Automatic Few-shot Selection on In-Context Learning for Aspect Term Extraction

Koki Imazato
Department of Creative Informatics
Kyushu Institute of Technology
Fukuoka, Japan
imazato.kouki927@mail.kyutech.jp

Kazutaka Shimada
Department of Artificial Intelligence
Kyushu Institute of Technology
Fukuoka, Japan
shimada@ai.kyutech.ac.jp

***Abstract*—In this paper, we handle aspect term extraction with In-Context Learning (ICL) as the target task. ICL is a method for learning Large Language Models. Instead of updating the model's parameters, prompts are provided to guide it to perform a target task. While the strength of ICL is that the model does not need to undergo additional training, the prompt includes input-output instances that cause instability in accuracy. The instances are called few-shot. Hence, selecting appropriate instances has the most important role in ICL. For this purpose, we propose a selection method with active learning. Active learning is a method of selecting instances that are useful to the model for training from unlabeled data. We regard the active learning-based approach as a sub-task for the target task. We introduce two types of sub-tasks and evaluate the effectiveness of them in the target task.**

***Index Terms*—Aspect term extraction, In-context learning, Few-shot learning, Active learning, GPT-4**

I. Introduction

In recent years, the field of natural language processing has made dramatic progress from the rise of Large Language Models (LLMs). In general, applying a language model to a target task requires additional training and parameter updating to generate a strong and robust model for the target task. However, preparing a large number of training data is costly. On the other hand, several LLMs can be applied to the target task without additional learning and parameter updating. One such learning method is In-Context Learning (ICL). ICL is not to update parameters in the LLM to a particular task. In ICL, the model predicts the output by using a prompt with the task description. ICL mainly consists of two types of learning methods: zero-shot learning (hereafter referred to as zero-shot) and few-shot learning (hereafter referred to as few-shot). In zero-shot, only the task description and test data are given to the prompt. In few-shot, not only the prompt in zero-shot, but also a small amount of instances, namely inputs and outputs, are contained in the prompt. In fact, large-scale pre-training models, such as llama [1] and GPT-3 (Generative Pre-trained Transformer-3), have shown excellent performance using zero-shot and few-shot. [2]. However, ICL is susceptible to the influence of the prompts provided and can be negatively affected by the random selection of the few-shot instances [3].

In this paper, we investigate a method for selecting the appropriate few-shot instances. We handle aspect term extraction as the target task and use GPT-4 as the LLM. In other words, this paper aims to select more effective few-shot instances to obtain the best prompts for GPT. For the selection of few-shot instances, active learning [4] is one prospective approach. Active learning is a method of selecting instances that are useful to the model for training from unlabeled data. In active learning, we call the few-shot instance selection a sub-task. Fig. 1 shows the outline of our method in this paper. The purpose of the target task is to extract an aspect term from a sentence, e.g., "pizza" from "this pizza is good", by using GPT, i.e., aspect term extraction without fine-tuning. Although GPT can deal with this task without fine-tuning, as mentioned above, the good few-shot instances contribute to improving the extraction accuracy. Therefore we introduce a selection model based on a pre-trained language model (PLM) in the sub-task. In this paper, we compare two types of similar NLP tasks in the sub-task: positive-negative classification and named entity recognition. Note that the training data for the sub-tasks are not related to the target task, namely aspect term extraction. The fine-tuned PLM for a sub-task selects instances from few-shot candidates on the basis of some criterion in the sub-task. By using the selected few-shot instances, GPT predicts aspect terms in a sentence from test data.

