Is Prompting What Term Extraction Needs?

Hanh Thi Hong Tran, Carlos-Emiliano González-Gallardo, Julien Delaunay, Antoine Doucet, Senja Pollak

Is Prompting What Term Extraction Needs?

Hanh Thi-Hong Tran[1,2,3][0000−0002−5993−1630], Carlos-Emiliano González-Gallardo[1][0000−0002−0787−2990], Julien Delaunay[1][0009−0001−9247−5745], Antoine Doucet[1][0000−0001−6160−3356], and Senja Pollak[3][0000−0002−4380−0863]

[1] University of La Rochelle, L3i, La Rochelle, France
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] Jožef Stefan Institute, Ljubljana, Slovenia
{thi.tran,carlos.gonzalez_gallardo,firstname.lastname}@univ-lr.fr
senja.pollak@ijs.si

Abstract. Automatic term extraction (ATE) is a natural language processing (NLP) task that reduces the effort of manually identifying terms from domain-specific corpora by providing a list of candidate terms. This paper summarizes our research on the applicability of open and closed-sourced large language models (LLMs) on the ATE task compared to two benchmarks where we consider ATE as sequence-labeling (*iobATE*) and seq2seq ranking (*templATE*) tasks, respectively. We propose three forms of prompting designs, including (1) sequence-labeling response; (2) text-extractive response; and (3) filling the gap of both types by text-generative response. We conduct experiments on the ACTER corpora in three languages and four domains with two different gold standards: one includes only terms (ANN) and the other covers both terms and entities (NES). Our empirical inquiry unveils that above all the prompting formats, text-extractive responses, and text-generative responses exhibit a greater ability in the few-shot setups when the amount of training data is scarce, and surpasses the performance of the templATE classifier in all scenarios. The performance of LLMs is close to fully supervised sequence-labeling ones, and it offers a valuable trade-off by eliminating the need for extensive data annotation efforts to a certain degree. This demonstrates LLMs' potential use within pragmatic, real-world applications characterized by the constricted availability of labeled examples.

Keywords: Term extraction · LLMs · prompting · in-context learning

1 Introduction

Terms are "*the designation of a defined concept in a special language by a linguistic expression.*" (ISO 1087). They are beneficial not only for several terminographical tasks by linguists (e.g., specialized dictionary construction [14]) but also for several downstream tasks (e.g., topic detection [5] and information retrieval [15]). To minimize the effort needed to extract terms from domain-specific corpora, automatic term extraction (ATE) approaches have been proposed.

TermEval 2020: Shared Task on Automatic Term Extraction, organized as part of the CompuTerm workshop [17], presented an important step forward

in systematic comparison among several ATE systems with the introduction of a new manually-annotated corpus, namely ACTER corpora [17]. The corpora contain domain-specific texts from four different fields in three languages with two versions of gold standards (with or without named entities). This is also the dataset on which we conduct our experiments.

After the evolution of transformer-based token classifiers toward term extraction (e.g., XLMR [21,22,23,24]), recent years witnessed the blossoming of large-scale generative models with the advent of prompt engineering [16]. Despite several works with state-of-the-art (SOTA) performance on downstream tasks [8,11], no application has been found on ATE tasks yet, and the performance of sequence-labeling tasks is still significantly below supervised baselines.

The main contribution of our work is threefold:

1. We conduct an empirical evaluation of term extraction using three distinct approaches, where we treat the task as (1) a sequence labeling task, (2) a seq2seq ranking task, and (3) a generative task using LLMs prompting;
2. We investigate the potential of LLMs' prompting for our ATE tasks to highlight their valuable insights in both rich- and low-resourced language niches;
3. We experiment with open and closed-sourced LLMs with comprehensive error analysis. This allows a task-oriented comparison among models and enriches the debate concerning the importance and utility of open LLMs.

