TIMS: A Novel Approach for Incrementally Few-Shot Text Instance Selection via Model Similarity

1st Tianjie Ju 3rd Gongshen Liu*
School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University,
Shanghai, China
{jometeorie, lgshen}@sjtu.edu.cn

2nd Han Liao
School of Computer and Information Science
Southwest University
Chongqing, China
hanliao@email.swu.edu.cn

***Abstract*—Large-scale pre-trained models' demand for high-quality instances forces people to consider how to select instances for annotation with limited resources. Nonetheless, little attention has been paid to the scenario where the number of instances that ultimately need to be annotated is agnostic. Meanwhile, the anisotropy of the sentence vector output by pre-trained models makes it hard to represent the instance itself well. Faced with the two challenges, we propose an incrementally few-shot instance selection approach (TIMS) based on model similarity and outlier detection, which suits the starting step of active learning well and serves as a better benchmark for few-shot learning. Specifically, TIMS determines the representative candidate set by calculating the similarity between changes in model parameters caused by each instance and by the full dataset. Meanwhile, Isolation Forest is adopted to select instances from the candidate set for annotation, which prevents selected instances from being too similar. Comprehensive experiments on WikiLingua & SQuAD show that TIMS outperforms other algorithms across almost every circumstance. It inspires us that the proper implementation of model similarity detection and outlier detection is of great help to select representative instances incrementally.**

***Index Terms*—Incrementally Few-Shot Instance Selection, Active Learning, Outlier Detection, Text Generation**

I. Introduction

Recently, transformer-based pretrained language models (T-PTLMs) have achieved great success in almost every Natural Language Processing (NLP) task [1]. Meanwhile, more attention is being paid to few-shot natural language processing [2], [3] to solve the over-reliance on annotated data for downstream tasks of T-PTLMs. However, **these previous works only focus on training task itself, ignoring the selection strategies of training instances** [4], which leads to a large amount of manpower being wasted on unimportant instance annotating, especially on training tasks that are difficult to annotate, such as summarization and question answering

Although several works attempt to focus on few-shot text instance selection [4], there still remains many problems with these works. **Firstly, these approaches tend to use contextual embeddings as representations of training instances**, while studies have shown that the output layer of T-PTLMs tend to degenerate and occupy an anisotropic cone in the vector space [5], [6]. **Secondly, these approaches cannot select instances incrementally**, i.e. the selected instance size is predetermined and cannot be changed at will, otherwise it may cause severe performance degradation. **Finally, these approaches rely heavily on a well-trained downstream model**. Selection strategies in active learning utilize information from downstream models, such as loss, gradient, etc. [7], [8], which ignores the situation where a well-trained downstream model is absent.

In this paper, we focus on how to incrementally select few-shot text instances for training. Different from active learning (AL) [9], [10] and anomaly detection (AD) [11], [12], we introduce a new scenario as Incrementally Few-Shot Instance Selection (IFIS) (as shown in Fig. 1), in which instance size that ultimately needs to be annotated is agnostic to us. Few instances ($\leq$ 10) are continuously selected each time and referred to domain experts for annotation.

Following this scenario, an IFIS approach based on model similarity is proposed to fully exploit the value of pre-trained models (as shown in Fig. 2). With existing pre-trained models (such as BERT [13] and BART [14]), we compare the similarity between parameter changes generated by pre-training with a single instance and by pre-training with domain datasets [15], and use the similarity as the basis for selecting instances. In addition, to prevent the selected instances from being too similar, Isolation Forest [16] is adopted during each incremental selection process, which significantly increases the performance of IFIS in the initial stage.

Comprehensive experiments are conducted on document summarization [17] and reading comprehension [18]. Our proposed approach incrementally picks out more valuable instances on both tasks and achieves higher performance with lower variance.

Our contributions are as follows:

- We propose a novel approach for incrementally few-shot instance selection. It is especially helpful when instance size that ultimately needs to be annotated is agnostic to people.

*Corresponding author.

- We leverage parameter information of powerful pre-trained models to select typical instances which best reflect domain datasets instead of directly using anisotropic contextual embeddings.
- We provide a better benchmark for few-shot learning and a starting step before applying active learning [19] to select instances since it performs better and does not have large variance issue as found in random sampling.
- We provide a novel idea for outlier detection via model similarity detection to find instances with low correlation to the domain dataset.

II. Related Works

Active Learning: AL attempts to maximize a model's performance gain while annotating the fewest instances possible [10]. The main propose of AL is to design a query strategy to judge the value of each instance. According to different query strategies, existing AL approaches can be roughly divided into uncertainty sampling, Expected Model Change (EMC), Query-By-Committee (QBC) [8] and diversity sampling [20]. Besides diversity sampling, other sampling strategies require a well-trained downstream model in order to perform well [7]. Similarly, diversity sampling can be roughly divided into the following three approaches:

- Representative sampling: select some of the most representative instances, such as clustering methods [4] in different domains.
- Outlier-based sampling: select instances based on the degree of outliers between instances (such as Isolation Tree [16], [21])
- Real scene diversity sampling: fairly sample according to the diversity and distribution of the real scene.

