Large Language Models for Few-Shot Automatic Term Extraction

Shubhanker Banerjee[1,2(✉)], Bharathi Raja Chakravarthi[2], and John Philip McCrae[1,2]

[1] ADAPT Centre, Dublin, Ireland
john.mccrae@adaptcentre.ie
[2] School of Computer Science, University of Galway, Galway, Ireland
shubhanker.banerjee@adaptcentre.ie,
bharathiraja.asokachakravarthi@universityofgalway.ie

Abstract. Automatic term extraction is the process of identifying domain-specific terms in a text using automated algorithms and is a key first step in ontology learning and knowledge graph creation. Large language models have shown good few-shot capabilities, thus, in this paper, we present a study to evaluate the few-shot in-context learning performance of GPT-3.5-Turbo on automatic term extraction. To benchmark the performance we compare the results with fine-tuning of a BERT-sized model. We also carry out experiments with count-based term extractors to assess their applicability to few-shot scenarios. We quantify prompt sensitivity with experiments to analyze the variation in performance of large language models across different prompt templates. Our results show that in-context learning with GPT-3.5-Turbo outperforms the BERT-based model and unsupervised count-based methods in few-shot scenarios.

Keywords: few-shot · automatic term extraction · large language models

1 Introduction

Terms are linguistic expressions that refer to domain-specific concepts and are integral to domain-specific languages (Cabré 1999; Fowler 2010). For instance, *Reinforcement Learning with Human Feedback* is relevant in natural language processing but not psycho-linguistics. Automatic Term Extraction (ATE) involves extracting terms from text using automated tools, and recent advancements in pre-trained language models (PLMs) have significantly improved ATE performance (Lang et al., 2021).

Large language models like GPT-3, with billions of parameters, excel in zero-shot and few-shot in-context learning for various NLP tasks (Brown et al., 2020). Building fully supervised term extraction models is costly and challenging due to domain-specific variations and the scarcity of annotated data. Large language

A. Rapp et al. (Eds.): NLDB 2024, LNCS 14762, pp. 137–150, 2024.
https://doi.org/10.1007/978-3-031-70239-6_10

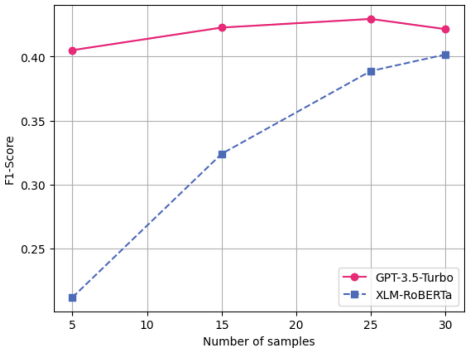

Fig. 1. Main findings: The figure shows F1-scores achieved by fine-tuned XLM-RoBERTa and GPT-3.5-Turbo averaged over 4 domains namely Heart Failure, Wind Energy, Equitation and Corruption. In-context learning with GPT-3.5-Turbo substantially outperforms model fine-tuning for 5, 15, 25 and 30 samples.

models, with extensive parameters and pre-training datasets, potentially alleviate the need for large annotated datasets by performing few-shot in-context ATE. To validate this, we compare large language models in few-shot settings with a BERT-sized XLM-RoBERTa model and unsupervised count-based term extraction methods, adhering to the truly few-shot setting standard (Perez et al., 2021).

To quantify this impact of prompt structure on task performance in the case of ATE we carry out experiments with 8 different prompt templates. Secondly, in recent studies, multiple in-context sample selection strategies have been explored (Rubin et al., 2022; Liu et al., 2022). In our experiments, we follow the k-nearest sample retriever method proposed by Liu et al. (2022) and carry out ablations to demonstrate its effectiveness.

The average F1 scores of fine-tuned XLM-RoBERTa and GPT-3.5-Turbo over 4 term annotated datasets are shown Fig. 1. We find that in-context learning with OpenAI's GPT-3.5-Turbo produces substantially better results as compared to the fine-tuned XLM-RoBERTa model. Additionally, experiments with unsupervised count-based term extractors demonstrate their ineffectiveness when compared to in-context learning in few-shot scenarios.

