.0 | **73.0** ± 2.2 | **82.1** ± 2.0 |
| | Cluster | **64.5** ± 2.8 | 72.5 ± 2.6 | 81.3 ± 1.9 |
| | Dynamic | 62.4 ± 3.5 | 72.3 ± 3.3 | 81.6 ± 2.1 |
| | SentCluster | 51.7 ± 4.9 | 63.4 ± 4.5 | 73.4 ± 2.4 |
| | Cluster-ccnet | **70.7** ± 2.8 | **78.2** ± 3.1 | **84.1** ± 2.5 |
| Sentence Pair Classification | Uniform | 52.9 ± 5.2 | 54.1 ± 4.7 | 57.4 ± 5.7 |
| | Frequency | **59.5** ± 7.2 | **61.0** ± 8.5 | **63.6** ± 9.1 |
| | Cluster | 56.4 ± 5.3 | 59.5 ± 7.6 | 62.8 ± 8.6 |
| | Dynamic | 55.0 ± 4.9 | 57.8 ± 5.7 | 62.2 ± 8.5 |
| | SentCluster | 52.6 ± 4.7 | 52.9 ± 2.9 | 54.0 ± 3.8 |
| | Cluster-ccnet | **55.9** ± 5.7 | **58.5** ± 6.9 | **62.9** ± 6.9 |
| Other Text Classification | Uniform | 44.8 ± 3.9 | 47.5 ± 2.3 | 49.4 ± 2.1 |
| | Frequency | 44.4 ± 3.5 | 47.3 ± 2.0 | 49.1 ± 1.9 |
| | Cluster | 45.0 ± 3.7 | 48.1 ± 2.0 | 49.5 ± 1.9 |
| | Dynamic | **45.5** ± 3.5 | **48.5** ± 2.2 | **49.8** ± 1.9 |
| | SentCluster | 43.5 ± 4.1 | 45.7 ± 2.5 | 47.8 ± 1.7 |
| | Cluster-ccnet | **46.6** ± 3.4 | **48.9** ± 2.2 | **49.9** ± 1.8 |

Table 2: Results on downstream tasks. Best performing model for each $k$ and each task group is in bold and the second best is underlined.

analysis in Sec 4.2 also reflected that training on the Cluster task distribution should be better compared to others. Finally, note that using the Dynamic curriculum over task sampling further improves the performance over cluster-based approach. This overall trend is also clearly reflected in the overall average performance across all the 19 tasks in Figure 1. Figure 2 further shows that, for the Cluster tasks, constructing tasks from more diverse domain such as CommonCrawl can improve downstream performance even when using the same amount of data for training.

**Ablation over SentPair.** We introduced the sentence pair task to enable better learning of sentence pair tasks such as natural language inference. These task remove the train-test mismatch in the input format as the sentence pair tasks contain pairs of sentences as input where as SMLMT only proposes single sentence classification tasks. To assess

| Task | Model | 8 | 16 | 32 |
|---|---|---|---|---|
| MRPC | Clustering | 54.63 ± 4.69 | 54.00 ± 3.63 | 58.28 ± 4.90 |
| | + SentPair | **55.88** ± 6.68 | **57.13** ± 5.15 | **60.12** ± 3.58 |
| Scitail | Clustering | **60.63** ± 4.29 | 59.89 ± 4.20 | 67.89 ± 5.59 |
| | + SentPair | 58.86 ± 4.81 | **67.94** ± 2.92 | **73.56** ± 2.79 |

Table 3: Ablation: training with & without SentPair.

| Static Distribution | $\lambda_t$ | 8-shot | 16-shot | 32-shot |
|---|---|---|---|---|
| Cluster | 0.5 | 54.0 | 58.7 | 63.0 |
| Cluster | 0.25 | 55.2 | 59.0 | 64.6 |
| Frequency | Anneal | 56.5 | 60.9 | 65.8 |
| Cluster | Anneal | **56.8** | **61.7** | **66.4** |

Table 4: Ablation: static tasks and the value of mixing proportion $\lambda_t$ used in dynamic curriculum.

the efficacy of the SentPair task, we trained a word-cluster model with and without the SentPair task and evaluated it on few-shot sentence pair tasks of Scitail and MRPC. Results are in Table 3. We can see that the unsupervised SentPair task improves performance under most settings, sometimes by large margins up to 8% absolute.

