Clustering-based Sampling for Few-Shot Cross-Domain Keyphrase Extraction

Prakamya Mishra[♣,*], **Lincy Pattanaik**[♣,*], **Arunima Sundar**[♣,*]
Nishant Yadav[♣], **Mayank Kulkarni**[♠]
[♣]University of Massachusetts, Amherst [♠]Amazon AGI
{prakamyamish, lpattanaik, asundar, nishantyadav}@umass.edu
maykul@amazon.com

Abstract

Keyphrase extraction is the task of identifying a set of keyphrases present in a document that captures its most salient topics. Scientific domain-specific pre-training has led to achieving state-of-the-art keyphrase extraction performance with a majority of benchmarks being within the domain. In this work, we explore how to effectively enable the cross-domain generalization capabilities of such models without requiring the same scale of data. We primarily focus on the few-shot setting in non-scientific domain datasets such as OpenKP from the Web domain & StackEx from the StackExchange forum. We propose to leverage topic information intrinsically available in the data, to build a novel clustering-based sampling approach that facilitates selecting a few samples to label from the target domain facilitating building robust and performant models. This approach leads to large gains in performance of up to 26.35 points in F1 when compared to selecting few-shot samples uniformly at random. We also explore the setting where we have access to labeled data from the model's pretraining domain corpora and perform gradual training which involves slowly folding in target domain data to the source domain data. Here we demonstrate further improvements in the model performance by up to 12.76 F1 points.

1 Introduction

Keyphrases are a set of words that convey the most salient topics of an article or a document, and identification of such keyphrases can be very useful in extracting key information from the long documents through summarization (Zhang et al., 2004; Qazvinian et al., 2010), semantic and faceted search (Gutwin et al., 1999; Sanyal et al., 2019) and document retrieval (Jones and Staveley, 1999). Recently, a lot of work has been done in using language models (LMs) for extracting keyphrases using generative models through keyphrase generation (Zhang et al., 2017; Meng et al., 2017; Chen et al., 2018; Ye and Wang, 2018; Chen et al., 2019; Yuan et al., 2020; Ye et al., 2021). However, in this work we focus on encoder-only keyphrase extraction (Alzaidy et al., 2019; Sahrawat et al., 2020; Martinc et al., 2020; Tokala et al., 2020), specifically framing the task as a sequence tagging in the BIO schema format (Sahrawat et al., 2020; Kulkarni et al., 2022). KBIR (Kulkarni et al., 2022) showed that the task and domain-specific pre-training helps in learning rich representations of the keyphrases and leads to better downstream keyphrase extraction performance compared to models that are pre-trained using a task-agnostic objective like Masked Language Modeling. Task-specific pre-training of LMs for keyphrase extraction requires abundance of supervised data with documents and their corresponding keyphrases. Obtaining human annotated data can be a very expensive, error-prone and an inefficient process, hence a majority of the labelled datasets for keyphrase extraction are from the scientific domain (Hulth, 2003; Krapivin and Marchese, 2009; Kim et al., 2010; Augenstein et al., 2017; Meng et al., 2017), as authors provide keywords with their scientific article to improve discoverability. However, pre-training on domain-specific data often results in poor downstream keyphrase extraction performance on out of domain data.

Fine-tuning with a sufficiently large dataset typically allows the model to generalize well beyond the pre-training domain. However, for low-resource domains, such data can be difficult to obtain at scale. Few-shot learning is a setup extensively explored with very large language models and typically in-context (Brown et al., 2020; Lin et al., 2022; Srivastava et al., 2022), however we focus on the more niche setup of few-shot learning using fine-tuning for sequence tagging with encoder-only models. Keyphrase-aware PLMs are trained

[*]Indicates equal contribution

Findings of the Association for Computational Linguistics: EACL 2024, pages 1232–1250

to build strong representations for keyphrases in text and we hypothesize that we are able to leverage these embeddings to bootstrap a model by fine-tuning it only a few-samples from the target domain in order to obtain satisfactory performance.

In this work we investigate what sampling strategy, given a limited budget of up to 100 annotations, allows us to select data points from a low-resource target domain for annotation that would be the most effective few-shot samples for fine-tuning. We further explore if we can leverage access to scientific-domain pre-training data OAGKx (Çano and Bojar, 2020) used by the present state-of-the-art keyphrase extraction model, KBIR (Kulkarni et al., 2022) to bootstrap model performance. The main contributions of this work are summarised below:

- We explore the generalization capabilities of the KBIR model on two datasets simulated as low-resource target domains, OpenKP (Xiong et al., 2019) & StackEx (Yuan et al., 2020), using few-shot learning through fine-tuning with a sequence tagging training objective with encoder-only models.
- We propose a novel clustering-based few-shot sampling approach that leverages intrinsically available sub-domain information as topics from the dataset to extract few-shot samples to be labelled from the target domains and be used for fine-tuning. This leads to significant gain in performance across two different training regimes compared to sampling few-shot datapoints uniformly at random.
- We also demonstrate through a case study of several variants of Clustering-based sampling using Jaccard similarity, Cosine similarity and ChatGPT (OpenAI, 2023) prompting to improve diversity in the few-shot samples and show this does not correlate with model performance.

2 Related Work

Keyphrase Extraction We focus on encoder-only models that perform keyphrase extraction as a sequence tagging task (Alzaidy et al., 2019; Sahrawat et al., 2020; Martinc et al., 2020; Tokala et al., 2020) that require fine-tuning with labelled data for a given domain. Unsupervised keyphrase extraction (Mihalcea and Tarau, 2004; Rose et al., 2010; Campos et al., 2020; Schopf et al., 2022) is an area of research that focuses on scaling to multiple domains without the need for retraining models (Zero-Shot) but rather focusing on language structure to identify keyphrases. However, Unsupervised methods typically underperform their Fine-tuned counterparts for a given domain. We aim to bridge the gap between these two methods by using as little data as possible (Few-shot). The KBIR model (Kulkarni et al., 2022) demonstrates that using only 130 training samples from SemEval 2010 (Kim et al., 2010) where the domain aligns with pre-training domain, is sufficient to obtain state-of-the-art results despite seeing very few data points. This serves as our motivation to further explore few-shot fine-tuning as sequence labeling for keyphrases and also propose methods to bootstrap performance for different domains.

Domain Adaptation Teaching a model to maximize performance on a single low-resource (target) domain, by leveraging a single high-resource (source) domain is a well studied area in NLP (Chelba and Acero, 2004; Florian et al., 2004; Blitzer et al., 2006; Daumé III, 2007; Blitzer et al., 2007; Peng and Dredze, 2017). Wang et al., 2020 propose an effective learning procedure, Meta Fine-Tuning (MFT) that learns the embeddings of class prototypes from multi-domain training sets and assigns topicality scores using the kNN-augmented Example Selection (KATE) (Liu et al., 2022b). However, our setup differs from traditional domain adaptation in that we want to adapt from the pre-training source domain rather than a fine-tuned source domain to a fine-tuned target domain.

Few-Shot Learning With the advent of larger generative models few-shot learning has become a popular paradigm where the samples are provided in the prompt and in-context learning is leveraged to improve performance (Brown et al., 2020; Lin et al., 2022; Srivastava et al., 2022). An extension of this work demonstrates that fine-tuning such large generative models (Liu et al., 2022a) and encoder-based models (Logan IV et al., 2022) results in better performance by recasting classification tasks as generation tasks, with contemporary work making a fair comparison between both these approaches (Mosbach et al., 2023). Cross-Domain Few-Shot fine-tuning has been explored for Named Entity Recognition (NER) in an N-way K-shot setting, where multiple (N) domains trained on large

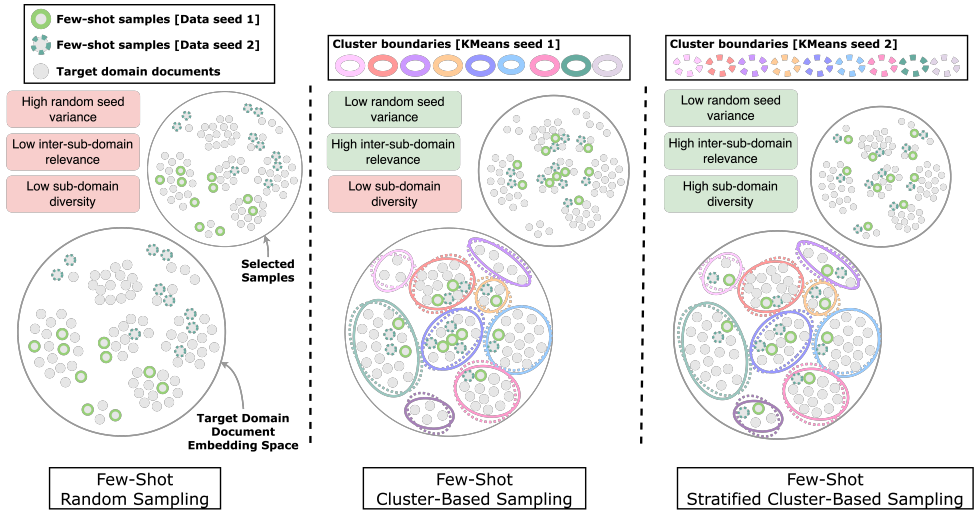


Figure 1: Demonstration of few-shot sample selection from a target domain document embedding space using several sampling approaches.

amounts of source domain NER data and few-shot (K) samples are used for target training (Fang et al., 2023; Das et al., 2022; Hou et al., 2020). However, to the best of our knowledge these techniques have not been explored to conduct few-shot fine-tuning when using critically few samples.