2 Related Work

Automatic Term Extraction. Machine learning-based methods aimed at identifying terms on the basis of underlying patterns in the occurrence context relax the frequency hypothesis with algorithms like random forests (Rigouts Terryn et al., 2021) and XGBoost (Hazem et al., 2022) demonstrating good task performance. The performance has been further improved by deep learning models such as XLM-RoBERTa (Lang et al., 2021) and mBERT (Hazem et al., 2022)

across languages and domains. Lang et al. (2021) show that formulating term extraction as a sequence labelling problem yields better results as compared to span classification or a sequence-to-sequence problem when fine-tuning XLM-RoBERTa.

Truly Few-Shot Learning. Perez et al. (2021) introduced the paradigm of truly few-shot learning where they argue that previous work which uses large validation sets for model and prompt selection overestimates the performance of pre-trained language models in few-shot scenarios. This paradigm has been followed by works focused on few-shot problems (Gutierrez et al., 2022).

Prompt Design. To apply large language models to ATE we formulate it as a language generation problem aided by prompts designed for this task. Our prompt templates are motivated by previous work which focuses on reformulating various natural language processing tasks as generation problems. In particular, our prompt design is inspired by work done by Gutierrez et al. (2022) where they pose relation extraction as a language generation problem. They break down each prompt into 2 main components: the task instruction and the retriever message. For more details on their prompt design, we refer the reader to their paper.

Large Language Models. In recent years there has been significant progress in the development of large language models (LLMs) (Naveed et al., 2023). These highly parameterized models have been able to achieve state-of-the-art performance on a wide variety of natural language processing tasks (Tang et al., 2023; Wadhwa et al., 2023). Inspired by the good performance of GPT-3.5-Turbo[1] model on information extraction tasks such as named entity recognition (Wang et al., 2023; Zhang et al., 2024) we carry out experiments to assess its applicability to few-shot automatic term extraction.

3 Methodology

We undertake a study to evaluate few-shot in-context learning performance of large language models[2] on term extraction. To benchmark these results we compare them against full model fine-tuning of a BERT-sized baseline PLM. The results are also compared against unsupervised count-based term extraction methods.

3.1 Validation Protocol

Perez et al. (2021) argue that hyperparameter tuning and prompt selection based on large validation sets are not truly representative of data-scarce few-shot learning scenarios. Furthermore, their experiments reveal that model selection decisions made on the basis of larger validation sets overestimate few-shot learning

[1] https://platform.openai.com/docs/models.

[2] Here we refer to models with more than 1B parameters as large language models.

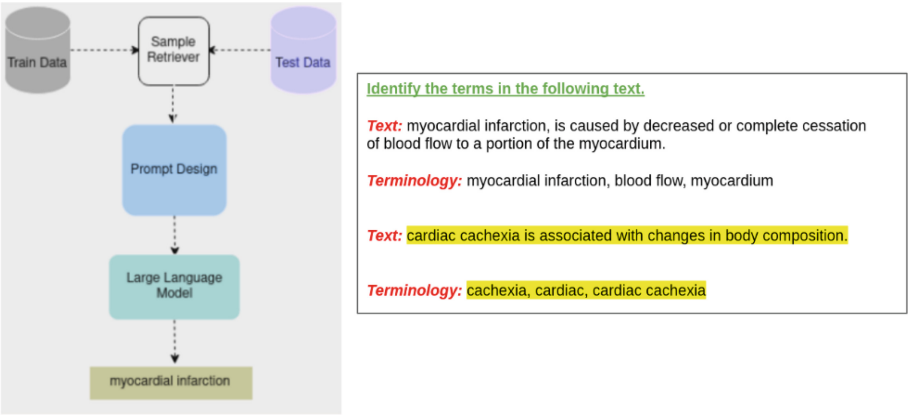

Fig. 2. Overall flow for large language model in-context learning for ATE in the left. The prompt structure is inspired by Gutierrez et al. (2022) and a prompt template used in our experiments with one-shot in-context learning has been shown on the right. The different segments of the prompt are indicated by different colours: task instruction in green (underlined) and red for the retrieval messages (italicized). The current input text and the terms output by the large language model are highlighted. (Color figure online)

performance. To avoid overestimation we follow their proposed truly few-shot setting and optimize the hyperparameters by validating on a validation set of 10 examples.