**Ablation for dynamic curriculum.** The dynamic curriculum over tasks requires two crucial choices: the static distribution and the value of mixing proportion $\lambda_t$. We ablate over choices for these in Fig. 4 which reports average performance over 5 tasks, one each from the task groups considered. We find that using the Cluster tasks, created from static pre-computer word-embeddings, works better than using Frequency-based tasks as the static distribution. Moreover, annealing $\lambda_t$ from 0 to 1 over the training epochs is better than using a fixed value of $\lambda_t$ throughout training.

### 4.4 Evaluation on FewRel 2.0 benchmark

FewRel (Han et al., 2018; Gao et al., 2019) is a common benchmark for few-shot learning in NLP, which consists of many few-shot relation classification tasks created by sub-sampling from a pool of relation labels. The resemblance to the popular few-shot benchmarks like MiniImageNet (Vinyals et al., 2016) makes FewRel one of the few widely used datasets for training and evaluating NLP meta-learning methods.

Before submitting to the competition site for test set results, we first use the validation set to select the best model(s), where we observed that the Cluster approaches performs better than the other task proposals (see validation results in Supplementary).

| Model | 1-shot | | 5-shot | |
|---|---|---|---|---|
| | $N = 5$ | $N = 10$ | $N = 5$ | $N = 10$ |
| *Unsupervised* | | | | |
| Cluster | 60.1 | 44.1 | 78.8 | 65.2 |
| Cluster-ccnet | 61.3 | 46.1 | 80.4 | 67.7 |
| *Supervised* | | | | |
| Proto-Adv (CNN) | 42.2 | 28.9 | 58.7 | 44.4 |
| Proto-Adv (BERT) | 41.9 | 27.4 | 54.7 | 37.4 |
| BERT-Pair | 67.4 | **54.9** | 78.6 | 66.9 |
| Cluster-ccnet | **67.7** | 52.9 | **84.3** | **74.1** |

Table 5: Results on Fewrel 2.0 test set.

We then compare their test set performance with previously published results: the BERT-Pair, Proto-Adversarial (CNN), and Proto-Adversarial (BERT) are *supervised* meta-learning models trained on FewRel training data and using BERT or CNN as the text encoder. See Gao et al. (2019) for details. Interestingly, our *unsupervised* meta-learned models that do not use any FewRel training data outperform the supervised baselines in the 5-shot setting. Performance is lower than BERT-Pair on 1-shot tasks, potentially because our models have not been trained for 1-shot tasks like BERT-Pair. Finally, fine-tuning our best model on the FewRel training data leads to the best overall performance.

## 5 Related Work

Meta-learning applications in NLP have yielded improvements on specific tasks (Gu et al., 2018; Chen et al., 2018; Guo et al., 2018; Yu et al., 2018; Han et al., 2018; Dou et al., 2019). Unsupervised meta-learning has been explored in computer vision (Hsu et al., 2019; Khodadadeh et al., 2019) and reinforcement learning (Gupta et al., 2018). Hsu et al. (2019) cluster images using pre-trained embeddings to create tasks. Metz et al. (2019) meta-learn an unsupervised update rule in a semi-supervised framework. Bansal et al. (2020b) developed the SMLMT approach to unsupervised meta-learning in NLP. Contemporary work (Murty et al., 2021) explored the use of clustering, though focused only on natural language inference tasks. Curriculum learning (Bengio et al., 2009) in the context of meta-learning has been unexplored in NLP, prior to this work. Jabri et al. (2019) found unsupervised curriculum to be beneficial for meta-reinforcement learning. We refer to Hospedales et al. (2020) for a comprehensive review of meta-learning.

Self-supervised learning has emerged as an efficient approach to representation learning in NLP

(Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2018; Radford et al., 2019; Yang et al., 2019). Multi-task learning of pre-trained models has shown improved results on many tasks (Phang et al., 2018; Liu et al., 2019a), including few-shot setting. Yin et al. (2020) leveraged entailment tasks for few-shot learning. Du et al. (2020) developed self-training methods for semi-supervised few-shot learning. Recently, extremely large language models have been shown to have few-shot capacities (Brown et al., 2020), while Schick and Schütze (2020) demonstrated few-shot capacities for small models in the semi-supervised setting. Meanwhile, Bansal et al. (2020a,b) showed meta-learning to be effective at improving few-shot performance in multi-task and unsupervised settings, as well as improving performance for small models.