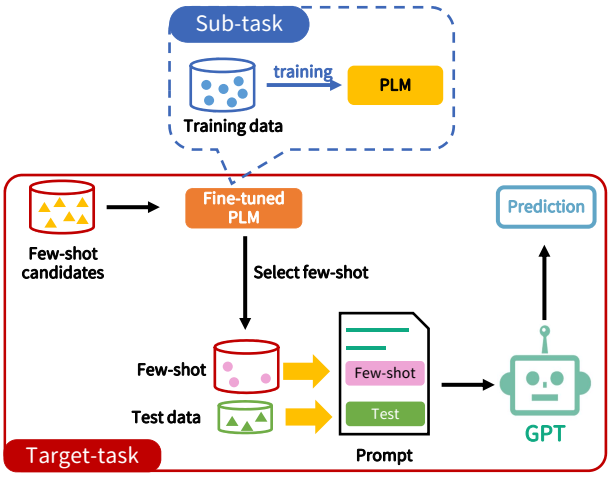

Fig. 1. The outline of our method in this paper.

The contributions of this paper are as follows:

- We propose a new approach for appropriately and automatically selecting instances from unlabeled data using fine-tuned models in a sub-task.
- We compare two types of sub-task models for the target task: positive-negative classification and named entity recognition for aspect term extraction.
- We show the effectiveness of the proposed method, as compared with the zero-shot approach and the random selection model.

II. Related Work

In ICL, models are not specialized for any particular task. By providing appropriate prompts, the models can obtain good performance even if the models do not use any training data. Radford et al. [5] have reported that models pre-trained with large data sets and parameters can achieve sufficient accuracy even in the zero-shot. The fact that zero-shot produces satisfactory results on unknown data indicates that ICL performs well without specialized data for a task.

Wan et al. [6] have evaluated an entity extraction task from low-resource data using zero-shot learning and randomly selected few-shot learning with GPT-3.5. Although ICL is effective in low-resource situations, one drawback is that its accuracy often depends on the few-shot instances. Liu et al. [7] have reported an approach to select appropriate few-shot instances in ICL. They proposed a method to search for data that are semantically similar to the test data and used the selected instances for the few-shot. As compared with a random selection approach, they showed that the selected instances were effective through several NLP tasks, such as reading comprehension and generation. Therefore, the selection has the most important role for ICL. We introduce two types of sub-tasks to the selection approach.

In this paper, we focus on active learning. Active learning can be used for a wide variety in NLP, such as parsing [8], machine translation [9], named entity recognition [10], and sentiment classification [11]. Active learning is to select appropriate instances that are useful for learning a model. It is natural for a few-shot selection to be effective. Köksal et al. [12] and Margatina et al. [13] have conducted studies combining active learning and ICL. Köksal et al. used a pre-trained language model to obtain features from unlabeled data and selected few-shot data based on these features. Margatina et al. compared several sampling approaches, such as uncertainty and similarity-based, for the selection. In this study, we also focus on the uncertainty sampling strategy in a sub-task.

III. Proposed Method

This section describes the proposed method for the target task in this paper, namely aspect term detection. The process consists of the main part, which uses GPT for the target task, and the sub-tasks, which are based on active learning.

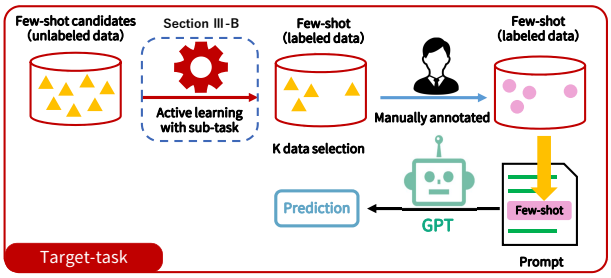

Fig. 2. The overview of the prediction process in the target-task.

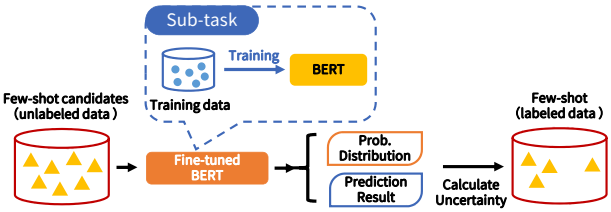

Fig. 3. Active learning algorithm.