3.2 Large Language Model Evaluation

In this section, we discuss how we reformulate ATE for few-shot in-context learning. We then discuss the structure of prompts used for inference and also the approach we use for retrieval of in-context examples.

Task Formulation. In order to use large language models for ATE, we reformulate it as a language generation task. We tokenize the domain-specific corpus into sentences and transform each sentence into a prompt. The overall flow from the input sentence to the extracted terms is illustrated in Fig. 2.

Prompt Design. As shown by Sclar et al. (2023), in few-shot scenarios small changes to the prompt design have a significant impact on the performance of the large language models. To quantify the variation in performance across different prompts we carry out experiments with 8 different prompt templates. The basic structure of all our prompt templates is motivated by Gutierrez et al. (2022). Each prompt template comprises a task-specific instruction and a retriever message. We categorize the templates into two types: in Type I, we

Table 1. Prompt Templates

Prompt Type	No.	Template
Type I	1.	Identify the terms in the following text. Input: Output:
	2.	Identify the terms in the following text. Text: Terminology:
	3.	Extract the terms from the following text. Input: Output:
	4.	Extract the terms from the following text. Text: Terminology:
Type II	5.	Find the domain-specific terms in the following text. Input: Output:
	6.	Find the domain-specific terms in the following text. Text: Terminology:
	7.	Identify the words or phrases that are relevant to the underlying domain of the input text. Input: Output:
	8.	Identify the words or phrases that are relevant to the underlying domain of the input text. Text: Terminology:

include the templates with simple task instructions namely: *Identify the terms in the following text.* and *Extract the terms in the following text.* whereas in Type II the task instructions are more detailed such as *Identify the words or phrases that are relevant to the underlying domain of the input text.* and *Find the domain-specific terms in the following text.* We hope to analyze the sensitivity of LLMs to word-level changes in the prompt by quantifying the variation in performance within Type I. Furthermore, to analyze performance as a function of complexity of task instruction we compare the average F1 scores between Type I and Type II. For more details on individual prompt templates, we refer the reader to Table 1.

Evaluation. We evaluate the performance of the large language models in a few-shot in-context setting. Each prompt consists of the instruction, retriever message, input sentence and a fixed number of labelled examples selected by the retriever module to allow in-context learning. To better understand the impact of in-context samples on performance we carry out experiments with varying numbers of such samples in the prompt. An example of a one-shot prompt for a specific prompt template is shown in Fig. 2.

To control hallucination we post-process the outputs and remove any generated term which has no match with any span of text in the input.

Retriever Module. The inclusion of in-context samples in the prompt has been shown to improve performance on downstream tasks (Li and Qiu 2023). We follow the k-nearest neighbour sample selection method proposed by Liu et al. (2022) for our retriever module. The training set is used to select k most similar examples for each test sample. We carried out initial experiments with multilingual sentence transformers[3] and RoBERTa-large on a set of 250 randomly chosen examples. The findings of these initial experiments show that using RoBERTa-large for in-context demonstration retrieval yields better results. Therefore, we use RoBERTa-large for in-context demonstration retrieval in our experiments.

[3] Specifically we used *paraphrase-MiniLM-L6-v2*.

Table 2. Dataset statistics: The number of sentences and terms in each domain.

Heart Failure	Train	Test	Valid
Sentences	158	1427	43
Terms	260	1725	91
Corruption	Train	Test	Valid
Sentences	301	171	110
Terms	186	98	64
Equitation	Train	Test	Valid
Sentences	827	856	180
Terms	577	393	129
Wind Energy	Train	Test	Valid
Sentences	834	154	60
Terms	623	104	92

4 Experiments

4.1 Datasets

The ACTER dataset (Rigouts Terryn et al., 2020) contains term annotated data across 4 domains and 3 languages namely English, French and Dutch. The 4 domains are unrelated to each other and by carrying out experiments on all of them we hope to establish the applicability of large language models for few-shot term extraction to various domains. In this paper, we have limited the experiments to the English dataset. The dataset statistics can be found in Table 2. Here we provide a brief description of the domain corpora:

Heart Failure. This corpus is a collection of abstracts about heart failure collected on the basis of titles crawled for previous research (Hoste et al., 2019) on medical terminology extraction.

Equitation. The texts in the equitation corpus were collected manually from magazines and blogs and focus specifically on horseback riding.