This paper is organized as follows: Section 2 presents the related work while Section 3 describes the methods with the experimental setup, the datasets, and the evaluation metrics. The results with error analysis are discussed in Section 4 before we conclude with future work in Section 5.

2 Related work

2.1 Automatic Term Extraction

Traced back to the 1990s, term extraction was first proposed under the research of [4] with the two-step procedure: (1) extracting a list of candidate terms, and (2) determining their correctness. Traditional methods primarily relied on either linguistic or statistical aspects [6] or combined both [10]. The advancement of representation learning and neural networks has led to the exploration of various text embedding techniques for term extraction (e.g., local-global [1], non-contextual [26], and contextual [12] or their combinations [7]). Language models have also been applied to the task, as demonstrated in the *TermEval 2020* [17], e.g., feeding GloVe embeddings into a Bi-LSTM [17], feeding all possible extracted n-gram combinations into a BERT binary classifier [9]. In recent years, the evolution of term extraction has seen a shift towards treating the task as a sequence-labeling problem, extending beyond monolingual learning [12] to include cross-lingual and multilingual learning [13,21]. A systematic review of the tasks can be found in [20].

2.2 In-context Learning with LLMs

The emergence of LLMs has significantly improved performance across several downstream tasks [25]. Two strategies for incorporating LLMs into these tasks include fine-tuning and in-context learning (ICL). While the fine-tuning involves initializing a pre-trained model and conducting additional training epochs on task-specific supervised data, ICL leverages the LLM ability to generate texts with only a few task-specific examples as demonstrations. The concept of prompts with few-shot demonstrations was first introduced by [16], followed by an empirical analysis of the ICL paradigm with GPT-3 [2] and PaLM [3], in specific. With the release of ChatGPT[4] by OpenAI and the blossom of open-sourced LLMs, recent research focuses on evaluating its performance in various NLP tasks.We evaluate the performance of ChatGPT's reinforcement learning with human feedback (RLHF) model *gpt-3.5-turbo* and the open-sourced *Llama 2-Chat* model family with the ICL paradigm and compare it to the traditional sequence-labeling and fine-tuned seq2seq classifiers. To our knowledge, none of these two directions (seq2seq and LLMs) had been previously explored in the ATE task. Therefore, we would like to provide a comprehensive view of the ATE task from three perspectives: (1) as a token classification task; (2) as a seq2seq ranking task; and (3) for prompting.

3 Methodology

This section investigates the impact of semantically ambiguous and complex terms on prompt-based methods compared to the sequence-labeling baseline.

3.1 Sequence-labeling ATE

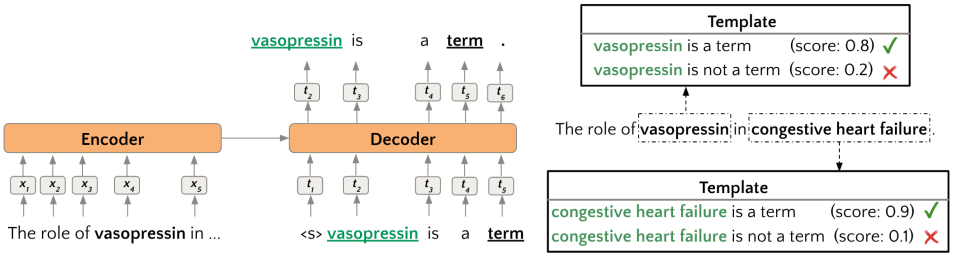
A common approach is to consider ATE as a token classification task, which assigns a label $y \in Y = \{B, I, O\}$ to each word x in a given sentence $X = \{x_1, ..., x_n\}$, where Y denotes the set of labels in IOB annotation regimes, and n denotes the length of the given sentence. We refer to this approach as *iobATE*. Inspired by [21]'s work, our baseline is the XLM-R token classifier with a standard hyperparameter configuration, which is also the SOTA in term extraction.