Training Instance Selection: Several works also focus on the importance of training instance selection itself. In [22], The authors found that different sampling leads to a large variance of model performance, which makes it difficult to compare across different Deep Learning models. In [4], the authors proposed a training instance selection strategy for few-shot neural text generation based on K-means++ [23]. However, contextualized embeddings at the output layer of T-PTLMs (such as BERT [13]) tend to degenerate and occupy an anisotropic cone in the vector space [5]. In addition, K-means++ is not suitable for our training scenario as the final instance size is agnostic. Assuming that only one training instance can be selected for annotating at a time, it tends to select the most central instance (as shown in Fig. 3(e)), which makes selected instances difficult to represent the full datasets well.

In this paper, since we mainly consider the scenario where the final size is agnostic and the well-trained downstream model is absence when selecting instances. Previous works that do not satisfy these two conditions will not be considered.

III. Problem Formulation

As the training scenario shown in Fig. 1, we assume that the training set D_{train} and the test set D_{test} follow the same distribution $p_{full}(x)$. Given the full set of instances $X \sim p_{full}(x)$ in D_{train}, we want to incrementally select $k(k \leq 10)$ instances each time for annotating and finally get a representative subset $D_{final} \subset D_{train}$ which leads to the optimal performance after training. Depending on the downstream task, these instances X generally refer to unannotated structured data such as document summarization and question generation in NLP.

More specifically, to better select a representative instance x_i, we make full use of (i) the information of the parameters H (concatenated by parameters of multiple Transformer Layers [26]) in the pre-trained model M and (ii) the outlier between x_i and the annotated dataset $D_{annotated}$ to select an instance which is representative and contains more information that instances in $D_{annotated}$ do not have.

IV. Our Approach

In this section, we first introduce an instance selection approach based on parameter changes before and after pre-training, which is the main framework of our proposed TIMS. Then we introduce the outlier detection technology based on Isolation Forest, so that the selected instances can cover a larger sample space. The overall algorithm is introduced to the end.

A. Model Similarity Detection

We first study the change in model parameters [25] caused by training an instance x from downstream datasets without annotation. Formally, the change is as follow:

$$\Delta \hat{H}_x \overset{def}{=} \hat{H}_x - H, \tag{1}$$

where $\hat{H}_x$ is the parameter that minimizes the loss function of the instance x pre-trained on the pre-trained model M. Specifically, we construct a dataset D_x which contains s multiple copies (80 in this paper) of instance x. Each data in D_x is obtained by random masking of x. We use Masked LM [13] with a large learning rate ($lr = 0.001$) as our pre-training task and finally get the new parameters $\hat{H}_x$:

$$\hat{H}_x \overset{def}{=} argmin_{H\in\Theta} \frac{1}{s} \sum_{i=1}^{s} L(x^i_{copy}, H), \tag{2}$$

where L is the loss function for the pre-training task, x^i_{copy} is is the i-th copy of the instance x. Using the same pre-training task, we pre-train on the full domain dataset D and obtain the change in model parameters:

$$\Delta \hat{H}_{dataset} \overset{def}{=} \hat{H}_{dataset} - H, \tag{3}$$

$$\hat{H}_{dataset} \overset{def}{=} argmin_{H\in\Theta} \frac{1}{N} \sum_{i=1}^{N} L(x_i, H), \tag{4}$$

where $\hat{H}_{dataset}$ is denoted as model parameters pre-trained in D_{train}, N is denoted as the total number of instances in D_{train}.

The change in model parameters indicates the response of a well-pretrained model M to a new dataset. By comparing the response generated by D_x with the response generated by

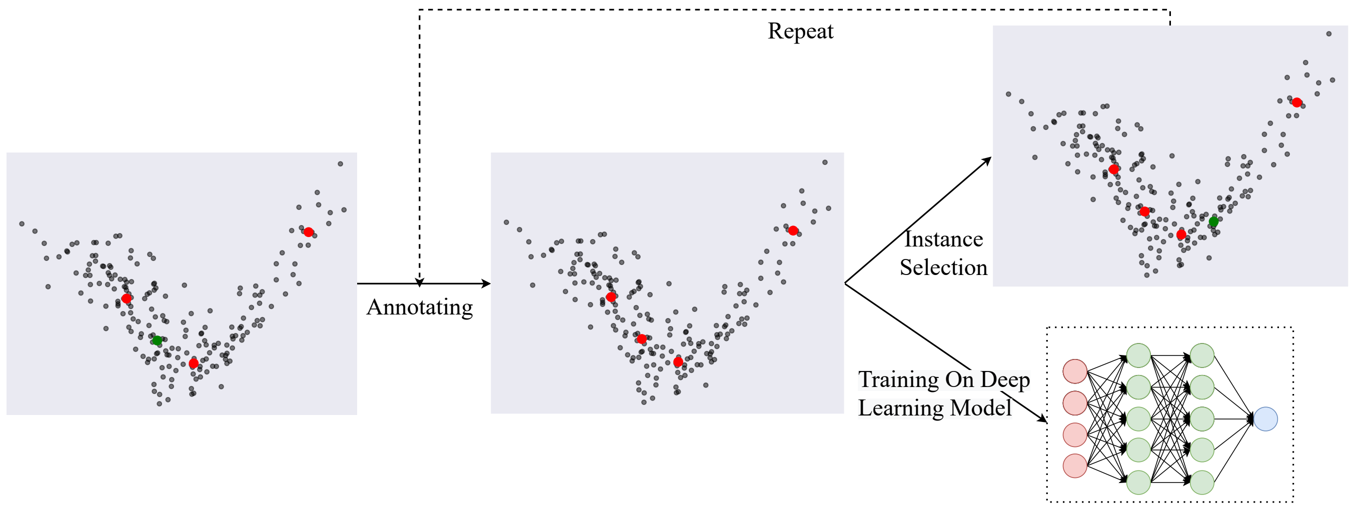


Fig. 1. Illustrative example of the training scenario (IFIS). Few Instances (green) are selected from training datasets (black) each step, and annotated by domain experts (red) for deep learning.