Corruption. The texts in the corruption corpus belong to the juridical domain. These documents were manually collected from the EU, United Nations and Transparency International and contain legal documents about corruption policies, newspapers and Wikipedia articles.

Wind Energy. The documents in the wind energy corpus were collected from TTC corpus (Clouet et al., 2012).

4.2 Baselines

Unsupervised Term Extraction. Count-based term extractors rely on a positive correlation between termhood and the frequency of occurrence in a domain corpus. Since these methods are unsupervised, we use them as baselines for few-shot term extraction. In our experiments, we use C-Value (Frantzi and Ananiadou, 1996) and ComboBasic (Astrakhantsev et al., 2015) to benchmark the performance of large language models.

Fine-Tuning. Recent work on fine-tuning PLMs for ATE has reported good results by posing term extraction as a sequence tagging problem (Lang et al., 2021). Following this paradigm as the standard we fine-tune XLM-RoBERTa-base as the baseline PLM to identify terms using a BIO tagging scheme (Carreras et al., 2003). Hyperparameters such as learning rate, batch size, gradient accumulation steps, and warm-up ratio are optimized using tree-structured Parzen estimator[4] on the validation set of 10 examples mentioned above.

4.3 Implementation

To quantify variation in performance with a variation in the number of samples we carry out experiments with 5, 15, 25 and 30 training examples. GPT-3.5-Turbo has a maximum input context length of 4k tokens, to avoid errors due to exceeding the token limit in the input prompt we limit our experiments to 30 samples. 10 validation samples are used to simulate a truly few-shot setting. In order to limit costs, the models are evaluated on a test set of 150 samples. We use the Hugging Face library[5] for fine-tuning XLM-RoBERTa on our training set. We use PyATE library[6] for the implementation of the unsupervised count-based term extractors. OpenAI's open-source library is used to query GPT-3.5-Turbo[7]. To quantify the sensitivity of performance to various prompt templates we carry out all our experiments with the large language models using 8 different prompts. Hyperparameters are optimized using the Optuna library[8] over 20 trial runs with the tree-structured Parzen estimator. To provide robust and convincing conclusions, we run all experiments (including ablation studies) with 5 different seeds and report all results as the mean and standard deviation of all experiments. These measures are computed using Numpy library[9]. We used random seeds 1, 2, 3, 4 and 5 in our experiments. Furthermore, to ensure reproducibility we ensure that all the libraries/frameworks used in our experiments are open-source.

[4] https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html.
[5] https://huggingface.co/.
[6] https://github.com/kevinlu1248/pyate.
[7] https://github.com/openai/openai-python.
[8] https://optuna.org/.
[9] https://numpy.org/.

Table 3. The performance of fine-tuning XLM-RoBERTa with and without LoRA on K = 5, 15, 25 and 30 samples compared with unsupervised count-based term extractors and in-context learning with GPT-3.5-Turbo. This table illustrates the mean and standard deviation (in the format $mean_{std}$) of precision, recall and F1-scores over all the random seeds. The unsupervised methods are deterministic, they do not exhibit any variation across different runs therefore std = 0 has not been illustrated in the table. For GPT-3.5-Turbo the mean and standard deviation of F1-scores over all the random seeds and all the prompt templates have been illustrated.