3 Few-shot Keyphrase Extraction

In this work, we investigate if we can effectively sample data from target domains D_t having N documents, to be annotated and used for fine-tuning in a few-shot setting. In line with prior work (Sahrawat et al., 2020; Kulkarni et al., 2022), we setup keyphrase extraction as a sequence tagging task using the BIO schema (B-KEY, I-KEY, O) using HuggingFace (Wolf et al., 2020). Given a sequence of tokens $x_i = \{x_i^1, ..., x_i^n\}$, the model is trained to predict a sequence of labels $y_i = \{y_i^1, ..., y_i^n\}$, where each $y_i^j \in$ {B-KEY, I-KEY, O} label represents whether the j^{th} input token of the i^{th} document in D_t is either a beginning of the keyphrase (B-KEY), inside of the keyphrase (I-KEY), or outside of the keyphrase (O). We further quantify the impact of obtaining labeled data in the source (pre-training) domain D_s having M documents. As our sampling strategies do not rely on labels we simulate low-resource domains in large-scale labelled data allowing us to train on a few data points but evaluate on a large number of high-quality test points. The use of the labeled data is considered the equivalent of an annotation and we don't conduct any annotation ourselves.

3.1 Access to only Target Domain Data

For keyphrase extraction in a cross-domain setting where there is no availability of labelled data from the source domain (pre-training data D_s), few-shot fine-tuning of the pre-trained model is done using a small number of k samples $X_* = \{x_i^*, .., x_k^*\}$ only from the target domain D_t, in order to adapt the source domain model to the new domain. Here sampling approaches can play a major role in contributing to the cross-domain model performance. In this section, we explore sampling approaches to improve few-shot model performance in cross-domain settings where there is no availability of labelled data from the source domain.

3.1.1 Random Sampling

One of the most common and widely used methods for extracting samples for few-shot learning is Random Sampling (Lin et al., 2022; Cong et al., 2021). We used random sampling to establish a baseline for the few-shot keyphrase extraction, where a small number of samples k are selected uniformly at random ($X^* : \{x_i^*, ..., x_k^*\} \leftarrow U(D_t, k)$)[1] from D_t to fine-tune the KBIR model and its vanilla counterpart RoBERTa in a few-shot setting. The algorithm for random sampling is shown in App. F.

Random sampling is easy to implement and does not add any computational overhead to the sampling process. One of the limitations of such a sampling approach is that it is a lottery-based approach

[1] $U(D_t, k)$ samples k documents from D_t uniformly at random.

where it is equally like to select high-quality as well as low-quality samples, resulting in high variation in the performance of the model (Zhang et al., 2020; Schick and Schütze, 2020). For example, Fig. 1 illustrates how different subsets of samples can be selected using few-shot random sampling based on different data seeds. As shown in the figure, in the case of random sampling, both the data seeds (seed for random sampling) select samples that belong to the different topical segments (upper & lower hemisphere of the target domain document embedding space) of the target domain datasets, which might lead to high variation in the few-shot training data distribution with respect to the fixed target domain data distribution in the few-shot setting.

3.1.2 Clustering-based Sampling

Random Sampling on the other hand leads to high variance in sample selection and also might results in low diversity in selected samples w.r.t target domain causing poor domain adaption in the models trained in few-shot cross-domain settings.

In this work, we propose a clustering-based sampling approach that leverages topic information intrinsically available in the target domain data for selecting high-quality few-shot samples for robust domain adaption in cross-domain settings.

Given just D_t, we hypothesize that there exist a set of k samples $X_* = \{x_1^*, ..., x_k^*\}$ in the target domain dataset that can be used to train a model in a few-shot cross-domain setting that can maximize its generalization capabilities, robustness, and performance on the downstream task. A target domain can consist of several subdomain topics as shown in Fig. 1, and in order to train a model to generalize on the target domain using X_* from the target domain, each x_i^* should have the maximum coverage over all these sub-domain topics and should be representative of D_t.

In the clustering-based sampling approach, we first identify these sub-domains and documents belonging to these subdomains using KMeans clustering. We extract d-dimensional sentence embeddings $E_t = \{e_1^x, ...e_N^x\}$ of all the x_i in D^t using Sentence Transformer (Reimers and Gurevych, 2019), and use KMeans clustering on top of E_t to create c sub-domain clusters $C = \{C_1, ...C_c\}$ of D^t. We use C to generate d-dimensional sub-domain embeddings $E_C = \{e_1^C, ...e_c^C\}$ for each of the c sub-domains (sub-domain centers), which will represent the topic of the corresponding sub-domain. Here the sub-domain embeddings e_i^C embeds information about the sub-domain topic corresponding to C_i, and are computed by taking the mean over $\forall e_i^x$ corresponding to $x_i \in C_i$. We use E_C to give a score to each x_i in D^t, representing a relevance score of x_i to all the sub-domain topics corresponding to the clusters in C. In order to identify high-quality representative samples X_*, we use a cosine-similarity-based scoring function that would give a higher score to a sample that has high relevance with all the sub-domain topics. Given a document $x_i \in D_t$ having an embedding e_i^x, we score x_i using the scoring function defined in equation 1, where δ represents the cosine-similarity between two d-dimensional embeddings. The documents are then ranked based on their scores (s_i) and the top-scoring k documents are selected as the few-shot samples represented by $X_* = \{x_1^*, ..., x_k^*\}$, as shown in equation 2. The algorithm for clustering-based sampling is shown in App. F.

$$S : \{s_1, ..., s_N\}; s_i = (\sum_{j=1}^{c} \delta(e_j^C, e_i^x))/c \quad (1)$$

$$X_* = \{x_1^*, ..., x_k^*\} = \arg topk_{x_i \in D_t}(S) \quad (2)$$

As shown in Fig. 1, such a clustering-based sampling approach in a few-shot cross-domain setting would generate samples that are not only representative of the target domain, i.e., are relevant to the majority of sub-domain topics, but are also relatively robust to different KMeans seeds.

Although the clustering-based few-shot sampling approach will select high-quality representative samples from the target domain, they still might lack diversity as most of these samples can come from only the sub-domain clusters that are more general in nature. This might lead to missing samples from highly localized sub-domain topics, which in turn results in compromising the optimal representational capacity of selected few-shot samples w.r.t to the target domain.

In order to select samples evenly from such localized sub-domains, we propose another variant of clustering-based sampling called **Stratified Clustering-based sampling**. In this variant of clustering-based sampling, the few-shot samples are first ranked based on the scoring function defined in equation 2, and then a proportionately equal number of top-scoring samples within each cluster are selected to create a set of k few-shot samples. Here the proportion of samples (w.r.t sub-domains) in the few-shot samples is consistent with

their corresponding proportions in the target domain. The stratified variant of the clustering-based sampling approach slightly compromises on selecting top-scoring samples in order to increase diversity and representativeness in the samples by even incorporating samples from localized sub-domains (App. D.2).

3.2 Access to Source Domain Data

In the cross-domain setting where we also have access to the source-domain data D_s (pre-training domain), along with D_t, it is beneficial to use both of them together to better fine-tune a pre-trained model for domain adaption (Xu et al., 2021). In this section, we explore the gradual training setup (Xu et al., 2021), and how we incorporate clustering-based sampling in it.

3.2.1 Gradual Training

Both the random and clustering-based sampling approaches only sample data from D_t which can have a significant drift in distribution from D_s. Fine-tuning a pre-trained model in such a setting using only the D_t can limit its domain adaption on a new domain with significant distribution drift. So in this work, we also explore the gradual training setup for smoother domain adaption in a cross-domain few-shot setting.