A. Target task

Aspect term extraction aims to extract aspect terms, which evaluate a particular thing or case, from a review sentence. The following is an example of a sentence-aspect pair:

- Sentence: Italian pizza is the best.
- Aspect term: Italian pizza

Fig. 2 shows the overview of the proposed method. First, k-instances are selected from unlabeled candidates via active learning as explained later in Section III-B. Basically, the number of k is a small value and is determined experimentally. Since these instances are unlabeled, a human worker needs to annotate each instance. It is to detect aspect terms in each instance. The annotated instances are used as few-shot for GPT.

B. Sub-task

For the selection of few-shot instances, we focus on two sub-tasks: similar to the target task and similar to dataset characteristics. The first one is named entity recognition (NER) in this paper. Since the target task in this paper, namely aspect term extraction, is a sequence labeling problem, the task setting is similar to NER. The second one is the positive-negative (PN) classification of sentences. Since the target task is a task in sentiment analysis, the characteristics of a dataset for fine-tuning are similar to PN classification.

The overview of few-shot selection by uncertainty sampling strategy is shown in Fig. 3. Machine learning models are trained for each sub-task. The trained model is then used for the selection of few-shot instances using the uncertainty sampling strategy of active learning. The uncertainty sampling strategy is to preferentially select instances with a low confidence value from the trained model for the task.

1) NER: NER is to detect the named entities from the text. It also involves classifying each entity into categories such

as person, organization, location, and product. An example of named entities in a sentence is as follows:

- Sentence: Today I went from one station to another, including Tokyo, Shinagawa, and Ikebukuro, but Shinjuku Station was too crowded.
- Named Entities: [Location] Tokyo Shinagawa Ikebukuro Shinjuku

While NER is to extract entities in a sentence, the target task is to extract aspect terms. In addition, entities tend to be aspect terms. Therefore, there is a strong relationship between them.

We refer to this model as few-shot-L_{NER}. few-shot-L_{NER} is fine-tuned by a dataset about a NER task. Here, we go back to the above example. This sentence contains many entities. However, the target task is aspect term extraction. In the example, only "Shinjuku" is the aspect term in the sentence because there is no sentiment or opinion for "Tokyo", "Shinagawa", and "Ikebukuro". At the same time, "Shinjuku" is accompanied by "too crowded". Generally, the number of aspect terms is smaller than the number of entities. In other words, if a sentence containing many entities and correct aspect term information is obtained, it is beneficial as an instance for the few-shot learning[1]. Therefore, we adopt instances with a large number of entities.

2) PN: PN classification is to classify a sentence into positive or negative opinions. An example of the PN classification of a sentence is as follows:

- Sentence: The dish was tasty. I liked it.
- Sentiment: Positive

While PN classification is one of the sentiment analysis tasks, the target task, aspect term extraction, is also a famous sentiment analysis task. Therefore, datasets for both tasks have essentially similar characteristics, i.e., inside sentiment analysis.

We refer to this model as few-shot-L_{PN}. few-shot-L_{PN} is fine-tuned by a dataset about a PN classification task. Here, we introduce a score, the least confidence score (LCS). LCS is computed as follows:

$$\mathrm{LCS} = 1.00 - \max\left(\mathrm{softmax}\left(p\right)\right) \tag{1}$$

where p is the output vector of a PN classification model. In other words, it computes the complementary value of the maximum probability from the model for PN classification. It can assess the uncertainty of a prediction. The idea of LCS is illustrated in Fig. 4. For example, a sentence with low LCS denotes that the model can predict the sentiment with higher confidence. It implies that the sentence contains much information for sentiment analysis tasks because the PN classification is easy for the sentence. On the other hand, if a sentence produces high LCS, classifying the sentence into positive or negative is difficult. Here, assume that LLMs can identify aspect terms with some level of accuracy. In this situation, easy instances for the classification are not beneficial for the few-shot learning. On the other hand, information from difficult instances contributes to improving the accuracy of the target task because the instances deserve to be used as a prompt for LLMs. Therefore, we adopt instances with high LCS in the few-shot selection.

[1]Note that the extracted instance is annotated by a human worker for the target task, as mentioned in Section III-A.