D_{train}, we can get the similarity of the instance x to the full dataset as $sim(\Delta\hat{H}_x, \Delta\hat{H}_{dataset})$. It's like a domain expert learning something new. A domain expert (M) learns new knowledge from two datasets (D_x and D_{train}) and produces variations ($\Delta\hat{H}_x$ and $\Delta\hat{H}_{dataset}$) to the existing knowledge system respectively. Assuming that $\Delta\hat{H}_x$ and $\Delta\hat{H}_{dataset}$ are similar, the knowledge gained by the domain expert from D_x will be very close to that obtained from D_{train}, indicating that instance x is a good representation of the full dataset. Therefore, we tend to select instances with higher similarity for annotation especially when there are few instances that can be annotated.

Furthermore, the response $\Delta\hat{H}$ can in turn be regarded as a vector in the parameter space Θ. During comparing, the response $\Delta\hat{H}_{dataset}$ is fixed. As for the response $\Delta\hat{H}_x$, it is not difficult for a large pre-trained model M to fit a single instance. Thus, the magnitude $|\Delta\hat{H}_x|$ of the vector is very small. The only significant difference is the direction in which $\Delta\hat{H}_x$ changes. This means that cosine function is a optional approach to measure how similar a single instance is to the dataset:

$$sim(\Delta\hat{H}_x, \Delta\hat{H}_{dataset}) \overset{def}{=} cos(\Delta\hat{H}_x, \Delta\hat{H}_{dataset}). \quad (5)$$

Finally, we assign a score to each instance according to the formulation in (5). In the process of each incremental selection, only the first k instances with the highest scores are selected for annotation. Since only angles of $\Delta\hat{H}_x$ and $\Delta\hat{H}_{dataset}$ are concerned, we do not need to spend much time (about 1-2 seconds per instance) to ensure that the pre-trained model M can accurately fit a single instance x during training. These scores make full use of the internal parameter information of a large pre-trained model and the distribution of downstream datasets, which make the selected instances more representative.

B. Outlier Detection

Our proposed TIMS based on model similarity detection have been able to select representative instances incrementally. However, it does not take into account the relationships between annotated instances, which may cause TIMS to select many extremely similar instances. Therefore, we incorporate an outlier detection approach based on Isolation Forest, so that the selected new instances are as different as possible from the annotated instances and cover a larger sample space.

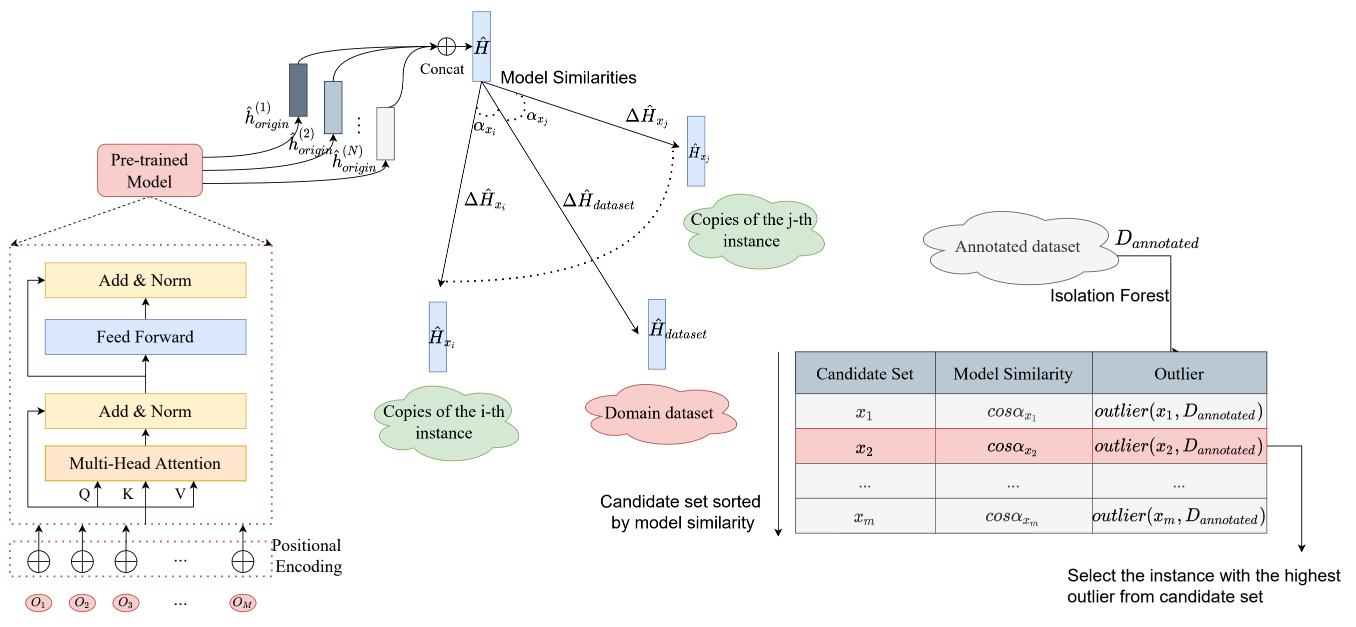
Isolation Forest is based on the idea that anomalous instances in a dataset tend to be more easily separated from other instances. Formally, it consists of a set of t iTrees denoted by $T = \{T_1, T_2, \cdots, T_t\}$ in a randomized manner [21]. Each iTree is constructed by recursively partitioning a sub-sample x' until all instances are isolated [16] using training data.