In the gradual training setup, we iteratively re-train a pre-trained model using k few-shot samples having different concentrations (k_1:k_2) of both the target domain as well as the source domain respectively, chosen uniformly at random. In each iteration, the model is initialized with the trained weights from the previous iteration. In the first iteration, we start with the pre-trained weights, and in the later iterations, we increase the concentration of target domain few-shot samples by increasing the number of target domain samples and differently from the original work, decreasing the number of source domain samples for smoother domain adaptation from source to the target domain. In such a few-shot training setup, the model is iteratively re-trained on a set of few-shot samples whose distribution gradually shifts from the source domain to the target domain leading to smoother data distribution shift compared to direct fine-tuning on the target, resulting in smoother domain adaption.

While such a training setup leads to a smoother domain adaption, it also comes with an increase in the computational cost by a factor of the number of iterations involved.

3.2.2 Gradual Training + Clustering-based Sampling

In section 3.1.2 we explained how using clustering-based few-shot sampling approaches leads to a relatively higher-quality representative (w.r.t target domain) sample selection from the target domain data compared to random sampling, resulting in better domain adaption in the few-shot cross-domain setting. So in this work, we also explore a gradual training setup where instead of sampling target domain samples uniformly at random, we select few-shot samples using clustering-based sampling approaches. Doing so would not only lead to a smoother data distribution shift in the few-shot samples because of gradual training but also will use relatively higher-quality representative samples from the target domain for few-shot cross-domain iterative training.

4 Experimental Setup

In this work, we investigate the generalization capability of the **KBIR** model and its vanilla counterpart **RoBERTa**, on the keyphrase extraction task on out-of-domain datasets with respect to the scientific domain-specific OAGKx (Çano and Bojar, 2020) dataset on which KBIR was pre-trained.

	Train	Validation	Test
OpenKP	134K	6.6K	6.6K
StackEx	300K	16K	16K

Table 1: Dataset statistics for **OpenKP** & **StackEx**

4.1 Data

We conduct our cross-domain experiments on the OpenKP (Xiong et al., 2019) dataset that consists of documents from a collection of Bing search web pages and the StackEx (Yuan et al., 2020) dataset that consists of question-answer pair articles from Stack Exchange website[2]. Both these datasets are from non-scientific domain consisting of documents from various sub-domains like news, politics, healthcare, movies, programming, music and so on. Dataset statistics are provided in Table 1. We uniformly sample the train set down to 22k for computational efficiency. We use **OpenKP** and **StackEx** datasets as **target domains** and use **OAGKx** as the **source domain**.

[2]https://stackexchange.com/

4.2 Implementation Details

We conduct our experiments over multiple exemplars and models and multiple weight initialization, data sampling, and clustering center seeds to ensure statistical significance. Details on hyperparameters, clustering setup and evaluation are available in Appendix A, B and C respectively.

4.3 Baselines & Upperbounds

Random: We randomly initialize the classification head weights for KBIR and RoBERTa to perform inference.

PatternRank: We use the current state-of-the-art unsupervised keyphrase extraction in PatternRank (Schopf et al., 2022) that leverages part-of-speech tag matching and BERT-based models to generate candidate keyphrases and serves as our strong baseline.

MANNER: We use MANNER (Fang et al., 2023), a Cross-Domain Few-Shot support and query-based architecture in an N-way K-Shot sequence tagging framework as a strong baseline. We conducted a thorough literature review of Cross-Domain Few-Shot setups to find similar setups for Named Entity Recognition in MANNER (Fang et al., 2023) that we had to make minor adjustments to serve as a strong baseline. Fang et al. (2023) leverages a support and query based architecture to setup an N-way K-shot cross-domain sequence tagging framework that has demonstrated to be very effective outperforming previous SoTA such as CONTaiNER (Das et al., 2022) and L-TapNet (Hou et al., 2020). A major caveat is that they use significantly more data (> 1000 samples) in their few-shot experiments and even more data to conduct source domain training. We recreated these experiments by maintaining the number of data points seen across the training as K=[5, 10, 50, 100] to be comparable with our best performing model setting. We do so in both settings where source domain data is and isn't available for training.

Full-Fine Tune: We use the aforementioned 22k uniformly sampled data points from a given target dataset in order to fine-tune the model for upper bound performance.

	Dataset	KBIR	RoBERTa	PatternRank
Zero-shot	OpenKP	1.64	1.82	7.4
	StackEx	1.00	0.07	15.38
Full Finetune	OpenKP	48.43	50.62	N/A
	StackEx	62.20	60.99	N/A

Table 2: Zero-shot and full fine-tuning exact match F1-score performances

4.4 Few-shot Learning

4.4.1 Access to only Target Domain Data

Random Sampling (R): We select k few-shot samples uniformly at random only from the target domain as the few-shot samples (Section 3.1.1).

Clustering-based Sampling (C): We select k top-scoring samples only from the target domain as the few-shot samples, based on the scoring function defined in the equation 2 (Section 3.1.2).

Stratified Clustering-based Sampling (SC): We first score each sample in the target domain using the scoring function defined in equation 2, and then set select a proportionately equal number of top-scoring samples from each sub-domain clusters, totaling to k few-shot samples (Section 3.1.2).

4.4.2 Access to Source Domain Data

We use 4 iterations to retrain the model sequentially using different concentrations of the target dataset [0.2, 0.4, 0.6, 1] in each iteration with the remaining concentration filled in by the source datatset.

Gradual Training + Random Sampling (G+R): We train the model iteratively using a total of k few-shot samples consisting of different proportions (in each iteration) of samples selected uniformly at random from both the target domain as well as the source domain (Section 3.2.1).

Gradual Training + Stratified Clustering-based Sampling (G+SC): We train the model iteratively using a total of k few-shot samples consisting of different proportions (in each iteration) of samples selected from source as well as target domain. In this setting, the samples from the source domain are selected uniformly at random, from the target domain selected using stratified clustering-based sampling (Section 3.2.2).

5 Results

Sampling strategy is important when only target domain data is available We observe over in Ta-

	Source Data Available	KBIR				RoBERTa			
		5	10	50	100	5	10	50	100
OpenKP Dataset									
MANNER	No	$1.27_{0.49}$	$5.56_{3.15}$	$16.63_{1.22}$	$19.35_{1.98}$	$1.59_{0.58}$	$2.87_{2.39}$	$15.33_{2.29}$	$17.33_{1.91}$
R	No	$0.03_{0.01}$	$0.80_{0.05}$	$1.38_{0.01}$	$1.36_{0.02}$	$0.33_{0.28}$	$1.14_{0.24}$	$6.34_{5.65}$	$7.80_{7.28}$
C	No	$0.00_{0.00}$	$0.00_{0.00}$	$4.13_{6.31}$	$13.60_{9.13}$	$0.92_{0.66}$	$1.24_{0.78}$	$10.43_{4.93}$	$19.73_{1.01}$
SC	No	$0.65_{0.74}$	$0.40_{0.62}$	$19.19_{3.73}$	$27.71_{1.99}$	$0.13_{0.25}$	$0.00_{0.01}$	$24.96_{3.77}$	$27.78_{3.94}$
MANNER	Yes	$2.92_{0.90}$	$5.01_{4.38}$	$11.48_{2.04}$	$16.81_{0.57}$	$1.02_{0.89}$	$1.48_{2.55}$	$11.80_{0.67}$	$14.74_{0.66}$
G + R	Yes	$2.49_{2.75}$	$11.91_{7.82}$	$29.35_{2.62}$	$31.65_{1.64}$	$1.46_{0.29}$	$6.13_{1.97}$	$27.24_{1.07}$	$27.89_{1.62}$
G + SC	Yes	$8.01_{4.63}$	$\mathbf{16.78_{0.88}}$	$\mathbf{31.95_{1.29}}$	$\mathbf{33.78_{0.81}}$	$\mathbf{8.42_{0.72}}$	$16.75_{0.96}$	$29.58_{0.76}$	$30.96_{0.93}$
StackEx Dataset									
	Source Data Available	KBIR				RoBERTa			
		5	10	50	100	5	10	50	100
MANNER	No	$2.05_{0.57}$	$1.34_{0.26}$	$12.42_{2.42}$	$17.10_{3.71}$	$2.72_{0.55}$	$0.24_{0.20}$	$0.01_{0.01}$	$4.67_{6.29}$
R	No	$0.00_{0.00}$	$0.64_{0.09}$	$10.11_{8.88}$	$2.47_{0.00}$	$0.00_{0.00}$	$0.00_{0.00}$	$14.41_{9.09}$	$29.91_{2.01}$
C	No	$0.00_{0.00}$	$0.00_{0.00}$	$4.64_{1.03}$	$14.96_{9.09}$	$0.00_{0.00}$	$0.00_{0.00}$	$6.84_{4.46}$	$16.09_{5.24}$
SC	No	$0.32_{0.60}$	$0.18_{0.36}$	$33.96_{2.42}$	$37.67_{0.85}$	$0.12_{0.14}$	$0.01_{0.02}$	$32.28_{1.60}$	$35.54_{2.13}$
MANNER	Yes	$3.92_{0.92}$	$1.24_{0.42}$	$4.18_{3.79}$	$15.94_{1.39}$	$3.25_{2.28}$	$1.10_{0.95}$	$7.13_{0.58}$	$9.23_{6.27}$
G + R	Yes	$9.93_{9.96}$	$\mathbf{23.98_{11.85}}$	$34.97_{1.82}$	$40.59_{1.04}$	$3.47_{1.42}$	$15.52_{1.74}$	$33.15_{1.32}$	$39.51_{0.53}$
G + SC	Yes	$\mathbf{14.08_{5.79}}$	$19.47_{1.68}$	$\mathbf{38.46_{0.71}}$	$\mathbf{42.11_{0.92}}$	$12.91_{8.16}$	$20.53_{2.19}$	$36.63_{1.31}$	$39.06_{1.51}$