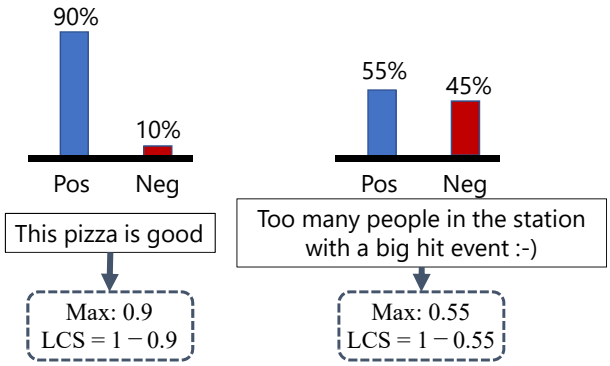

Fig. 4. The idea of LCS. The left sentence is easy for PN classification: high positive score and low negative score. As a result, LCS becomes low ($0.1 = 1.0 - 0.9$). On the other hand, the right one is difficult and the high LCS ($0.45 = 1.0 - 0.55$). We assume that a sentence with a high LCS is informative as a few-shot instance.

TABLE I
TYPES AND NUMBER OF NAMED ENTITIES IN THE DATASET

type	number
Personal name	2980
Corporate name	2485
Political organization name	1180
Other organization name	1051
Location name	2157
Facility name	1108
Product name	1215
Event name	1009

IV. EXPERIMENT

A. Dataset

We used a dataset developed by Kurihara et al. [14] for the aspect term extraction task. This dataset consists of tweets from Twitter. In the dataset, not only opinion terms but also aspect terms were annotated manually. The number of tweets for the experiment was 4762. We used 3809 tweets (4/5) as the candidate data for the few-shot selection and 953 tweets (1/5) as the test data, respectively.

For the few-shot selection, we need NER and PN classification models and datasets. We used a Japanese NER dataset[2] published by Stockmark as the training data. This dataset is manually annotated with named entities in sentences extracted from Wikipedia. The total number of data is 5343. Table I shows the distribution of named entities. We also used the ACP Corpus [15] developed by Kaji and Kitsuregawa for the PN classification model. We used 10000 sentences selected randomly from the corpus.

[2]https://github.com/stockmarkteam/ner-wikipedia-dataset

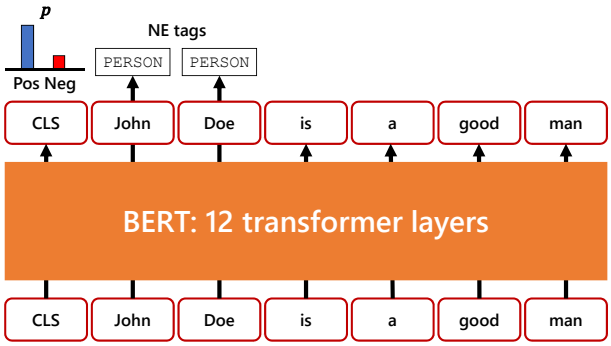

Fig. 5. BERT. In this paper, it classifies a sentence into positive or negative by using the probability from the CLS token and also recognizes the named entity labels of each token.

Task Description

Extract the aspect being assessed from the review sentence wrote in Japanese. For the aspect you extract please add special tokens '$$' and '$$' to surround it, and copy the rest content the same as the original sentence.
Notice that one sentence must have only one subject.
Below are some examples.

Few-shot

Input : 菊名駅鬼混み(笑)
Output : $$菊名駅$$鬼混み(笑)
Input : <mention> <mention> <mention> 青梅街道駅前のサンドイッチ屋さん、大変美味しゅうございました
Output : <mention> <mention> <mention> $$青梅街道駅前のサンドイッチ屋さん$$、大変美味しゅうございました
Input : 増毛駅前大混雑(^-^;)
Output : $$増毛駅$$前大混雑(^-^;)

Test

Input : 博多のラーメンが一番おいしい
Output :

Fig. 6. Prompt for the target task. The number of few-shot examples depends on the experiment setting. This figure denotes 3-shot examples.