5-shot				
	Heart Failure	Corruption	Wind Energy	Equitation
	Precision/Recall/F1	Precision/Recall/F1	Precision/Recall/F1	Precision/Recall/F1
GPT-3.5-Turbo	$47.8_{1.6}/61.5_{1.7}/53.7_{0.9}$	$18.2_{0.6}/69.8_{1.5}/28.9_{0.9}$	$19.5_{0.9}/71.9_{1.1}/30.7_{1.0}$	$35.6_{1.9}/76.6_{0.4}/48.6_{1.8}$
XLM-R	$10.3_{7.7}/13.0_{15.0}/10.0_{8.6}$	$26.1_{9.1}/10.3_{2.2}/14.1_{2.0}$	$19.0_{1.9}/30.5_{6.8}/23.1_{2.0}$	$33.2_{5.5}/45.0_{8.0}/37.3_{2.4}$
15-shot				
GPT-3.5-Turbo	$49.8_{2.0}/62.7_{1.4}/55.5_{1.0}$	$19.9_{0.9}/70.5_{0.9}/31.1_{1.2}$	$21.4_{0.4}/73.4_{1.3}/33.1_{0.4}$	$35.8_{1.1}/76.1_{0.3}/48.7_{1.0}$
XLM-R	$59.0_{3.3}/24.2_{5.4}/33.9_{4.8}$	$31.6_{6.5}/20.6_{3.9}/24.1_{1.9}$	$24.9_{2.6}/40.5_{4.3}/30.7_{2.5}$	$36.4_{3.1}/46.5_{2.3}/40.7_{2.3}$
25-shot				
GPT-3.5-Turbo	$51.1_{1.5}/63.4_{0.9}/56.6_{1.0}$	$19.9_{1.0}/67.0_{1.5}/30.6_{1.3}$	$21.2_{1.3}/71.3_{1.8}/32.7_{1.5}$	$36.5_{1.4}/74.6_{0.5}/49.0_{1.3}$
XLM-R	$60.4_{1.9}/41.4_{2.7}/49.0_{1.6}$	$33.3_{7.5}/23.8_{7.3}/26.1_{3.0}$	$29.2_{2.4}/50.5_{8.3}/36.5_{1.0}$	$46.4_{4.4}/42.0_{5.7}/43.6_{2.8}$
30-shot				
GPT-3.5-Turbo	$51.6_{1.7}/64.5_{0.8}/57.3_{0.8}$	$21.1_{0.5}/68.0_{2.9}/32.2_{0.8}$	$21.5_{0.5}/67.9_{1.9}/32.7_{0.6}$	$36.8_{0.9}/75.3_{1.9}/49.4_{0.8}$
XLM-R	$62.5_{2.3}/41.4_{5.2}/49.5_{4.5}$	$31.6_{5.7}/30.7_{3.0}/30.7_{1.5}$	$32.8_{3.3}/32.3_{6.2}/32.2_{6.7}$	$48.7_{4.7}/48.6_{7.5}/48.2_{4.7}$
Unsupervised Methods				
ComboBasic	5.5/38.2/9.6	2.3/36.5/4.3	4.7/34.0/8.2	1.8/58.1/3.6
CValue	5.4/37.7/9.5	2.2/35.5/4.2	4.6/33.8/8.2	1.8/57.1/3.6

5 Results and Discussion

5.1 Main Results

Our main experimental results can be found in Table 3. It is important to note that GPT-3.5-Turbo outperforms all other models in almost all cases on the F1-score, often by large margins in the range of approximately 3–45%. On the heart failure domain, GPT-3.5-Turbo has the best performance and achieves an average F1-score of 55.7% over all sample sizes. We also note that count-based term extractors are substantially outperformed by XLM-RoBERTa across all the domains, this is in line with previous results reported by Lang et al. (2021). This observation indicates that in few-shot scenarios in-context learning with LLMs is a better alternative than unsupervised count-based extractors.

While the overall better performance of the GPT model as compared to other models can be explained by its larger size and diverse pre-training corpus, the relatively lower performance on Corruption, Wind Energy and Equitation where the Precision drops by about 12–30% on average as compared to the Heart Failure domain indicates lack of domain specificity in the extracted terms. This drop in specificity is accompanied by substantial improvements in coverage on gold standard terms shown by the high recall values in the range of 70–76% on average across these domains. This observation indicates a significant number

Table 4. The average F1-scores of prompt templates calculated the GPT-3.5-Turbo model.

		Corruption	Heart Failure	Wind Energy	Equitation
Type I	Template 1	30.1	55.4	32.8	48.3
	Template 2	30.2	55.5	32.8	48.2
	Template 3	29.7	54.9	31.3	48.0
	Template 4	29.7	54.8	31.2	48.0
Average		29.9	55.1	32.0	48.1
Type II	Template 5	31.2	57.0	33.2	50.8
	Template 6	31.4	56.9	33.2	50.9
	Template 7	31.6	55.8	31.8	48.6
	Template 8	31.6	56.0	31.9	48.5
Average		31.4	55.6	32.5	49.7

of false positives in the extracted terms; we discuss this point in more detail in Sect. 5.4.