Table 3: Few-shot fine-tuning exact match F1-score performances for different number of exemplars. Here we bold the highest F1-scores for all values of k. The values are averaged over 4 seed settings with variance as subscript.

	5	10	50	100
G + SC	$\mathbf{8.01_{4.63}}$	$\mathbf{16.78_{0.88}}$	$\mathbf{31.95_{1.29}}$	$\mathbf{33.78_{0.81}}$
G + SC-J	$2.05_{1.46}$	$12.69_{1.64}$	$26.19_{0.82}$	$31.03_{1.06}$
G + SC-C	$0.28_{0.37}$	$7.90_{2.29}$	$26.45_{1.72}$	$30.05_{0.87}$
G + SC-ChatGPT	$3.25_{2.30}$	$3.78_{1.08}$	$20.74_{4.10}$	$20.74_{3.68}$

Table 4: Exact match F1-score performance of KBIR model on the OpenKP test set for the G+SC variants.

ble 3, both the datasets that leverage the clustering-based heuristics result in significant boosts in performance (up to +26.35 F1). We see the gap between Random performance increase with number of exemplars as the model is able to train on more diverse and representative data. We observe that at times RoBERTa seems to outperform (up to +6.3 F1) KBIR and this is expected since there is no domain adaptation that KBIR can successfully exploit and RoBERTa is trained on more diverse pre-training data.

Access to source domain labelled data enhances sampling strategy impacts We observe in Table 3, over both the datasets and models that leveraging clustering over Random sampling when using Gradual training (**G+**) consistently results in statistically significant differences. As hypothesized, we find that access to labelled source data allows the KBIR model to learn from the few-shot samples more effectively (up to +3.05 F1) than RoBERTa. Further, it also outperforms (up to +12.76 F1) the strategy with only access to target data.

Reasonable performance for a fraction of the data We observe in Table 2 and 3 that we are able to match up to 69.75% of OpenKP and up to 67.70% of StackEx full fine-tuning performance while using only 0.45% of the data (K=100). This is significant as we evaluate on sufficiently large test sets as described in Section 4.1. Further, we are able to outperform PatternRank and MANNER consistently which Random sampling cannot. Interestingly, MANNER regresses performance when source data is included as it expects significant source data in a source-training step which is unavailable at the same scale and thus serves to confuse it. We observe no performance regression when also evaluated in source domain on KP20k (Meng et al., 2017) in Section 6.

Stratified clustering-based samplings leads to relatively higher inter-sub-domain sample relevance, but compromises on intra-sub-domain semantic diversity Semantic similarity between two document embeddings increases as the cosine distance between them decreases. Although the few-shot samples using **SC** have higher diversity in terms of the number of samples from each sub-domain compared to **R** (Fig. 8 in App. D), cosine distance variation from the corresponding sub-domain centers is relatively lower (lower intra-sub-domain semantic diversity) whereas the mean cosine distance is higher (Fig. 9 in App. D), making them semantically closer, relevant to other sub-domains (higher inter-sub-domain relevance), and relatively distant from the corresponding sub-domain center (sub-domain topic representation), relative to **R**.

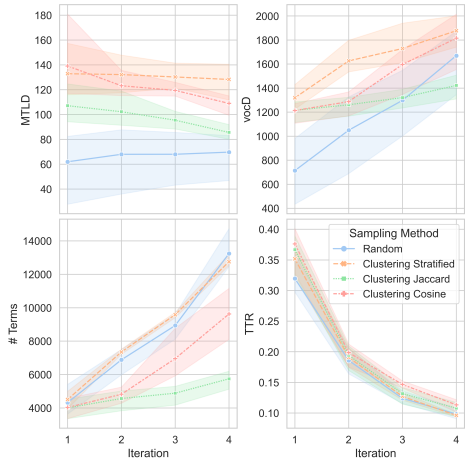


Figure 2: Lexical diversity metrics values per iteration for OpenKP samples in the gradual training setup.

5.1 Case Study: Optimizing G+SC

We observe from Table 3 that both **SC** and **G+SC** lead to significantly better performance than their Random counterparts in **R** and **G+R**. From the few-shot sample analysis in Fig. 9 & Fig. 8 in App. D, we observe that over various seeds, the few-shot samples selected using **R** not only belong to a diverse set of clusters from many sub-domains with disproportionate contributions similar to **SC** but are exhibit relatively varying cosine distance from their corresponding sub-domain center embeddings, w.r.t **SC** as seen in Std. deviation of **R** & **SC** in Fig. 9 from App. D). **SC** exhibits samples that are relatively distant from their corresponding sub-domain centers resulting in relatively higher relevance (selected based on equation 2) to all the other sub-domains. Thus the samples are relatively farther in cosine distance from their corresponding sub-domain center embeddings with low variance.

We explore if improving the low intra-sub-domain semantic diversity in **G+SC** while maintaining high inter-sub-domain diversity results in better performance. We propose the three variants of **G+SC** which enforce higher intra-sub-domain semantic diversity using greedy heuristics in the stratified sampling approach from the target domain data. For each setup we start with the top-scoring samples in each of the sub-domain cluster.

G+SC with Greedy Jaccard Similarity Selection (G+SC-J): The subsequent set samples in the corresponding sub-domains are selected, that has the least token-level Jaccard similarity with the previously selected samples in the corresponding sub-domains till a total of k samples are selected from the target domain.

G+SC with Greedy Cosine Similarity Selection (G+SC-C): The subsequent set samples in the corresponding sub-domains are selected, that has the least sentence-level cosine similarity with sentence embeddings of the previously selected samples in the corresponding sub-domains till a total of k samples are selected from the target domain.

G+SC with Greedy ChatGPT prompting (G+SC-ChatGPT): We prompt (App. E) ChatGPT (OpenAI, 2023) to generate a diverse set of keyphrase extraction labelled data similar to these top-scoring samples for the corresponding sub-domains.

In all the above-mentioned variants the random sampling from the source data and the gradual training approach is the same as that of **G+R**.

The quality of samples is dependent on the trade-off between their degree of relevance to other sub-domains (top-scoring samples) and their intra-sub-domain semantic diversity We report the performance of these variants of **G+SC** on the experiments described in Section 4.4 in Table 4. From Fig. 9 in App. D, we observe that although the samples selected in **G+SC-J** and **G+SC-C** have relatively higher diversity in terms of cosine distance from the corresponding sub-domain cluster centers resulting in higher intra-sub-domain semantic diversity. However, performance of these variants across both the datasets are poor compared to **G+SC**. We believe the primary reason for this is the steep decrease in the number of samples distant from the sub-domain cluster center due to such strong heuristics resulting in a decrease in the relevance of these samples to all the sub-domain topics, and the overall sample quality (representativeness).

In order to further investigate intra-sub-domain semantic diversity in the gradual training setup, we use textual lexical diversity metrics (Shen, 2021) such as MTLD (Measure of Textual Lexical Diversity), vocD (Vocab Density), the number of terms introduced, and TTR (Term Token Ratio) to analyze textual lexical diversity over the iterations of all the above mentioned gradual training-based approaches as shown in Fig. 2. The higher the values of these metrics the higher the textual lexical diversity (McCarthy and Jarvis, 2010).