B. Model

We used two neural-based models for the target task and the sub-tasks. The model of the target task is GPT, and that of the sub-tasks is BERT.

1) GPT-4: GPT-4, developed by OpenAI, is one of the most famous large-scale language models. GPT-4 is well-known as a model that is more accurate than GPT-3.5. In the experiment, the value of temperature was set to zero. Temperature is a parameter that controls the diversity of the generated text. The lower the temperature, the less diverse the generated sentences. In addition, more consistency in generation is expected.

2) BERT: For the sub-tasks, namely NER and PN classification, we used BERT [16] developed by Google. BERT is one of the most famous pre-trained language models based on the encoder in Transformers and 12 layers. It is well-known as a powerful model that can be fine-tuned by additional learning. We fine-tuned BERT with the dataset explained in Section IV-A. Fig. 5 shows the overview of BERT. We utilized BERT developed by Tohoku-University[3]. The loss function and optimizer were CrossEntropy and Adam. The batch size was 16. The learning rate and the number of epochs were 1e-5 and 15 for NER and 5e-5 and 8 for PN, respectively.

C. Setting and Baseline

Our method needs the number k, namely the number of instances in the few-shot learning. We compared $k = 2$ and $k = 8$ in this experiment. Hence, there are four types of proposed models: 2-shot-$\mathrm{L_{NER}}$, 8-shot-$\mathrm{L_{NER}}$, 2-shot-$\mathrm{L_{PN}}$, and 8-shot-$\mathrm{L_{PN}}$.

As baseline models, we used zero-shot learning. In addition, we compared our models with two random selection models: 2-shot-$\mathrm{L_{RAND}}$ and 8-shot-$\mathrm{L_{RAND}}$.

D. Prompt

GPT-4 needs a prompt for the target task. Fig. 6 shows the prompt that we used in the experiment. The prompt includes the task description, few-shot instances, and test data. Following [17], we specified that aspect terms in the output should be marked special symbols ($$).

[3]https://huggingface.co/tohoku-nlp/bert-base-japanese-v3

E. Criteria

We evaluated the accuracy of the target task with the exact match and partial match. The exact match is counted when all characters between the gold standard and the output from GPT-4 match. The partial match is counted when each character between the gold standard and the output from GPT-4 matches. The following example illustrates the exact and partial matches:

Sentence: Italian pizza is the best.

Gold	Italian pizza
Output	pizza
Exact	0
Partial	5

On the basis of the exact and partial matches, we compute recall and precision rates. Then we also compute the F1-score in the experiment.

F. Results and Discussion

Table II shows the experimental result. In the table, the bold values are the best scores in each criterion.

As compared with zero-shot-L, the performance of our methods is dramatically improved. The result shows the effectiveness of few-shot learning. For few-shot-$\mathrm{L_{NER}}$, 8-shots obtained better F1 scores than 2-shots (0.461 vs. 0.522 for Exact and 0.498 vs. 0.553 for Partial). Although the best F1

TABLE II
EXTRACTION ACCURACY OF ASPECT TERM

	Exact match			Partial match		
	Pre	Rec	F1	Pre	Rec	F1
zero-shot-L	0.226	0.224	0.225	0.311	0.380	0.342
2-shot-$\mathrm{L_{RAND}}$	0.554	0.552	0.553	**0.606**	0.587	0.596
2-shot-$\mathrm{L_{NER}}$	0.462	0.461	0.461	0.475	0.523	0.498
2-shot-$\mathrm{L_{PN}}$	**0.569**	**0.568**	**0.568**	0.585	0.628	0.606
8-shot-$\mathrm{L_{RAND}}$	0.537	0.535	0.536	0.549	**0.651**	0.596
8-shot-$\mathrm{L_{NER}}$	0.522	0.522	0.522	0.573	0.535	0.553
8-shot-$\mathrm{L_{PN}}$	0.558	0.557	0.558	0.594	0.632	**0.613**

score on the exact match was 2-shot-L_{PN}, the best F1 score on the partial match was 8-shot-L_{PN}. Overall, the accuracy tends to be improved by the larger number k. However, we need to verify and compare other numbers in detail, e.g., 5, 10, or more. On the other hand, as described in Fig. 2, our method needs correct labels for the few-shot instances. Hence, the number k should be small. The difference in accuracy between 2-shots and 8-shots is not large, especially few-shot-L_{PN}. This result shows the effectiveness of few-shot-L_{PN}, as compared with few-shot-L_{NER}.