Considering the structure of iTrees is equivalent to that of Binary Search Tree (BST), the anomaly score *outlier* of an instance x can be defined as [16], [28]:

$$outlier(x, D) = 2^{\frac{-E(h(x))}{c(|D|)}}, \quad (6)$$

where D is the dataset, $E(h(x))$ is the average of path length of x, $c(|D|)$ is the average path length of unsuccessful searches in BST.

The instance with the largest outlier can be selected using Isolation Forest. However, it is always computationally intensive and is likely to select severely anomalous instances from unannotated datasets. Therefore, for each step, we first use model similarity detection (as described in the previous section) to select m instances with high similarity as a candidate set $D_{candidate}$. Next, we compute the $outlier(x, D_{annotated})$ of the instances in $D_{candidate}$ relative to $D_{annotated}$, respectively. Finally, the instance with the highest outlier in $D_{candidate}$ is selected as a new selected instance.



Candidate Set	Model Similarity	Outlier
x_1	$cos\alpha_{x_1}$	$outlier(x_1, D_{annotated})$
x_2	$cos\alpha_{x_2}$	$outlier(x_2, D_{annotated})$
...	...	...
x_m	$cos\alpha_{x_m}$	$outlier(x_m, D_{annotated})$

Fig. 2. The overall architecture of TIMS. For each step, we first compare the cosine similarity between parameter changes caused by pre-training with a single instance and by pre-training with domain datasets to obtain the candidate set. Then we compute the outlier of each instance in the candidate set relative to the annotated dataset, and select the instance with the largest outlier for annotation.

This is based on a natural assumption: when D_{train} is large ($N \geq 10000$) and $D_{candidate}$ is small ($m \leq 100$), all instances in the candidate set have high model similarity, so that they are strongly representative. In this case, we can turn our focus to capturing information not contained in the selected dataset, i.e. causing the new instance to deviate from the distribution of other selected instances. This approach provides a great help for incrementally instance selection.

Algorithm 1 presents our process of incrementally few-shot text instance selection based on TIMS. After adding the step of outlier detection, TIMS can avoid selecting extremely similar instances in few-shot scenarios.

V. Experiments

A. Datasets

We experimented our proposed approach on the following two well-studied open datasets: **WikiLingua** [29], a large-scale, multilingual dataset for the evaluation of crosslingual abstractive summarization systems; **SQuAD** [30], a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles. For the WikiLingua corpus, we mainly test on the English text summarization data.

B. Experimental Setup

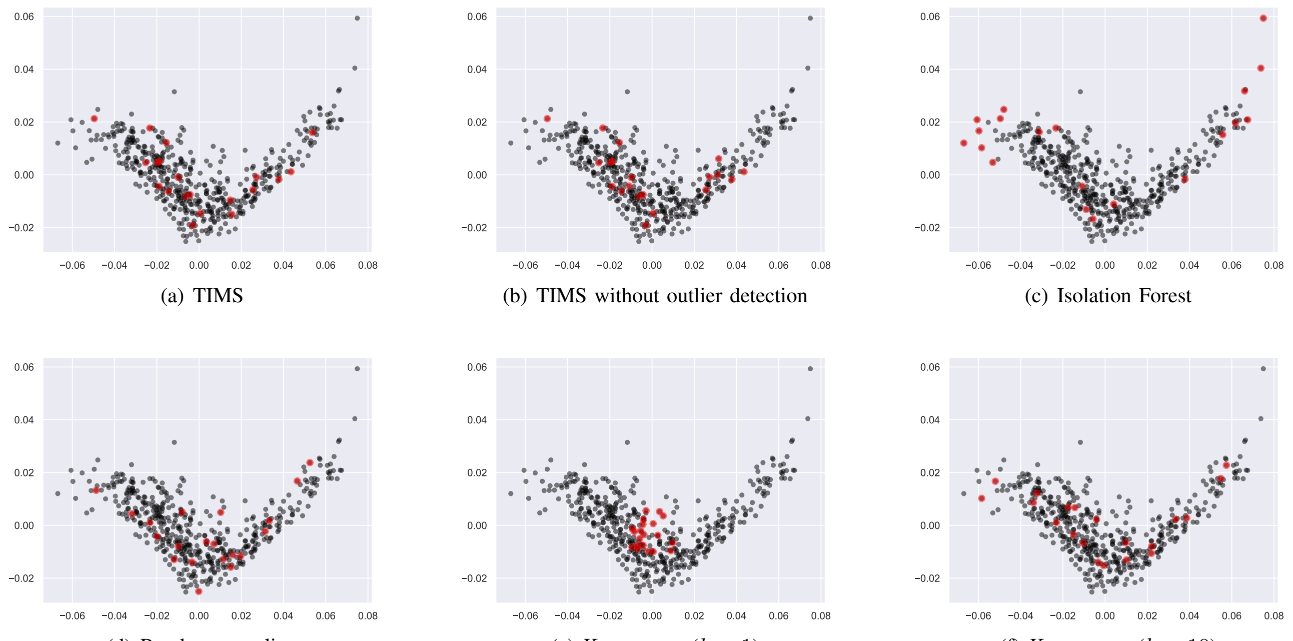
For both datasets, 30,000 instances are randomly sampled as the training set and 10,000 instances are randomly sampled as the test set. In each experiment, we incrementally select $k(k \leq 10)$ instances from the training set for annotation using