Diving into the finer details of XLM-RoBERTa fine-tuning, it is important to note that it has reasonable performance considering that the training set consists of a very small number of samples. Amongst all the domains XLM-RoBERTa has the best performance on Heart Failure, this can be attributed to regular term structure in this domain e.g. the suffixes '-tion', '-ophy' are common to many terms. Furthermore, the relatively higher values for ComboBasic on the Heart Failure domain suggest lexical overlap amongst the terms in this domain. Similarly, the high lexical diversity in term structures across the other domains can be attributed to the lower performances. In terms of coverage, XLM-RoBERTa behaves differently than GPT-3.5-Turbo and has high precision but low recall across the domains. The fact that GPT-3.5-Turbo makes predictions on the basis of knowledge acquired through a large pre-training corpus whereas XLM-RoBERTa is inherently regularized through task-specific fine-tuning can be used to explain this phenomenon.

We see an average improvement of around 2% for GPT-3.5-Turbo as the training set grows from 5 to 30 samples. This is important and shows that increasing the number of in-context samples arbitrarily does not guarantee large improvements in performance. However, as expected XLM-RoBERTa full-model fine-tuning exhibits large monotonic improvements in the F1-score going up to 50% with increasing sample size. Furthermore, as can be seen from the results in Table 3, for 30 samples XLM-RoBERTa converges on the GPT model.

5.2 Prompt Sensitivity

As described in Sect. 3.2 we categorize the templates into two categories: Type I with a simple task instruction and Type II with a detailed description of the task. The average score of each template is shown in Table 4. As can be seen

Table 5. Average F1-score of kNN-based demonstration retrieval compared to random demonstration selection for in-context learning.

	Corruption	Heart Failure	Wind Energy	Equitation
kNN	30.7	55.8	32.3	48.9
Random	28.7	54.9	30.2	48.0

from the results the performance of Type II templates is better than Type I on average. Furthermore, the change of retriever message has negligible impact on task performance ($\approx$ 0–1%).

This result is not surprising as the task instruction *Find the domain-specific terms in the following text.* in templates 5 and 6 and the task instruction *Identify the words or phrases that are relevant to the underlying domain of the input text.* in templates 7 and 8 describe the term extraction task in greater detail. A comparison with templates 1–2 with templates 3–4 shows that replacement of the word *Identify* with *Extract* leads to a slight degradation in performance. Thus indicating that although performance is sensitive to word-level changes, the impact may not be significant. Overall, the change in the prompt template did not lead to a large variation in the task performance; detailed task instructions had slightly better performance along expected lines.

5.3 Ablation

In Table 5, we present ablation studies demonstrating the effectiveness of the kNN-based demonstration selection used in our experiments. Experiments are carried out with randomly selected in-context samples instead of semantically similar samples selected through kNN for each test input without changing other aspects of the experimental setup. Comparison of the random demonstration retriever with kNN-based retriever module shows the better performance of kNN-based in-context sample selection strategy.

5.4 Error Analysis

As discussed in Sect. 5.1, GPT-3.5-Turbo suffers from the problem of high recall whereas full-model fine-tuning of XLM-RoBERTa leads to lower recall values on all four domains. In this section, we carry out a qualitative analysis of false positive and false negative predictions made by the models. We find that both models are good at identifying acronyms such as *LVEF* and *CRT*. Lang et al. (2021) make the same observation in their experiments as well. They attribute this to the presence of a substantial number of acronyms in the training dataset. However, the ability of GPT-3.5-Turbo to identify acronyms without task-specific adaptation is notable. A comparison of the terms generated by GPT-3.5-Turbo across the domains shows that while the predictions for Heart Failure are highly specific such as *biventricular* and *peak oxygen uptake*, for the other domains the