G. Error analysis

Two examples of actual errors from few-shot-L_{NER} are as follows. In the examples, correct and predicted terms are in bold text.

- Correct: **横浜駅の再開発**が凄すぎるンゴ (***Yokohama Station redevelopment*** *is too amazing*)
- Prediction: **横浜駅**の再開発が凄すぎるンゴ (***Yokohama Station*** *redevelopment is too amazing*)
- Correct: **府中駅と府中本町駅と分倍河原駅**は統合しろ。乗り換え不便なんじゃー! (***Fuchu station, Fuchu Honmachi station, and Bubaigawara station*** *should be integrated. It's inconvenient to transfer!*)
- Prediction: **府中駅**と府中本町駅と分倍河原駅は統合しろ。乗り換え不便なんじゃー! (***Fuchu station***, *Fuchu Honmachi station, and Bubaigawara station should be integrated. It's inconvenient to transfer!*)

While the correct aspect terms tend to be long words, the predicted terms tend to be short. In other words, few-shot-L_{NER} tended to predict shorter lengths. One reason was the selection of few-shot instances. An actual instance from few-shot-L_{NER} is as follows:

- Sentence:「世界」の駅乗降車数ランキング。1 位・新宿駅 2 位・池袋駅 3 位・渋谷駅 4 位・大阪駅 (梅田駅含む)5 位・横浜駅 6 位・北千住駅 7 位・名古屋駅 (名鉄・近鉄含む)8 位・東京駅 9 位・品川駅 10 位・高田馬場駅、、、日本ってやばい国 ww (*World station passenger traffic rankings: 1st - Shinjuku Station, 2nd - Ikebukuro Station, 3rd - Shibuya Station, 4th - Osaka Station (including Umeda Station), 5th - Yokohama Station, 6th - Kitasenju Station, 7th - Nagoya Station (including Meitetsu and Kintetsu), 8th - Tokyo Station, 9th - Shinagawa Station, 10th - Takadanobaba Station... Japan is an amazing country lol*)
- Target: 日本 (*Japan*)

The basic idea of few-shot-L_{NER} is to select instances with many entities. On the other hand, as mentioned in Section III-B1, the number of aspect terms in a sentence is not always large. Moreover, the condition of a larger number of entities in a sentence implies that the sentence length is longer. As we can see in the above example, it contains many entities, namely station names, and one aspect term "Japan". As a result, GPT might learn from this instance; i.e., the aspect term is very short, as compared with the sentence length.

As mentioned in Section IV-F, few-shot-L_{PN} tended to obtain the better performance. One reason was that LCS is based on information from the inside of few-shot-L_{PN}, namely from CLS tokens in BERT, while few-shot-L_{NER} just used information of the output, namely the number of entities. few-shot-L_{NER} requires a selection approach that involves the inside of the model or the machine learning algorithm. This result shows the effectiveness of the selection by LCS.