Algorithm 1 Incrementally Few-Shot Text Instance Selection based on TIMS

```
Input:
    Pre-trained model $M$ with parameters $H$, full dataset $D_{train}$, final size $K$, candidate size $m$;
Output:
    Subset $D_{final} \subset D_{train}$ for annotation
1:  $\hat{H}_{dataset} \leftarrow M.pretrain(D_{train})$
2:  for each $x \in D_{train}$ do
3:      $\hat{H}_x \leftarrow M.pretrain(x.copy())$
4:      $x.score \leftarrow cos(\hat{H}_x - H, \hat{H}_{dataset} - H)$
5:  end for
6:  $D_{final} \leftarrow \varnothing, D_{unannotated} \leftarrow D_{train}$
7:  while $D_{final}.length() < K$ do
8:      $D_{candidate} \leftarrow D_{unannotated}.select\_top\_score(m)$
9:      for each $x \in D_{candidate}$ do
10:         $x.outlier \leftarrow outlier(x, D_{final})$
11:     end for
12:     $x_{chosen} \leftarrow D_{candidate}.select\_top\_outlier()$
13:     $D_{unannotated} \leftarrow D_{unannotated} - x_{chosen}$
14:     $D_{final} \leftarrow D_{final} \cup x_{chosen}$
15: end while
```

different algorithms until the total number reaches $K(K \leq 150)$.

Although several works in active learning provide ideas for instance selection as mentioned in section 2, most of them are not suitable for our training scenario, or rely heveaily on a well-trained downstream model. In this paper, we only



(a) TIMS (b) TIMS without outlier detection (c) Isolation Forest

(d) Random sampling (e) K-means++ ($k = 1$) (f) K-means++ ($k = 10$)

Fig. 3. PCA-based visualization results of incrementally selected instances of different algorithms. We randomly sample 500 instances (black dots) from the WikiLingua dataset and use each algorithm to incrementally select 20 instances (red dots) for annotation.

focus on instance selection itself, and do not apply downstream models in the selection process. Therefore, the following approaches are chosen as baselines:

- Random Sampling. Random sampling is the most commonly used algorithm. It samples from the training set one by one without any bias, and can reflect the characteristics of the dataset to a certain extent.
- K-means++ (k). Since the K-means++ based approach can only select a fixed number of instances at once, it is not suitable for our scenario where the final size is agnostic. We adjust it and denote it as K-means++ (k). The general idea is to first split the whole unannotated dataset into $k(k \leq 10)$ clusters, then select the instance which is nearest to the center from each cluster. We repeat the following steps until the number of selected instances reaches K.
- Isolation Forest. Isolation Forest is a commonly used algorithm in anomaly detection as described in section 3. We use this algorithm as a lower bound to examine the effect of simply selecting outliers for annotation.

In order to fairly show the results of different instance selection algorithms, identical training models and evaluation metrics are adopted on downstream tasks. Due to the small amount of data, only distilled pretrained models are adopted for downstream tasks. Specifically, for the WikiLingua dataset, we finetune on the open-sourced DistilBart-xsum [32], [33] with selected instances and use Rouge-L [34] as the evaluation metrics. For the SQuAD dataset, we finetune on the open-sourced DistilBert-base-uncased [13], [35] and use Exact Match (EM) as the evaluation metrics.

All architectures are implemented using PyTorch [36] and HuggingFace [37]. When pre-training on the full dataset, the learning rate and batch size are set to 0.0001 and 64 respectively. When pre-training on a single instance, the learning rate and batch size are set to 0.001 and 16 respectively. Meanwhile, we generate multiple replicas of the single instance using mask strategy in BERT.

C. Evaluation Results

Visualization of Selection Approaches To intuitively demonstrate the preference of various approaches for selecting instances, we randomly sample serveral instances from the WikiLingua dataset for testing and use Principal Component Analysis (PCA) [38] to reduce the dimension of instances (as shown in Fig. 3).

For random sampling (Fig. 3(d)), its selection result completely depends on the probability density of the instance $p_{full}(x)$. Therefore, when few instances are selected, it has large uncertainty: some selected instances are too close while some deviate severely from the full dataset. For K-means++ (Fig. 3(e)(f)), it relies heavily on the number of instances (k) selected each time. All instances it selects will be near the center of the k clusters. For Isolation Forest (Fig. 3(c)), it tends to select anomalous instances, which does not reflect the dataset itself well. For TIMS (Fig. 3(a)), it takes into account annotated instances while selecting representative instances to avoid instances being too similar. However, without outlier detection (Fig. 3(b)), TIMS may also face the situation where selected instances are similar between each other and cover smaller sample space.