Higher rate of increase of target domain sample diversity over the iterations result in bet-

Configuration	K=5	K=10	K=50	K=100	Full Fine-Tune
KBIR	-	-	-	-	33.57
KBIR-OpenKP as G+SC	$6.39_{0.55}$	$11.19_{8.75}$	$24.33_{6.16}$	$25.87_{1.55}$	-
KBIR-StackEx as G+SC	$4.54_{2.18}$	$7.53_{4.89}$	$22.40_{5.35}$	$23.23_{2.17}$	-
RoBERTa	-	-	-	-	33.45
RoBERTa-OpenKP as G+SC	$3.85_{2.37}$	$13.14_{2.81}$	$23.75_{0.58}$	$24.05_{1.21}$	-
RoBERTa-StackEx as G+SC	$5.49_{11.42}$	$8.71_{4.87}$	$20.69_{8.32}$	$20.33_{9.72}$	-

Table 5: Cross-Domain Generalizability of Model evaluated on the Scientific Domain KP20k dataset

Configuration	K=100	K=250	K=500
R	$1.36_{0.02}$	$15.24_{0.71}$	$25.06_{7.50}$
SC	$27.71_{1.99}$	$31.03_{0.82}$	$37.08_{4.24}$

Table 6: Exploring the value of K for Data Saturation of the Stratified Clustering compared to Random

ter domain adaption From Fig. 2 and Table 3 we observe that the performance of the model in the gradual training setting depends on both, the diversity (higher MTLD, vocD, # of Terms with lower TTR) in each iteration as well as the rate of increase of diversity in subsequent interactions. Although **G+SC-J** and **G+SC-C** maintain higher overall MTLD & vocD (initial iterations) throughout the iterations relative to **G+R**, **G+R** and **G+SC** outperforms them as they have a higher rate of increase in diversity over the iterations, despite **G+R** having relatively lower diversity in each iteration.

6 Cross-domain Generalization

We evaluate model performance on the source domain data to analyze whether the model is able to generalize across domains and not catastrophically forget the source domain. We do so by evaluating against the KP20k (Meng et al., 2017) corpus which consists of scientific articles as seen in Table 5.

We observe that both the model despite being trained in a cross-domain setting remain fairly competitive against a fully-fine tuned model on the source domain data. Demonstrating that our proposed framework does not degrade the model's generalization performance.

7 Data Saturation

We also explored if scaling up the value of K allows us identify the point at which Random (R) outperforms our proposed methods in Table 6. We observe performance of R at K=500 is similar to SC at K=100, suggesting that it might require significantly more data and hypothesizing this data saturation number may be well into the thousands.

8 Conclusion & Future Work

In this work, we explored the generalization capabilities of the KBIR for keyphrase extraction across different domains using few-shot fine-tuning. We proposed a novel Clustering-based few-shot sampling approach that uses sub-domain information as topics for extracting high-quality few-shot samples in a cross-domain setting, which leads to a significant gain in performance compared to randomly sampling few-shot samples. We also demonstrated that the gradual training regime in a few-shot setting performs better than its counterparts. We conducted a case study of similarity metrics and prompts that could enhance clustering-based sampling to quantify improvements to the Gradual training regime. Further exploration is required on heuristics that could further improve data diversity and if these findings hold true for in-context learning settings for keyphrase generation.

9 Limitations

This project involves a huge set of experiments with multiple data seeds, model seeds, and KMeans clustering seeds. We had initially planned to conduct few-shot experiments for keyphrase generation as well but owing to limited time and compute power we later focused only on keyphrase extraction, that too only on two particular datasets and models. On the technical side, there is no comparable baseline for few-shot keyphrase extraction so we had to benchmark the baseline by Cross-Domain Few-Shot Fine-tuning Named Entity Recognition literature, which is also sequence tagging based. Further, we do not explore generalized domain adaptation techniques such as DAPT (Gururangan et al., 2020), as these require large amounts of data and compute resources, whereas our focus is to maximize performance when using minimal data and compute. For clustering, we chose k-means as it is a simple method and worked reasonably well for our

use case, however, other more sophisticated methods could help boost performance. Also, there are no labeled sub-topics of the documents in these keyphrase extraction datasets so it was a challenge to judge the quality of sub-topics after clustering. Further, the source domain experiments may be slightly biased towards KBIR as the source domain is scientific data, however, the results and trends still hold on the RoBERTa model albeit with a slightly worse performance which is expected and further strengthening our claims on the robustness of our proposed method. Lastly, while our experiments are most effective for low-resource domains we conduct experiments on simulations of these in high-resource domains, we do so primarily to test on a large number of high quality samples but further work is required to truly annotate low-resource domain data.

Given the rapid development of large-scale models, coupled with their inherent robust few-shot learning capabilities, it will be an interesting direction to use the proposed sampling strategy Large Language Models (LLMs) for improving the diversity in in-context examples. In our experiments, we restricted the model size to be same as the KBIR model (present SOTA for keyphrase extraction). In future it would be interesting to see how much downstream performance depends on the quality of few-shot samples as we scale the model size. Experimenting with much diverse datasets would further help to establish the generalisability of the proposed sampling approach.

10 Ethical Consideration

We didn't find any significant harm in applying fine-tuning on cross-domain few-shot training. The methods we explore are general-purpose methods for low-resource tasks and domain adaptation.

References

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*, WWW '19, page 2551–2557, New York, NY, USA. Association for Computing Machinery.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

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, Prague, Czech Republic. Association for Computational Linguistics.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Erion Çano and Ondřej Bojar. 2020. Two huge title and keyword generation corpora of research articles. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6663–6671, Marseille, France. European Language Resources Association.

Ciprian Chelba and Alex Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lo. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 285–292, Barcelona, Spain. Association for Computational Linguistics.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase generation with correlation constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4057–4066, Brussels, Belgium. Association for Computational Linguistics.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019. Title-guided encoding for keyphrase generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6268–6275.

Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. 2021. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 28–40, Online. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Jinyuan Fang, Xiaobin Wang, Zaiqiao Meng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. MANNER: A variational memory-augmented model for cross domain few-shot named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4261–4276, Toronto, Canada. Association for Computational Linguistics.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Decis. Support Syst.*, 27(1–2):81–104.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223.

Steve Jones and Mark S. Staveley. 1999. Phrasier: A system for interactive document retrieval using keyphrases. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 160–167, New York, NY, USA. Association for Computing Machinery.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden. Association for Computational Linguistics.

Mikalai Krapivin and Maurizio Marchese. 2009. Large dataset for keyphrase extraction.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2022. Learning rich representation of keyphrases from text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 891–906, Seattle, United States. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

Matej Martinc, Blaz Skrlj, and Senja Pollak. 2020. TNT-KID: transformer-based neural tagger for keyword identification. *CoRR*, abs/2003.09166.

Philip M. McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42:381–392.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. 2023. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation.

OpenAI. 2023. Chatgpt (june 1 version) [large language model].

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100, Vancouver, Canada. Association for Computational Linguistics.

Vahed Qazvinian, Dragomir R. Radev, and Arzucan Özgür. 2010. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 895–903, Beijing, China. Coling 2010 Organizing Committee.

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.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. *Automatic Keyword Extraction from Individual Documents*, chapter 1. John Wiley Sons, Ltd.

Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. In *Advances in Information Retrieval*, pages 328–335, Cham. Springer International Publishing.

Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, Partha Pratim Das, Samiran Chattopadhyay, and T. Y. S. S. Santosh. 2019. Enhancing access to scholarly publications with surrogate resources. *Scientometrics*, 121(2):1129–1164.

Timo Schick and Hinrich Schütze. 2020. Few-shot text generation with pattern-exploiting training. *CoRR*, abs/2012.11926.

Tim Schopf, Simon Klimek, and Florian Matthes. 2022. Patternrank: Leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR*, pages 243–248. INSTICC, SciTePress.

Lucas Shen. 2021. Measuring political media slant using text data.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Santosh Tokala, Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. 2020. SaSAKE: Syntax and semantics aware keyphrase extraction from research papers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5372–5383, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chengyu Wang, Minghui Qiu, Jun Huang, and Xiaofeng He. 2020. Meta fine-tuning neural language models for multi-domain text mining. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3094–3104, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. 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 5175–5184, Hong Kong, China. Association for Computational Linguistics.

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. 2021. Gradual fine-tuning for low-resource domain adaptation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 214–221, Kyiv, Ukraine. Association for Computational Linguistics.

Hai Ye and Lu Wang. 2018. Semi-supervised learning for neural keyphrase generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4142–4153, Brussels, Belgium. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating diverse keyphrases as a set. 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 4598–4608, Online. Association for Computational Linguistics.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One size does not fit all: Generating and evaluating variable number of keyphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7961–7975, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Yong Zhang, Yang Fang, and Xiao Weidong. 2017. Deep keyphrase generation with a convolutional sequence to sequence model. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, pages 1477–1485.

Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. 2004. World wide web site summarization. *Web Intelli. and Agent Sys.*, 2(1):39–53.

A Hyperparameters

We experimented with different numbers of few-shot samples (k), i.e., 5, 10, 50, 100. We specify the hyperparameters used to reproduce our experiments in Table 7. As KBIR was pre-trained on the OAGKx dataset, we used a uniformly sampled subset of 22k data points from 23 million **OAGKx** dataset for our **source domain**.

For gradual training, we use 4 iterations to retrain the model sequentially. Here we also use different concentrations of the target dataset, i.e., [0.2, 0,4, 0,6, 1] in each iteration. The first iteration consists k few-shot samples having a source-to-target domain ratio (K_1:K_2) of 80:20 respectively, the second iteration constitutes a 60:40 source-to-target split, and so on with the final iteration constituting only target domain samples. Samples from the previous iterations remain and only new samples are added to meet the appropriate ratios. We do so to avoid seeing more data points than the budget under the guise of new iterations.

	Full Fine-tune	Few-shot
Number of epochs	5	50
Train batch size	32	32
Inference batch size	128	128
Gradient Accumulation	1	1
Learning rate	1e-5	1e-5
Learning rate scheduler	LINEAR	LINEAR
Early stopping used	yes	yes
Early Patience	3	3
Logging Steps	100	10
Adam ϵ	1e-6	1e-6
Warmup-proportion	0.01	0.01
Warmup-decay	0.00	0.00
Data seeds	-	[42, 67]
KMeans seeds	-	[27, 55]
Model seeds	-	[53, 80]
Target domain concentrations		[0.2, 0.4, 0.6, 1]
Gradual training iterations		4
Max generation length	512	512
Sequence-tagging Tags	"B", "I", "O"	"B", "I", "O"
22k dataset subsampling seed	42	

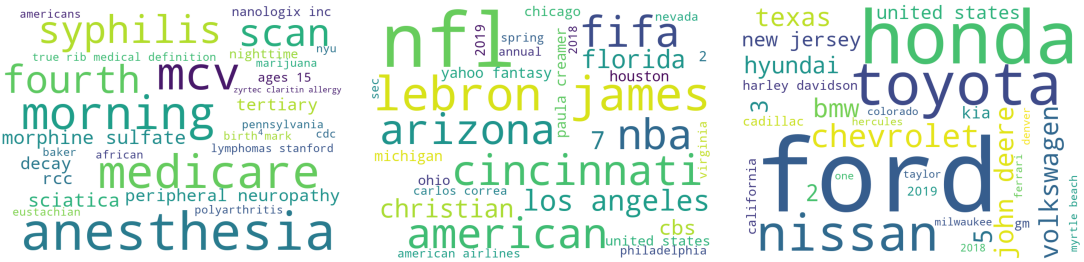
Table 7: Hyper-parameters for full fine-tuning & few-shot experiments.

We use 8 GeForce GTX 1080ti GPUs to run these experiments. Regarding training times, Roberta and KBIR models take nearly the same time for both full fine-tuning and gradual training on a particular dataset. Considering that we subsample 22k instances from both datasets, so full fine-tuning training takes 1 hr on average to train for a particular seed. On the other hand, few-shot training takes around 27 min on average across different seed values. In the case of gradual few-shot training, each seed takes little more than 1.5 hrs on average for 4 iterations for a particular k value.

B Cluster Analysis

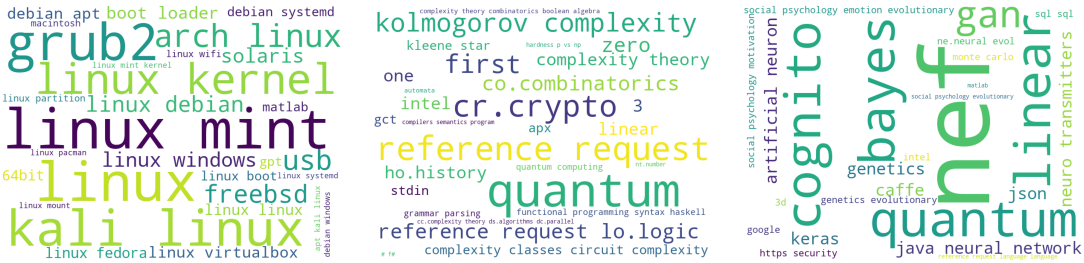
To generate the clusters in our proposed clustering-based sampling approaches, we used all-MiniLM-L6-v2[3] sentence transformer model for generating sentence embeddings of the documents, where the generated summaries were normalized. We used silhouette score analysis to identify an optimal number of clusters in each of the datasets and later investigated them with qualitative analysis using the word clouds generated from the cluster vocabulary. From silhouette score analysis we identified the optimal number of clusters in OpenKP as 15 (which

[3]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2



(a) Healthcare (b) Sports (c) Automobile

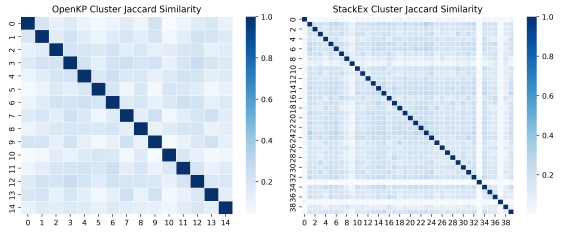
Figure 3: Wordclouds consisting of most frequent words belonging to three clusters in the **OpenKP** dataset, where the caption describes the corresponding sub-domain topics.



(a) Computer Science (Operating Systems) (b) Quantum Theory (c) AI/ML

Figure 4: Wordclouds consisting of most frequent words belonging to three clusters in the **StackEx** dataset, where the caption describes the corresponding sub-domain topics.

is also in line with (Xiong et al., 2019)) & 40 for the StackEx dataset. The silhouette score plots for OpenKP and StackEX are illustrated in Fig. 6. We further analyzed the quality of the generated clusters by investigating the inter-cluster similarity, which we expected to be low if the clusters are of good quality. Due to no access to the sub-domain labels in the above-mentioned datasets, we analyzed the inter-cluster similarity using Jaccard similarity between the clusters. Fig. 5 illustrates that on average the inter-cluster Jaccard similarity between all the combinations of clusters in both datasets was low, indicating less vocab similarity resulting in decent clustering. To get more insight into the vocabulary of these clusters, we also qualitatively analyzed the most common terms in these clusters. Fig. 3 & Fig. 4 show the word clouds for the most common terms in the OpenKP and StackEx datasets respectively, where we observe a clear distinction between the domains of these clusters. For example in Fig. 3, we can easily say by looking at the clusters (a), (b), and (c) consists of documents from Healthcare, Sports, and Automobile domains respectively, similarly in Fig. 4 clusters (a), (b), and (c) consists of documents from Computer Science (Operating Systems), Quantum Theory, and AI/ML domains respectively.

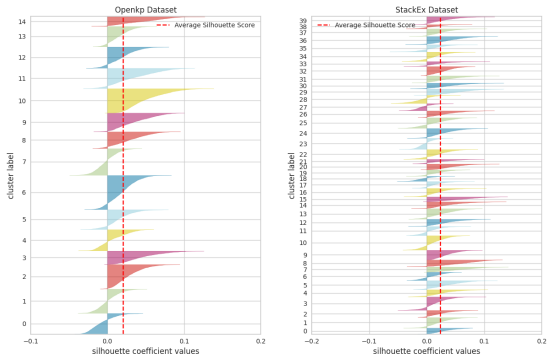


(a) OpenKP (b) StackEx

Figure 5: Inter-cluster Jaccard similarity between all the clusters in **OpenKP** and **StackEx** dataset.

C Evaluation Metric

In line with prior work (Sahrawat et al., 2020; Kulkarni et al., 2022), we report Exact Match F1 score as our primary metric using seqeval[4].



(a) OpenKP (b) StackEx

Figure 6: Sillouette plot for the optimal number (15 & 40) of KMeans clusters in **OpenKP** and **StackEx** dataset respectively.

D Few-shot Sample Analysis

For the few-shot cross-domain setting, we analyze and compare the quality of few-shot samples using the proposed sampling approaches and study their overall sub-domain cluster diversity, inter-sub-domain sample relevance, and intra-sub-domain semantic diversity. In this section, we dive deep into analyzing these metrics and how they relate to the overall performance of the model using different sampling approaches in a few-shot cross-domain setting.