## 6 Conclusion

We explored several approaches to self-supervised task distribution for meta-learning. Our results demonstrate improvements in few-shot performance over a wide-range of classification tasks. This demonstrates the utility of meta-learning from unlabeled data, opening up the possibility of large-scale meta-learning for pertinent applications in NLP such as continual learning, architecture search, learning for low-resource languages, and more.

## Acknowledgements

This work was supported in part by the Chan Zuckerberg Initiative, in part by IBM, and in part by the National Science Foundation under Award No. 1763618. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

## References

R Harald Baayen. 2002. *Word frequency distributions*, volume 18. Springer Science & Business Media.

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

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

Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. 1992. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pages 6–8. Univ. of Texas.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Junkun Chen, Xipeng Qiu, Pengfei Liu, and Xuanjing Huang. 2018. Meta multi-task learning for sequence modeling. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. 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 1192–1197.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. 2020. Self-training improves pre-training for natural language understanding. *arXiv preprint arXiv:2010.02194*.

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

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6251–6256, Hong Kong, China. Association for Computational Linguistics.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *International Conference on Machine Learning*, pages 1311–1320. PMLR.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. 2018. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*.

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

Abhishek Gupta, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. 2018. Unsupervised meta-learning for reinforcement learning. *arXiv preprint arXiv:1806.04640*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Empirical Methods in Natural Language Processing*, pages 4803–4809.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

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

Kyle Hsu, Sergey Levine, and Chelsea Finn. 2019. Unsupervised learning via meta-learning. In *International Conference on Learning Representations*.

Allan Jabri, Kyle Hsu, Abhishek Gupta, Ben Eysenbach, Sergey Levine, and Chelsea Finn. 2019. Unsupervised curricula for visual meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, pages 10519–10531. Curran Associates, Inc.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Siavash Khodadadeh, Ladislau Boloni, and Mubarak Shah. 2019. Unsupervised meta-learning for few-shot image classification. *Advances in neural information processing systems*, 32.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

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

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. 2019. Learning unsupervised learning rules. In *International Conference on Learning Representations*.

Shikhar Murty, Tatsunori Hashimoto, and Christopher D Manning. 2021. Dreca: A general task augmentation strategy for few-shot natural language inference. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1113–1125.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Jürgen Schmidhuber. 1987. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. Ph.D. thesis, Technische Universität München.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Suzanna Sia, Ayush Dalmia, and Sabrina J Mielke. 2020. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1728–1736.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. 2019. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7251–7259.

Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn*. Springer Science & Business Media.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239.

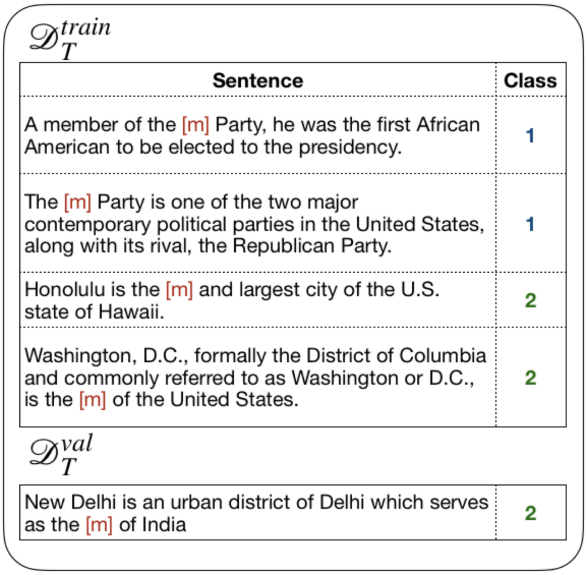


Figure 3: Example of a task in SMLMT.