3.2 Template-based ATE

The task is formulated as a template-based (seq2seq) ranking problem for ATE (*templATE*), where the original sentence $X = \{x_1, ..., x_n\}$ is the source sequence, and the templates $\{t_1, ..., t_m\}$ filled by the candidate term span $x_{i:j}$ are the target sequence during training. The method contains the following steps:

1. Identify the gold standard terms in a sentence (e.g., *The role of* ***vasopressin*** *in* ***congestive heart failure****...*).

[4] https://openai.com/chatgpt



(a) *templATE* training architecture. (b) *templATE* inference.

Fig. 1: *templATE* architecture

2. Create a positive template for gold standard terms: $< MASK >$ is a term. (e.g., ***vasopressin*** *is a term*; ***congestive heart failure*** *is a term*; ...).
3. Create a negative template for the rest: $< MASK >$ is not a term. (e.g., ***The*** *is not a term*; ***role*** *is not a term*; ***of*** *is not a term*; ***in*** *is not a term*;...).
4. **Training**: Feed into the mBART [5] [18] the terms with their related positive and only the other 30% of negative ones to reduce imbalance. For example:
   - Sentence: *The role of* ***vasopressin*** *in* ***congestive heart failure***.
   - Output: *The* is not a term; *role* is not a term; *of* is not a term; *sentence* is not a term; *in* is not a term; *vasopressin* is a term; ...
5. **Inference**: Calculate the term score for each n-gram ($n = \{1, 2, 3, 4\}$). If the positive score is higher, consider it as a term.

We used mBART with 5 epochs, a batch size of 32, and a max sequence length of 70. The training and inference steps of the templATE approach are visualized in Figure 1a and 1b, respectively.

3.3 LLMs Prompting

We propose *promptATE*, which uses the close-sourced ChatGPT's *gpt-3.5-turbo*[6] and the open-sourced *Llama 2-Chat* (i.e., *Llama 2-Chat-7B*[7], *Llama 2-Chat-13B*[8], and *Llama 2-Chat-70B*[9]) RLHF models to address the ATE task. The approach follows the general paradigm of in-context (few-shot) learning with a three-step procedure as in Figure 2 where (1) **Task Description** instructs *promptATE* to detect the candidate terms using terminological knowledge; (2) **Few-shot Demonstrations** gives the model a few examples; and (3) **Input Sentence** indicates the input sentence while *promptATE*'s output is highlighted in green.

[5] https://huggingface.co/facebook/mbart-large-50-many-to-one-mmt
[6] https://platform.openai.com/
[7] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[8] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf
[9] https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

Task Description Given input sentence X, construct a $Prompt(X)$ to give a descriptive overview of the task with the following steps:

1. ***SYSTEM_PROMPT***: First, *"You are ... extraction (ATE) system."* tells *promptATE* to produce the output using terminological knowledge. Second, *"I will provide you ... extract the terms"* indicates the input information, including the domain and sentence having domain-specific terms while *"and the output ... with examples."* shows the position of few-shot demonstrations and marks the end of the description.
2. ***USER_PROMPT_1***: *"Are you clear about your role?"*. triggers a response by the assistant explicitly asking for confirmation of the task comprehension.
3. ***ASSISTANT_PROMPT_1***: *"Sure. I am ready to ... get started."* is the acknowledgment by *promptATE* but designed by the user only.
4. ***PROMPT***: This guideline prompt defines how *promptATE* should perform the ATE task. In the guidelines, we provided the requirements and the output format to guide *promptATE*'s responses for further processing.