D.1 Overall Sub-domain Cluster Diversity in Few-shot Samples

We analyze the sub-domain diversity in a set of samples by observing how uniform the distribution is for the number of selected few-shot samples contributed from each sub-domain cluster. The more uniform this distribution, the more diverse the set of samples is. If this distribution is skewed towards a particular small set of clusters, the majority of the few-shot samples are corresponding to those sub-domain clusters resulting in a decrease in overall sub-domain cluster diversity.

In a few-shot cross-domain setting, the higher the overall sub-domain cluster diversity, the higher the coverage over all the sub-domains given just a small set of samples, resulting in higher representativeness of the corresponding samples w.r.t to the target domain data. From Fig. 7 & Fig. 8, we observe that in the case of the samples generated using **R** & **C**, over all the seed settings, the distribution of the number of selected few-shot samples contributed from each sub-domain cluster is slightly skewed to a few set of clusters, whereas in the case of **SC**, it is almost uniform as all the clusters contribute the approximately same number of samples (Section 3.1.2) resulting in better overall sub-domain cluster diversity over **R** and **C**, leading to performance improvements in **SC** over **C** and **R** in Table 3.

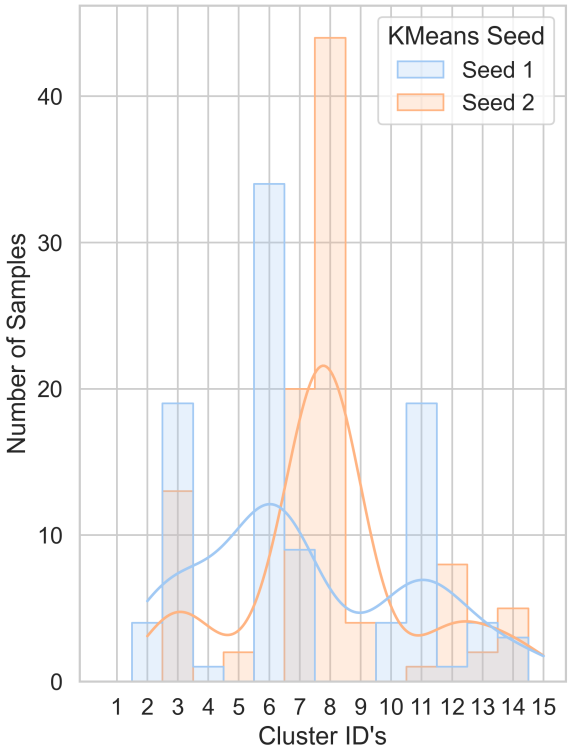


Figure 7: Distributions for the number of few-shot samples (total 100 samples) per cluster selected using the original cluster-based sampling approach (**C**) from the **OpenKP** dataset, for all the KMeans seeds.

D.2 Inter-sub-domain Few-shot Sample Relevance

For clustering-based sampling approaches explained in Section 3.1.2, we use equation 1 & 2 to score each sample based on their relevance with the other sub-domain cluster centers and pick the top scoring k samples as the few-shot samples. We illustrate the cosine distance distribution of such samples chosen in **SC** & **C** from their corresponding sub-domain cluster centers in Figure 9 over different KMeans seed settings (cosine distance calculated using the document embedding with the corresponding sub-domain cluster embedding). From the distribution plots for **SC** & **C**, we observe that on average these samples are distant from their

[4] https://huggingface.co/spaces/evaluate-metric/seqeval

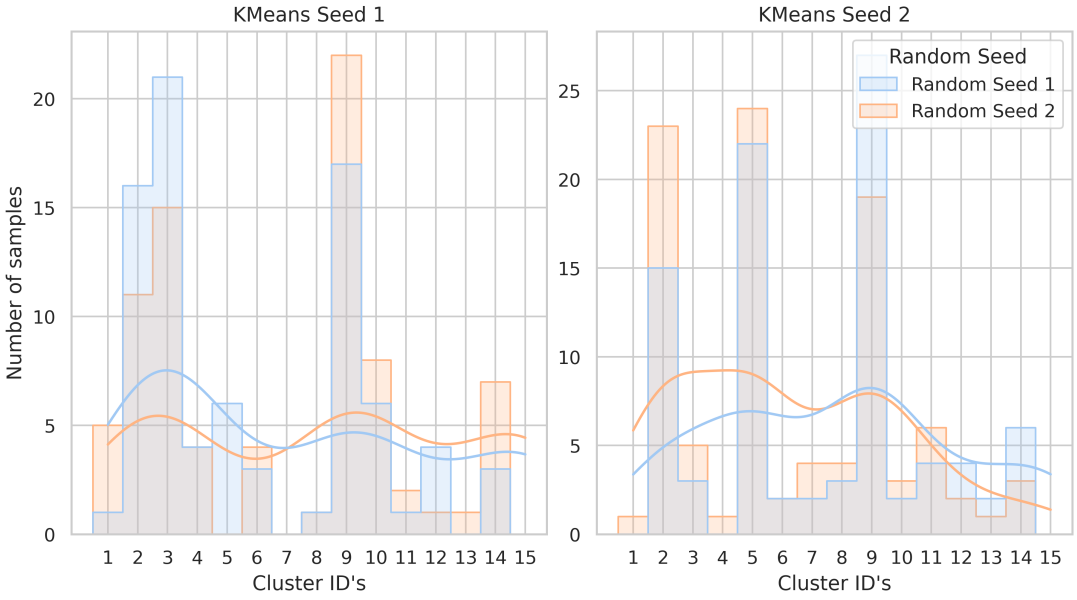


Figure 8: Distributions for the number of few-shot samples (total 100 samples) per cluster selected using the random sampling approach (**R**) from the **OpenKP** dataset, for all the random data seed. Here random sampling few-shot samples are assigned cluster ids using two sets of clusters based on two different KMeans seeds.

corresponding sub-domain cluster centers, while from the scoring function in equation 1 we know that these samples also have high relevance to other sub-domains (Section 3.1.2). So it is safe to conclude that the more distant the samples are from the corresponding cluster centers (in the direction of increased scoring function value), the more relevant they are to the other sub-domains, and vice versa. In the case of **R**, this cosine distance distribution is slightly right-skewed indicating low inter-sub-domain relevance resulting in poor performance compared to **SC** & **C**, where the samples have higher inter-sub-domain relevance inducing easier domain adaption (Section 3.1.2).

D.3 Intra-sub-domain Few-shot Sample Semantic Diversity

While the samples selected using **C** & **SC** are on average distant from their corresponding center (in the direction of increased scoring function value) resulting in rsamples with high relevance to other subdomains, the standard deviation of this distance is relatively smaller compared to the samples selected using **R**. As these cosine distances are calculated using embeddings from the Sentence Transformer, a smaller standard deviation of the cosine distance from the corresponding sub-domain clusters indicates higher semantical similarity, and vice versa. From Fig. 9, we observe that since **C** & **SC** have a smaller standard deviation in the corresponding cosine distance distributions compared to **R** indicates higher semantical similarity, suggesting lower intra-sub-domain few-shot sample semantic diversity.

D.4 Variants & Trade-off

From the discussion in Appendix D.1,D.2, and D.3, we conclude that while the samples selected using **C** & **SC** have high overall sub-domain cluster diversity and high inter-sub-domain relevance, they lack in intra-sub-domain semantic diversity. In order to improve upon the intra-sub-domain semantic diversity, we proposed **G+SC-J**, **G+SC-C**, and **G+SC-ChatGPT** that use greedy heuristic-based sample selection methods (Section 5.1) for increasing intra-sub-domain semantic diversity. From Fig. 9, we observe that these variants indeed increase intra-sub-domain semantical diversity, but while compromising on the inter-sub-domain relevance as the cosine distance distribution shifts toward the left indicating samples with lower relevance to other domains were selected (as explained in App. D.2). From Table 4, we also observe that although these variations generate samples with higher intra-sub-domain semantic diversity, they still end up performing poorly compared to **G+SC** as they also compromise on the relevance factor and the overall representativeness.