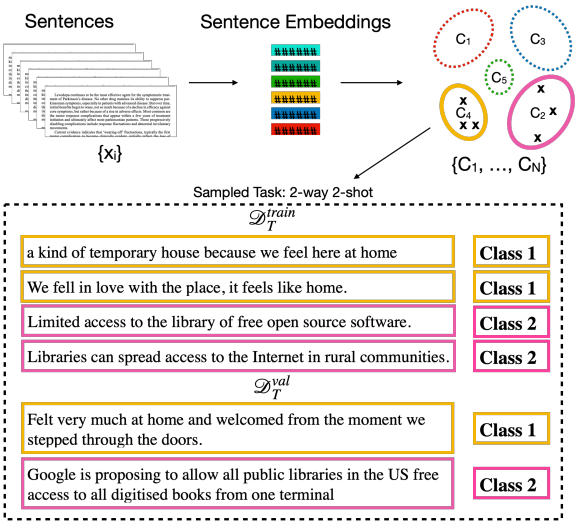


Figure 4: Illustration of SentCluster approach.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *CoRR*, abs/1901.11373.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. *arXiv preprint arXiv:1805.07513*.

## A Appendix

Examples of SMLMT and SentCluster can be seen in Figures 3 and 4.

| Model | 1-shot | | 5-shot | |
|---|---|---|---|---|
| | $N = 5$ | $N = 10$ | $N = 5$ | $N = 10$ |
| SentCluster | 34.2 | 21.2 | 52.3 | 36.3 |
| Uniform | 55.8 | 40.1 | 76.1 | 61.0 |
| Frequency | 57.6 | 41.6 | 78.1 | 61.5 |
| Cluster | 60.4 | 45.2 | 78.1 | 63.9 |
| Cluster-ccnet | **60.1** | **46.0** | **81.2** | **67.6** |

Table 6: Results on Fewrel 2.0 validation.

| Hyper-parameter | Value |
|---|---|
| Tasks per batch | 4 |
| $d$ | 256 |
| Attention dropout | 0.1 |
| Hidden Layer Dropout | 0.1 |
| Outer Loop Learning Rate | 1e-05 |
| Adaptation Steps ($G$) | 7 |
| Meta-training Epochs | 1 |
| Lowercase text | False |
| Sequence Length | 128 |
| Learning-rate Warmup | 10% of steps |
| SentPair ratio $\alpha$ | $\frac{1}{16}$ |
| Number of Tasks | 4 Million |
| Support samples per task | 80 |
| Query samples per task | 10 |
| Number of classes for tasks | [2,3,4,5] |
| Number of Clusters $M$ | 500 |
| Number of Clusters in SentCluster | 200k |
| $\lambda_t$ in Dynamic | Anneal |
| $m$ interval in Dynamic | 5000 |

Table 7: Hyper-parameters. The parameters relating to the task distributions are in the bottom section of the table.

### A.1 Additional Experiment Results

Results on FewRel 2.0 validation set using the different task distributions is shown in Figure 6. Full distribution of results in down-stream takss for the various self-supervised tasks can be seen in Fig. 5

### A.2 Fine-tuning hyper-parameters

The meta-learning methods learn the learn rate for fine-tuning, thus we only tune the number of steps to run fine-tuning by using development data from 2 tasks (Scitail, Amazon Electronics), following Bansal et al. (2020a,b). We found that running fine-tuning until the loss on the support set is small ($<= 0.01$) is an alternative that also performs competitively and does not require tuning the number of steps. The reported results followed the previous approach and the tuned number of steps of fine-tuning for $k = 8, 16, 32$ respectively were: (1) Uniform: 100,75,100 (2) Frequency: 25,150,75 (3) Cluster: 75,50,75 (4) Cluster-ccnet: 150,200,75 (5) SentCluster: 100,250,25 (6) Dynamic: 10, 100, 200. On FewRel we found 20 steps of updates on the support set to perform well on the validation data for all settings.

### A.3 Other Implementation Details

Since the Fewrel tasks consist of entity pair in the sentence it is important to mark these entities which define the relation to be classified. We used unused tokens in the BERT vocabulary to mark the positions of the entity mentions. Note the in the unsupervised models these unsused tokens get a zero-embedding and are only fine-tuned from the 1-shot or 5-shot support sets.

Hyper-parameters for meta-training are listed in Table 7. Dataset statistics for downstream classification tasks can be found in Bansal et al. (2020a) and few-shot splits can be downloaded from https://github.com/iesl/leopard.