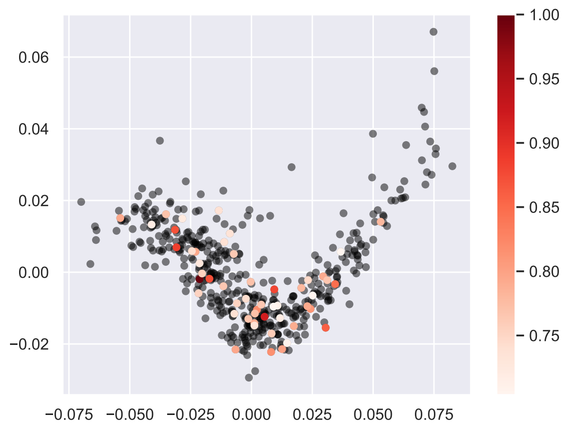


Fig. 4. Candidate set selected by TIMS. We randomly sample 500 instances (black dots) from the WikiLingua dataset and determine candidate set ($m = 50$) via model similarity. The shade of color indicates the score obtained in the model similarity detection stage.

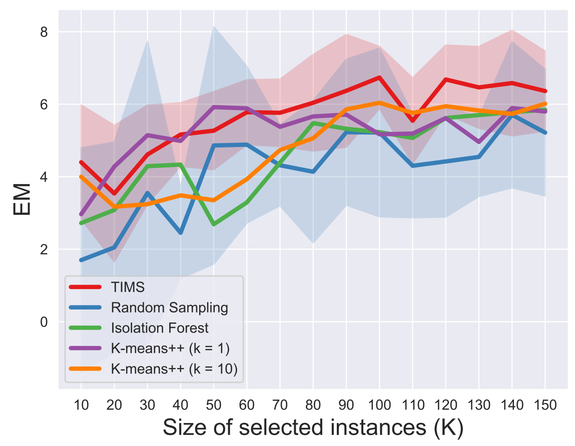


Fig. 5. The effect of different incrementally instance selection approaches on the SQuAD dataset. The figure shows only the variance of TIMS (red area) and Random Sampling (blue area).

Visualization of Candidate Set We show the candidate set computed by TIMS in Fig. 4. The shade of color represents the order of the instance in the candidate set. The candidate set ensures that instances selected by the second stage of TIMS (outlier detection) are strongly representative. However, Without the candidate set, TIMS will select many very close instances at the beginning (several dark instances are seem to be quite close).

Comparison of Selection Approaches In Table I, we compare the model performance based on different approaches. All results are averaged on 10 trials. As can be seen, TIMS outperforms the others in most cases. The performance of random sampling is consistently poor, and its large variance also implies that instance selection has a large impact on model performance.

Moreover, we conduct detailed experiments on the SQuAD dataset with 10 trials for each final size (K). In Fig. 5, we show the trends of different approaches as K increases. When K is small ($K < 50$), K-means++ ($k = 1$) performs slightly better. This is probably because it always selects the most central instances (as shown in Fig. 3(e)). In most cases, TIMS performs better among all approaches. Meanwhile, it is intuitive to see that the variance of TIMS is smaller than that of random sampling, which indicates that it can provide a more stable benchmark as instance selection for few-shot learning.

Outlier Detection What if we reverse the order of the instances according to the scores obtained by model similarity detection? In Table II, We select the five lowest-scoring instances from the WikiLingua dataset. As can be seen, four of these instances are too short to be annotated. Another instance, although longer in content, contains numerous repetitive and useless information, which is not suitable for summarization tasks as well. Therefore, this finding provides a novel approach

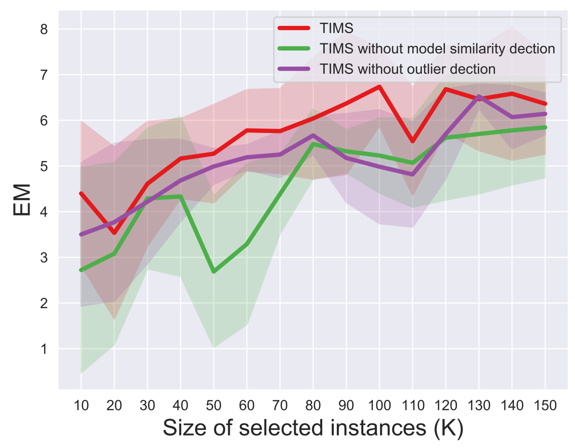


Fig. 6. Ablation studies of TIMS on the SQuAD dataset.

for outlier detection: since model similarity represents how far an instance's response deviates from the dataset's response, instances with lower scores are more likely to be outliers.

D. Ablation Study

Effects of Different Components Detailed ablation studies are conducted in the SQuAD dataset to examine the impact of each component on the results. In Fig. 6, we show the performance of TIMS and its individual components on different final sizes. It proves that both model similarity detection and outlier detection are necessary during the incrementally instance selection process. Comparing the purple and green lines, we can also see that model similarity detection plays a more important role, while outlier detection probably plays auxiliary role to avoid selected instances from being too similar.