TABLE III
NUMBER OF EXTRACTION FAILURE

	Number
zero-shot-L	12
2-shot-L_{RAND}	3
2-shot-L_{NER}	2
2-shot-L_{SA}	2
8-shot-L_{RAND}	3
8-shot-L_{NER}	0
8-shot-L_{SA}	3

Apart from the few-shot selection, there are some typical errors caused by GPT. i.e., GPT is a generative model based on large-scale pre-training. GPT sometimes wrongly adds supplementary information to the output on the basis of knowledge from pre-training. As a result, the output from GPT sometimes contained words that did not appear in the original sentence. The following is an example of this problem:

- Correct: **横浜駅からパシフィコ**まで歩いてみたけど少し後悔してる。 (*I tried walking* ***from Yokohama station to Pacifico****, but I regret it a little.*)
- Prediction: **横浜駅からパシフィコまでの道のり**歩いてみたけど少し後悔してる。 (*I tried walking* ***the road from Yokohama Station to Pacifico****, but I regret it a little.*)

The output sentence contains the word "の道のり (the road)" which does not exist in the input sentence. However, few-shot learning contributes to the reduction of this problem. Table III shows the number of such errors produced by zero-shot-L and each few-shot-L. As compared with zero-shot-L, the number of mistakes was reduced in few-shot-L. This is also the effectiveness of the few-shot learning.

V. Conclusion

We proposed a method of few-shot instance selection for in-context learning (ICL). We handle aspect term extraction as the target task. In few-shot selection, we used fine-tuned models by NER and PN classification datasets, respectively. The two few-shot example selection models were few-shot-L_{NER} and few-shot-L_{PN}. few-shot-L_{NER} adopted instances with large number of entities as few-shot. few-shot-L_{PN} adopted instances with high least confidence score (LCS) as few-shot. We evaluated the few-shot selection models with the extraction accuracy of the aspect term extraction from two perspectives: exact and partial matches. The few-shot-L_{PN} based on LCS achieved the highest F1 score of all methods: 2-shot for the exact match and 8-shot for the partical match. This result shows the effectiveness of the method based on selecting instances with LCS and PN classification for the aspect term extraction task with ICL. The few-shot-L_{PN} approach, LCS, is known as uncertainty sampling. There are several sampling

approaches in active learning, such as diversity sampling. In diversity sampling, different types of instances are selected. On the other hand, Wang et al. [17] have proposed a k-NN based approach that selects instances similar to the test data. Applying these approaches to our task is an interesting area for future work. In the future, we consider other types of combinations of a target task and sub-task.

ACKNOWLEDGMENT

This work was partly supported by JSPS KAKENHI Grant Number 23K11368.

REFERENCES

[1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

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

[4] Robert (Munro) Monarch. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Manning, 2021.

[5] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[6] Necva Bölücü, Maciej Rybinski, and Stephen Wan. impact of sample selection on in-context learning for entity extraction from scientific writing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5090–5107. Association for Computational Linguistics, 2023.

[7] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 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. Association for Computational Linguistics, 2022.

[8] Min Tang, Xiaoqiang Luo, and Salim Roukos. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 120–127. Association for Computational Linguistics, 2002.

[9] Yuekai Zhao, Haoran Zhang, Shuchang Zhou, and Zhihua Zhang. Active learning approaches to enhancing neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1796–1806. Association for Computational Linguistics, 2020.

[10] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 589–596, 2004.

[11] Shoushan Li, Shengfeng Ju, Guodong Zhou, and Xiaojun Li. Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 139–148. Association for Computational Linguistics, 2012.

[12] Abdullatif Köksal, Timo Schick, and Hinrich Schuetze. MEAL: Stable and active learning for few-shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 506–517. Association for Computational Linguistics, 2023.

[13] Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034. Association for Computational Linguistics, 2023.

[14] Satoshi Kurihara, Tomoya Mizumoto, and Kentaro Inui. [creation of a dataset of evaluated target and evaluative expression for sentiment analysis on twitter] twitter ni yoru hyouban bunseki wo mokuteki to shita hyouka taishou - hyouka hyougen detasetto sakusei (in japanese). *[The Association for Natural Language Processing]Gengo Shori Gakkai Dai 24 Kai Nenji Taikai Happyo Ronbunshu (in Japanese)*, pages 344–347, 2018.

[15] Nobuhiro Kaji and Masaru Kitsuregawa. Automatic construction of polarity-tagged corpus from HTML documents. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 452–459. Association for Computational Linguistics, 2006.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[17] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*, 2023.