Training Hardware: The models were trained on 32 V100 GPU. Training takes about 42 hours.

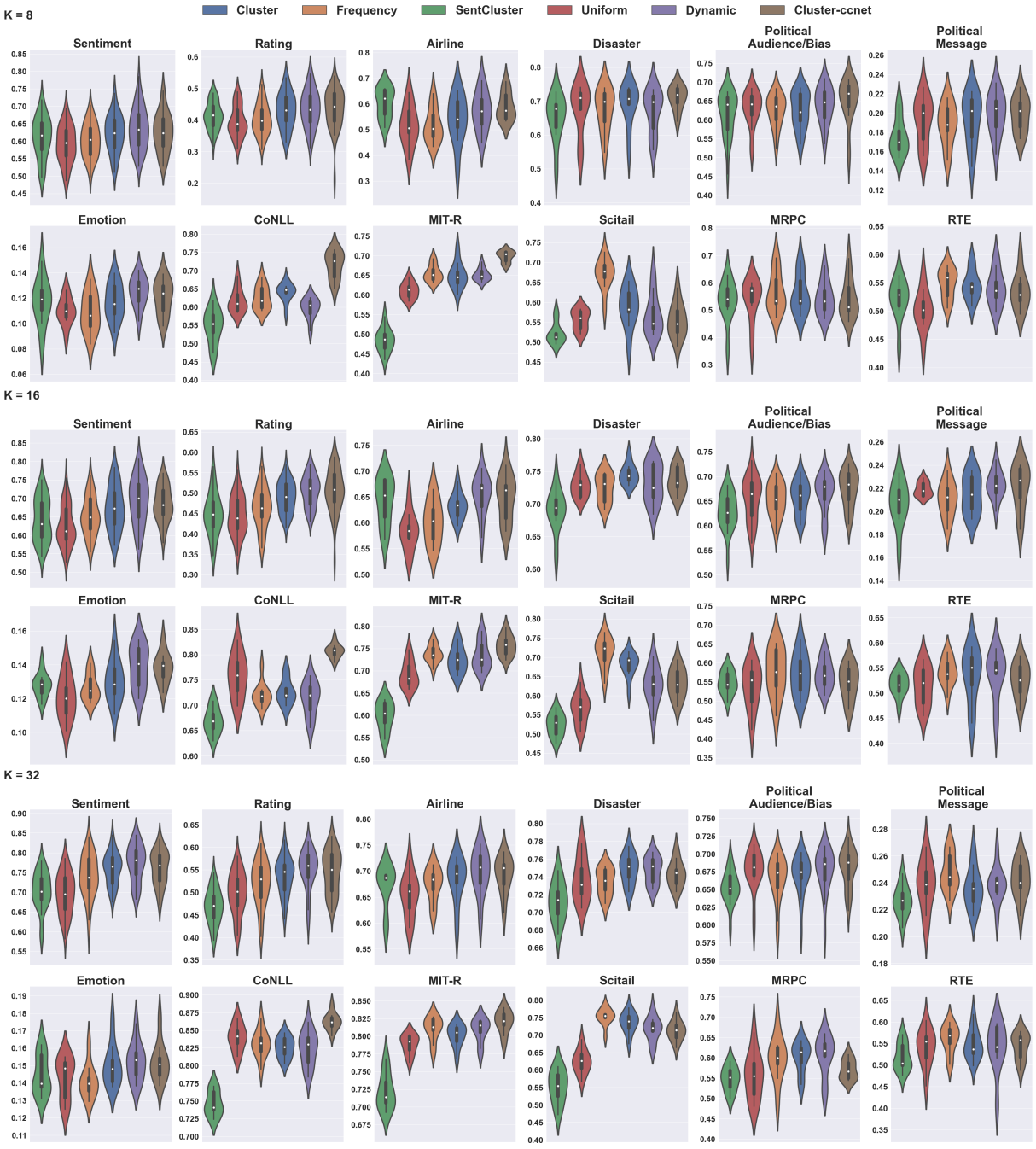


Figure 5: Results across all tasks. Sentiment and Rating are average of 4 domains used in (Bansal et al., 2020a).Each violin plot for a model shows the full distribution of accuracy across multiple runs (and domains).