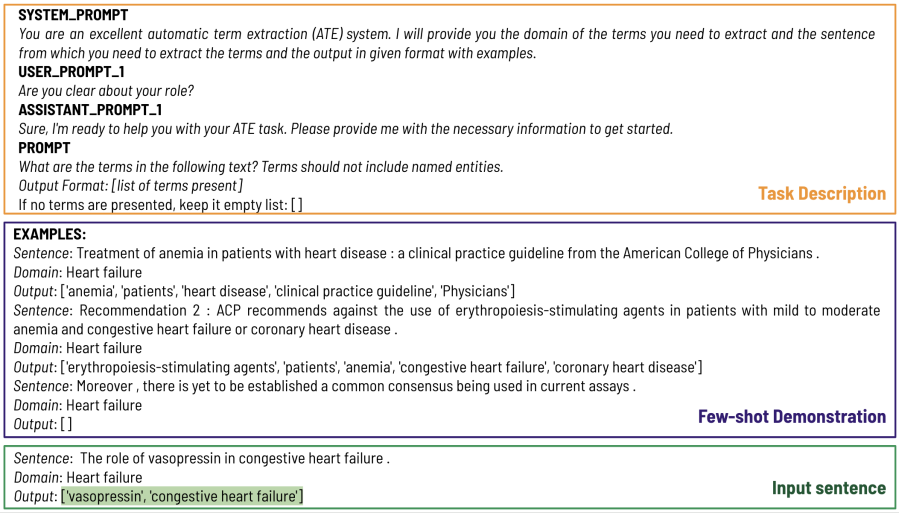


Fig. 2: A complete prompt with the output format #2 for the ANN version

Few-shot Prompting We focus on the few-shot demonstrations where we provide examples that are appended to the task description phase to regulate the format of outputs for each test input, as *promptATE* will generate outputs that mimic the demonstration format. For example, in the **Few-shot Demonstrations** rectangle of Figure 2, the demonstration sequentially packs a list of examples, each consisting of both the input and output sequences. The demonstration is set up as follows: The first two examples contain terms while the last

one is without terms inside the input sequence.All the examples of sequences are only from the test domain (Heart Failure) without any further information from the other three domains from ACTER corpora.

The following three output formats (OF) are tested: (1) *Sequence-labeling output (OF#1)*, where the output contains the information for each word label in the IOB annotation regime; (2) *List of candidate terms output (OF#2)*, which is the same format as our original gold standard; (3) *Generative output (OF#3)*, where we use unique tokens "@@" and "##" to surround the candidate terms.

ChatGPT vs. Llama 2-Chat We delved into the capabilities of few-shot demonstrations, employing both the close-sourced ChatGPT (*gpt-3.5-turbo*) [2] and the open-sourced *Llama 2-Chat* [19] RLHF models. While both exhibit remarkable language understanding and generation abilities, they employ divergent training methods and prompting mechanisms. Thus, slight modifications are required in the prompt structure while preserving the overarching concepts. By doing so, we aimed to evaluate how each model adapts to varying input cues and assess their respective adaptability in handling the same set of instructions. This study not only sheds light on the comparative performance of these RLHF models but also underscores their flexibility and versatility in comprehending and generating content, even when their underlying architectures differ significantly.

3.4 Data

The experiments have been conducted on ACTER v1.5 [17], a manually annotated collection of 12 comparable corpora (same domains in different languages) covering four domains (Corruption - Corp, Dressage - Equi, Wind energy - Wind, Heart failure - Htfl) in English, French, and Dutch. The corpora have two versions of gold standard annotations: one containing both terms and named entities (NES), and the other containing only terms (ANN). We apply the same configuration as in the TermEval 2020 shared task and related works [9,13,21] where Htfl domain of each language is considered the test set.

3.5 Evaluation metrics

The performance of each term extractor is assessed by strictly comparing the aggregated list of candidate terms identified across the entire test set against the manually designated gold standard list of terms, using precision, recall, and F1-score [9,13,21,22].

4 Results

Table 1 presents the comprehensive evaluation of three different ATE approaches on the Htfl domain in the ACTER dataset. For all our experiments, we fixed the Htfl domain as the test dataset, *iobATE* and *templATE* classifiers were trained

Table 1: Evaluation of different approaches on Htfl test set.