TABLE I

COMPARISONS OF DIFFERENT APPROACHES FOR FEW-SHOT TEXT INSTANCE SELECTION (ROUGE-L REPORTED ON THE WIKILINGUA DATASET AND EM REPORTED ON THE SQUAD DATASET).

	WikiLingua				SQuAD			
	$K = 10$	$K = 50$	$K = 100$	$K = 150$	$K = 10$	$K = 50$	$K = 100$	$K = 150$
Random Sampling	17.26 ± 0.31	19.50 ± 0.27	19.68 ± 0.32	19.86 ± 0.28	1.70 ± 3.26	4.86 ± 3.45	5.22 ± 2.45	5.22 ± 1.84
K-means++ ($k = 1$) [4]	17.18 ± 0.17	19.53 ± 0.13	19.73 ± 0.32	19.90 ± 0.13	2.97 ± 2.68	$\mathbf{5.92} \pm 1.94$	5.17 ± 1.61	5.80 ± 1.00
K-means++ ($k = 10$) [4]	$\mathbf{17.55} \pm 0.20$	19.47 ± 0.24	19.83 ± 0.22	19.93 ± 0.19	4.00 ± 5.37	3.35 ± 2.32	6.04 ± 0.34	6.01 ± 2.42
Isolation Forest [21]	17.12 ± 0.27	19.55 ± 0.25	19.79 ± 0.17	19.86 ± 0.19	2.72 ± 2.37	2.69 ± 1.74	5.23 ± 0.86	5.85 ± 1.16
TIMS	17.49 ± 0.23	$\mathbf{19.92} \pm 0.14$	$\mathbf{19.95} \pm 0.19$	$\mathbf{20.05} \pm 0.16$	$\mathbf{4.40} \pm 1.66$	5.27 ± 1.13	$\mathbf{6.74} \pm 0.90$	$\mathbf{6.36} \pm 1.16$

TABLE II

THE FIVE INSTANCES WITH THE LOWEST SCORES USING MODEL SIMILARITY DETECTION. SINCE THE CONTENT OF THE THIRD INSTANCE IS TOO LONG AND IS REPEATED, THE MIDDLE PART IS OMITTED.

This goes great alongside hamburgers.
You can also use A4 paper. Admire your handiwork.
One: Unus Two: Duo Three: Tres Four: Quattuor Five: Quinque Six: $\cdots$ Hundred and forty-seven: Centum quadraginta septem Three-hundred and fifty-six: Trecenti quinquaginta sex Seven-hundred and eighty-five: Septingenti octoginta quinque
Headset settings have the same menus as the Speaker Settings.
if this method fails to work, try the other methods stated on this site.

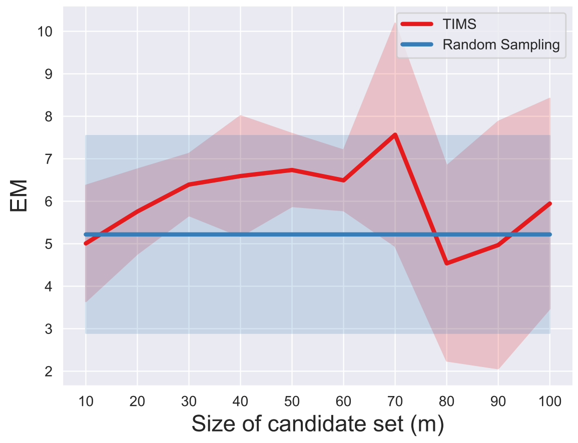


Fig. 7. Ablation studies of TIMS on different candidate set sizes (The number of selected instances is equal to 100).

Effects of candidate Set Sizes We investigate the effect of different candidate set sizes (m) on TIMS in the SQuAD dataset as well (as shown in Fig. 7). As can be seen from the figure, the candidate set size does have a certain impact on the results. When m is between 30-70, TIMS has a relatively satisfactory performance. As m increases, the variance (red area) of the results also increases gradually, which shows that blindly increasing m is not a good choice. Therefore, in this paper, we choose $m = 50$, which has a low variance and an acceptable performance.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we proposed a novel approach (TIMS) for incrementally few-shot text instance selection. It leverages information from parameters of pre-trained models instead of anisotropic sentence vectors and incrementally selects instances for annotation without the need for well-trained downstream models. Detailed Comparison with commonly used sampling strategies shows that the instances selected by TIMS perform better on downstream tasks. Moreover, TIMS provides a novel starting step before applying active learning and serves as a better benchmark for few-shot learning.

Important future directions for this work are: (i) testing on more different NLP tasks (e.g. data-to-text, machine translation) to demonstrate the applicability of our approach; (ii) Analyzing the impact of different outlier detection methods on the results; (iii) analyzing the minimum overhead when calculating the model similarity for each instance; (iv) and making TIMS more interpretable, so that we can understand why the algorithm select these instances instead of others.

VII. ACKNOWLEDGMENTS

This work was supported by the Joint Funds of the National Natural Science Foundation of China (Grant No. U21B2020).

REFERENCES

[1] Kalyan K S, Rajasekharan A, Sangeetha S. Ammus: A survey of transformer-based pretrained models in natural language processing[J]. arXiv preprint arXiv:2108.05542, 2021.

[2] Yin W. Meta-learning for few-shot natural language processing: A survey[J]. arXiv preprint arXiv:2007.09604, 2020.