Summary of our findings from Appendix D.1,D.2,D.3, and D.4:

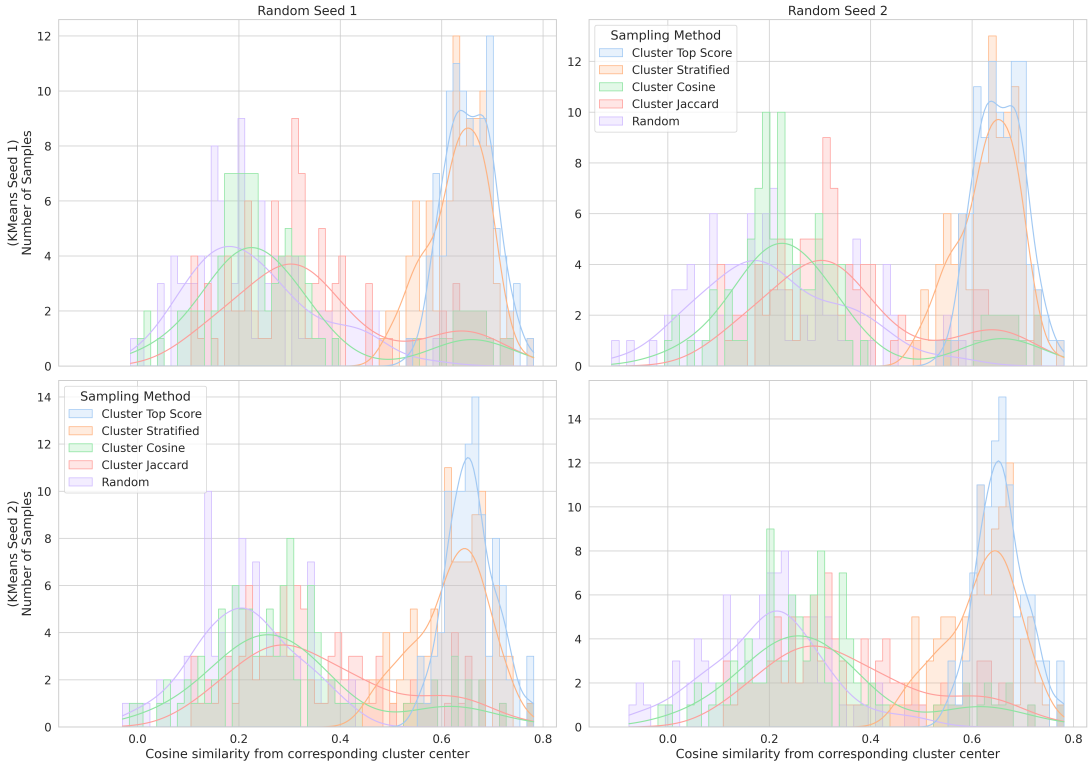


Figure 9: Distributions for the number of target domain few-shot samples (total 100 samples) selected from the **OpenKP** dataset vs their cosine similarity distances from the cluster centers of their corresponding cluster, for all the KMeans & random data seeds. Different colors represent different sampling strategies.

- Using clustering-based sampling approaches increases inter-sub-domain few-shot sample relevance while adding stratification in the sample selection further improves overall sub-domain cluster diversity.
- Higher inter-sub-domain sample relevance leads to lower intra-sub-domain semantic diversity.
- The overall performance of a model depends on the trade-off between the overall sub-domain cluster diversity, inter-sub-domain sample relevance, and intra-sub-domain semantic diversity in the samples selected using the sampling approach.

E ChatGPT Prompting

For few-shot gradual training, we also evaluated using samples generated by ChatGPT. For each dataset - OpenKP and StackEx, we used the top-scoring samples from the clusters as examples to ChatGPT API and asked it to generate 10 input-output examples for keyphrase extraction similar to the top-scoring sample in the cluster. Here is one example of prompt: 'I want to be able to generate data points to train a keyphrase extraction model. Here is a sample. document: 1 27 Overview Amenities Reviews Map Availability Lovely Remodeled Studio W Fireplace No cleaning Fee Park City UT USA Condo 394 sq ft Sleeps 4 Bedrooms Studio Bathrooms.....Our building has a bus stop right out the front door to the free Park City bus service with access to Main Street all ski areas outlet malls theaters shopping and restaurants Photos Treelined street A bus stop is right in front of the building Availability. keyphrases: lovely remodeled studio, home. Can you generate 10 similar data points in the domains similar to samples?'

F Sampling Algorithm

Algorithm 1 Random Sampling Algorithms

Source Dataset: D_s:$\{x_1^s, ..., x_M^s\}$
Target Dataset: D_t:$\{x_1^t, ..., x_N^t\}$
 # Few-shot Samples: k
 # Gradual Iterations: I
Pre-trained Model: π
Fine-tuned Model: π^*
Uniform Sampling Function: U: $D \rightarrow D^{"}$, where $||D^{"}|| = k$
 # Few-shot Source Domain Samples at i^{th} **Iteration:** k_1^i
 # Few-shot Target Domain Samples at i^{th} **Iteration:** k_2^i

Function `Rsample(`D,k`)`**:**
  `/* Random Sampling (R)` `*/`
  $X^* : \{x_i^*, ..., x_k^*\} \leftarrow U(D, k)$ ▷ $U(D, k)$ samples k documents from D uniformly at random
  return X_*

w/o Gradual Training
$X_* \leftarrow$`Rsample(`D_t,k`)` ▷ Few-shot samples
$\pi^* \leftarrow \pi(X^*)$ ▷ Fine-tune π

with Gradual Training
for $i = 1$ **to** I **do**
  $X^*_{source} : \{x_1^*, ..., x_{k_1^i}^*\} \leftarrow$`Rsample(`$D_s$,$k_1^i$`)`
  $X^*_{target} : \{x_1^*, ..., x_{k_2^i}^*\} \leftarrow$`Rsample(`$D_t$,$k_2^i$`)`
  $X^* \leftarrow X^*_{source} + X^*_{target}$ ▷ Few-shot samples
  $\pi^* \leftarrow \pi(X^*)$ ▷ Fine-tune π
  $\pi \leftarrow \pi^*$ ▷ Update π weights with π^* weights
end

Algorithm 2 Clustering-based Sampling Algorithms

Source Dataset: D_s:$\{x_1^s, ..., x_M^s\}$
Target Dataset: D_t:$\{x_1^t, ..., x_N^t\}$
 # Few-shot Samples: k
 # Gradual Iterations: I
Pre-trained Model: π
Fine-tuned Model: π^*
Sentence Transformer Embedding Model: M^{st}
 # Few-shot Source Domain Samples at i^{th} Iteration: k_1^i
 # Few-shot Target Domain Samples at i^{th} Iteration: k_2^i

Function `Rsample`(D,k)**:**
  `/* Random Sampling (R)` `*/`
  $X^* : \{x_i^*, ..., x_k^*\} \leftarrow U(D,k)$ ▷ $U(D,k)$ samples k documents from D uniformly at random
  return X_*
Function `Csample`(D,k)**:**
  `/* Clustering-based Sampling (C)` `*/`
  $E_t : \{e_1^x, ..., e_{\|D\|}^x\} \leftarrow M^{st}(\{x_1^t, ..., x_{\|D\|}^t\}); x_i^t \in D$ ▷ Sentence Embedding Generation
  $C : \{C_1, ..., C_c\} \leftarrow$ **KMeans**(E_t) ▷ Document Clustering
  for $i = 1$ **to** c **do**
    $e_i^C \leftarrow \frac{\sum_{j=1}^{\|C_i\|} e_j^x}{\|C_i\|}$; where $e_j^x \leftarrow M^{st}(x_j^t), \forall x_j^t \in C_i$ ▷ Sub-domain Embedding Generation
  end
  $E_C \leftarrow \{e_1^C, ..., e_c^C\}$
  for $i = 1$ **to** $\|D\|$ **do**
    $s_i = \frac{\sum_{j=1}^{c} \delta(e_j^C, e_i^x)}{c}$ ▷ Cosine Similarity Score (δ) between document embedding and sub-domain embeddings
  end
  $S \leftarrow \{s_1, ..., s_{\|D\|}\}$
  $X_* = \{x_1^*, ..., x_k^*\} = \arg topk_{x_i^t \in D}(S)$
  return X_*
w/o Gradual Training:
$X_* \leftarrow$`Csample`(D_t,k) ▷ Few-shot samples
$\pi^* \leftarrow \pi(X^*)$ ▷ Fine-tune π

with Gradual Training:
for $i = 1$ **to** I **do**
  $X^*_{source} : \{x_1^*, ..., x_{k_1^i}^*\} \leftarrow$`Rsample`$(D_s,k_1^i)$
  $X^*_{target} : \{x_1^*, ..., x_{k_2^i}^*\} \leftarrow$`Csample`$(D_t,k_2^i)$
  $X^* \leftarrow X^*_{source} + X^*_{target}$ ▷ Few-shot samples
  $\pi^* \leftarrow \pi(X^*)$ ▷ Fine-tune π
  $\pi \leftarrow \pi^*$ ▷ Update π weights with π^* weights
end