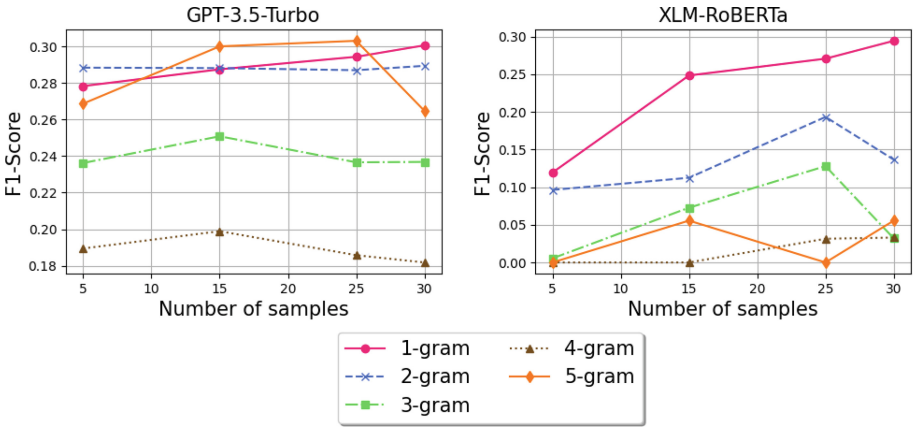

Fig. 3. Variation in average F1-scores of GPT-3.5-Turbo and full-model fine-tuned XLM-RoBERTa for varying sample sizes and term lengths. Overall results for term lengths ranging from 1 to 5 words are illustrated.

predictions include non-specific expressions. To illustrate, for Wind Energy we note the presence of a significant number of expressions such as *model* and *weight* which are not domain-specific. Similarly, we observe the presence of non-term expressions such as *diversion, addenda* in the output for the Corruption dataset and expressions like *touch, gymnastics* for the Equitation domain. These results indicate that while GPT-3.5-Turbo is good in highly specialized domains like Heart Failure, on more broader domains like Wind Energy issues of domain-specificity in the predictions arise. We attribute the lower precision of the GPT-3.5-Turbo model to the presence of such non-domain expressions in the output.

Experiments were conducted to evaluate the performance of GPT-3.5-Turbo and fine-tuned XLM-RoBERTa on extracting terms of varying lengths. The results have been demonstrated in Fig. 3. Both models performed well on shorter terms (1 to 2 words). However, GPT-3.5-Turbo outperformed XLM-RoBERTa on longer terms (3 to 5 words), with XLM-RoBERTa showing a performance gap of about 15% for 4-grams and 5-grams, likely due to the lack of longer terms in its training set.

6 Conclusion

In this work, we explored the potential of GPT-3.5-Turbo in-context learning for few-shot term extraction on 4 domains. We showed that for few-shot in-context term extraction, GPT-3.5-Turbo surpasses XLM-RoBERTa and count-based term extractors on all domains. Furthermore, the results show that even a small number of in-context samples leads to good task performance with diminishing gains as the number of in-context samples increases, this is an important result with the potential of significantly reducing costs associated with querying

the large language model. However, it is also important to note that the performance of XLM-RoBERTa converges on GPT-3.5-Turbo for 30 samples and while the input token limit of GPT-3.5-Turbo does not allow us to experiment with larger sample sizes, extrapolation of results shown here indicate that for larger sample sizes XLM-RoBERTa outperforms GPT-3.5-Turbo. We also discuss the performance of GPT-3.5-Turbo across the domains and show that while the extracted terms have high quality in specialized domains, for broader domains the performance drops. This is an open question with the potential of building a framework for term extraction with good performance on a wide range of domains. Besides posing this question we hope that this work can provide useful guidance for researchers working on few-shot term extraction.

7 Limitations

Although we have shown the good performance of GPT-3.5-Turbo in-context learning for term extraction as compared to fine-tuning a BERT-sized PLM for few-shot term extraction there are several limitations worth discussing. Due to budgetary constraints, we were limited to a smaller number of prompt templates. While our experiments show that variation in the prompt template doesn't cause significant variation in task performance, a wider search space can lead to better performance. To simulate a truly few-shot setting we have used a validation set of 10 samples, it is unclear if using a larger validation set at the cost of compromising the few-shot setting would reduce the gap in performance between XLM-RoBERTa and GPT-3.5-Turbo.set of prompt styles. Furthermore, here we have carried out experiments with GPT-3.5-Turbo, it is unclear whether in-context learning with other large language models will lead to an improvement in performance. We use kNN-based retriever module for selecting the in-context demonstrations, a retriever module better suited for selecting demonstrations for term extraction might lead to better results.

Acknowledgement. Author Shubhanker Banerjee was supported by Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT SFI Research Centre at University Of Galway.