[3] Wang Y, Yao Q, Kwok J T, et al. Generalizing from a few examples: A survey on few-shot learning[J]. ACM computing surveys (csur), 2020, 53(3): 1-34.

[4] Chang E, Shen X, Yeh H S and Demberg V. On Training Instance Selection for Few-Shot Neural Text Generation[J]. arXiv preprint arXiv:2107.03176, 2021.

[5] Wang L, Huang J, Huang K, et al. Improving neural language generation with spectrum control[C]//International Conference on Learning Representations. 2019.

[6] Li B, Zhou H, He J, et al. On the sentence embeddings from pre-trained language models[J]. arXiv preprint arXiv:2011.05864, 2020.

[7] Fu Y, Zhu X, Li B. A survey on instance selection for active learning[J]. Knowledge and information systems, 2013, 35(2): 249-283.

[8] Settles B. Active learning literature survey[J]. 2009.

[9] Balcan M F, Broder A, Zhang T. Margin based active learning[C]//International Conference on Computational Learning Theory. Springer, Berlin, Heidelberg, 2007: 35-50.

[10] Ren P, Xiao Y, Chang X, et al. A survey of deep active learning[J]. ACM Computing Surveys (CSUR), 2021, 54(9): 1-40.

[11] Hodge V, Austin J. A survey of outlier detection methodologies[J]. Artificial intelligence review, 2004, 22(2): 85-126.
[12] Chalapathy R, Chawla S. Deep learning for anomaly detection: A survey[J]. arXiv preprint arXiv:1901.03407, 2019.
[13] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
[14] Lewis M, Liu Y, Goyal N, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension[J]. arXiv preprint arXiv:1910.13461, 2019.
[15] Gururangan S, Marasović A, Swayamdipta S, et al. Don't stop pretraining: adapt language models to domains and tasks[J]. arXiv preprint arXiv:2004.10964, 2020.
[16] Liu F T, Ting K M, Zhou Z H. Isolation forest[C]//2008 eighth ieee international conference on data mining. IEEE, 2008: 413-422.
[17] El-Kassas W S, Salama C R, Rafea A A, et al. Automatic text summarization: A comprehensive survey[J]. Expert Systems with Applications, 2021, 165: 113679..
[18] Baradaran R, Ghiasi R, Amirkhani H. A survey on machine reading comprehension systems[J]. Natural Language Engineering, 2020: 1-50.
[19] Chang E, Shen X, Marin A, et al. The SelectGen Challenge: Finding the Best Training Samples for Few-Shot Neural Text Generation[J]. arXiv preprint arXiv:2108.06614, 2021.
[20] Liu P, Wang L, He G, et al. A Survey on Active Deep Learning: From Model-driven to Data-driven[J]. arXiv preprint arXiv:2101.09933, 2021.
[21] Das S, Wong W K, Fern A, et al. Incorporating feedback into tree-based anomaly detection[J]. arXiv preprint arXiv:1708.09441, 2017.
[22] Schick T, Schütze H. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners[J]. arXiv preprint arXiv:2009.07118, 2020.
[23] Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding[R]. Stanford, 2006.
[24] Fan Y, Gongshen L, Kui M, et al. Neural feedback text clustering with BiLSTM-CNN-Kmeans[J]. IEEE Access, 2018, 6: 57460-57469.
[25] Koh P W, Liang P. Understanding black-box predictions via influence functions[C]//International Conference on Machine Learning. PMLR, 2017: 1885-1894.
[26] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
[27] Liu F T, Ting K M, Zhou Z H. On detecting clustered anomalies using SCiForest[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2010: 274-290.
[28] Bruno R.. Preiss. Data Structure and Algorithms: With Object-oriented Design Patterns in Java[M]. John Wiley & Sons, 1999.
[29] Ladhak F, Durmus E, Cardie C, McKeown K. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization[J]. arXiv preprint arXiv:2010.03093, 2020.
[30] Rajpurkar P, Zhang J, Lopyrev K, Liang P. Squad: 100,000+ questions for machine comprehension of text[J]. arXiv preprint arXiv:1606.05250, 2016.
[31] Khandelwal U, Levy O, Jurafsky D, et al. Generalization through memorization: Nearest neighbor language models[J]. arXiv preprint arXiv:1911.00172, 2019.
[32] Shleifer S, Rush A M. Pre-trained summarization distillation[J]. arXiv preprint arXiv:2010.13002, 2020.
[33] Zheng C, Wang H J, Zhang K, et al. A Baseline Analysis for Podcast Abstractive Summarization[J]. arXiv preprint arXiv:2008.10648, 2020.
[34] Lin C Y. Rouge: A package for automatic evaluation of summaries[C]//Text summarization branches out. 2004: 74-81.
[35] Sanh V, Debut L, Chaumond J, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter[J]. arXiv preprint arXiv:1910.01108, 2019.
[36] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in neural information processing systems, 2019, 32: 8026-8037.
[37] Wolf T, Debut L, Sanh V, et al. Huggingface's transformers: State-of-the-art natural language processing[J]. arXiv preprint arXiv:1910.03771, 2019.
[38] Yang J, Zhang D, Frangi A F, et al. Two-dimensional PCA: a new approach to appearance-based face representation and recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2004, 26(1): 131-137.