References

Astrakhantsev, N.A., Fedorenko, D.G., Turdakov, D.Y.: Methods for automatic term recognition in domain-specific text collections: a survey. Ph.D. thesis (2015). https://doi.org/10.1134/S036176881506002X

Brown, T.B., et al.: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 6–12 December 2020, Virtual. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., Lin, H.-T. (eds.) (2020)

Cabré, M.T.: Terminology: Theory, Methods, and Applications, vol. 1. John Benjamins Publishing (1999)

Carreras, X., Màrquez, L., Padró, L.: Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, 31 May–1 June 2003. In: Daelemans, W., Osborne, M. (eds.), pp. 152–155. ACL (2003)

Clouet, E.L., Gojun, A., Blancafort, H., Guegan, M., Gornostay, T., Heid, U.: Reference lists for the evaluation of term extraction tools (2012)

Fowler, M.: Domain-Specific Languages. Pearson Education (2010)

Frantzi, K.T., Ananiadou, S.: 16th International Conference on Computational Linguistics, Proceedings of the Conference, COLING 1996, Center for Sprogteknologi, Copenhagen, Denmark, 5–9 August 1996, pp. 41–46 (1996)

Gutierrez, B.J., et al.: Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.), pp. 4497–4512. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.FINDINGS-EMNLP.329

Hazem, A., Bouhandi, M., Boudin, F., Daille, B.: Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20–25 June 2022. In: Calzolari, N., et al. (eds.), pp. 648–662. European Language Resources Association (2022)

Hoste, V., Vanopstal, K., Terryn, A.R., Lefever, E.: The trade-off between quantity and quality. Comparing a large crawled corpus and a small focused corpus for medical terminology extraction. Across Lang. Cult. **20**(2), 197–211 (2019)

Lang, C., Wachowiak, L., Heinisch, B., Gromann, D.: Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, 1–6 August 2021. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.), pp. 3607–3620. Association for Computational Linguistics (2021). https://doi.org/10.18653/V1/2021.FINDINGS-ACL.316

Li, X., Qiu, X.: Finding supporting examples for in-context learning. CoRR abs/2302.13539 (2023). https://doi.org/10.48550/ARXIV.2302.13539

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., Chen, W.: Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, 27 May 2022. In: Agirre, E., Apidianaki, M., Vulic, I. (eds.), pp. 100–114. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.DEELIO-1.10

Naveed, H., et al.: A comprehensive overview of large language models. CoRR abs/2307.06435 (2023). https://doi.org/10.48550/ARXIV.2307.06435

Perez, E., Kiela, D., Cho, K.: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, 6–14 December 2021, Virtual. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.), pp. 11054–11070 (2021)

Rigouts Terryn, A., Hoste, V., Drouin, P., Lefever, E.: Proceedings of the 6th International Workshop on Computational Terminology. In: Daille, B., Kageura, K., Rigouts Terryn, A. (eds.), pp. 85–94. European Language Resources Association (2020). ISBN: 979-10-95546-57-3

Rigouts Terryn, A., Hoste, V., Lefever, E.: HAMLET: hybrid adaptable machine learning approach to extract terminology. Terminology **27**(2), 254–293 (2021)

Rubin, O., Herzig, J., Berant, J.: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, 10–15 July 2022. In:

Carpuat, M., de Marneffe, M.-C., Ruíz, I.V.M. (eds.), pp. 2655–2671. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.NAACL-MAIN.191

Sclar, M., Choi, Y., Tsvetkov, Y., Suhr, A.: Quantifying language models' sensitivity to spurious features in prompt design or: how i learned to start worrying about prompt formatting. CoRR abs/2310.11324 (2023). https://doi.org/10.48550/ARXIV.2310.11324

Tang, L., et al.: Evaluating large language models on medical evidence summarization. NPJ Digit. Med. **6** (2023). https://doi.org/10.1038/S41746-023-00896-7

Wadhwa, S., Amir, S., Wallace, B.C.: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, 9-14 July 2023. In: Rogers, A., Boyd-Graber, J.L., Okazaki, N. (eds.), pp. 15566–15589. Association for Computational Linguistics (2023). https://doi.org/10.18653/V1/2023.ACL-LONG.868

Wang, S., et al.: GPT-NER: named entity recognition via large language models. CoRR abs/2304.10428 (2023).https://doi.org/10.48550/ARXIV.2304.10428

Zhang, M., Wang, B., Fei, H., Zhang, M.: In-context learning for few-shot nested named entity recognition. CoRR abs/2402.01182 (2024). https://doi.org/10.48550/ARXIV.2402.01182