Settings	English Precision	English Recall	English F1-score	French Precision	French Recall	French F1-score	Dutch Precision	Dutch Recall	Dutch F1-score
ANN versions									
Benchmarks									
iobATE	58.4 ± 0.216	46.1 ± 3.861	51.4 ± 2.439	70.0 ± 0.852	39.9 ± 4.241	50.7 ± 3.691	72.4 ± 1.464	58.7 ± 3.387	64.7 ± 1.699
templATE	29.1 ± 3.445	25.1 ± 3.232	27.0 ± 3.240	33.0 ± 5.587	29.9 ± 5.025	30.6 ± 1.219	31.5 ± 1.359	42.3 ± 3.534	36.1 ± 2.175
$promptATE_{Llama2-Chat-7B}$									
OF#1	12.4	4.8	6.9	7.5	9.3	8.3	19.2	14.4	16.5
OF#2	↓ **40.4**$_{-18.0}$	↑ **62.6**$_{+16.5}$	↓ **49.1**$_{-2.3}$	36.3	↑ **59.2**$_{+19.3}$	↓ **45.0** $_{-5.7}$	40.4	↑ **73.1**$_{+14.4}$	↓ **52.0**$_{-12.7}$
OF#3	40.3	26.8	32.2	↓ **58.5** $_{-11.5}$	23.4	33.4	↓ **53.8**$_{-18.6}$	41.6	46.9
$promptATE_{Llama2-Chat-13B}$									
OF#1	12.1	1.7	3.0	11.2	6.6	8.3	25.6	5.9	9.6
OF#2	35.0	↑ **63.4**$_{+17.3}$	↓ **45.1**$_{-6.3}$	38.4	↑ **59.2**$_{+19.3}$	↓ **46.6**$_{-4.1}$	43.3	↑ **75.0**$_{+16.3}$	↓ **54.9**$_{-9.8}$
OF#3	↓ **40.0** $_{-18.4}$	36.9	38.4	↓ **41.0**$_{-29.4}$	48.7	44.5	↓ **46.1**$_{-26.3}$	56.2	50.7
$promptATE_{Llama2-Chat-70B}$									
OF#1	15.6	5.7	8.3	4.6	3.9	4.2	23.7	8.2	12.2
OF#2	36.8	↑ **65.9**$_{+19.8}$	47.2	38.0	↑ **64.8**$_{+44.9}$	47.9	42.3	↑ **74.8**$_{+16.1}$	54.0
OF#3	↓ **46.4**$_{-12.0}$	50.0	↓ **48.1**$_{-3.3}$	↓ **47.1**$_{-22.9}$	51.4	↓ **49.2**$_{-1.5}$	↓ **50.5**$_{-21.9}$	67.3	↓**57.7**$_{-7.0}$
$promptATE_{gpt-3.5-turbo}$									
OF#1	10.8	14.4	12.3	11.3	11.6	11.4	18.3	14.1	15.9
OF#2	26.6	↑ **67.6**$_{+21.5}$	38.2	28.5	↑ **67.0**$_{+27.1}$	40.0	36.8	↑ **79.6**$_{+20.9}$	50.3
OF#3	↓ **39.6**$_{-18.8}$	48.3	↓ **43.5**$_{-7.9}$	↓ **45.5**$_{-24.5}$	50.8	↓ **48.0**$_{-2.7}$	↓ **61.1**$_{-11.3}$	56.6	↓ **58.8**$_{-5.9}$
NES versions									
Benchmarks									
iobATE	63.0 ± 0.735	49.1 ± 3.014	55.2 ± 1.893	71.3 ± 1.330	45.4 ± 3.552	55.4 ± 3.074	74.0 ± 1.330	59.6 ± 1.322	66.0 ± 0.736
templATE	30.7 ± 3.122	31.2± 1.203	31.0 ± 2.164	36.1 ± 3.926	32.2 ± 6.193	33.8 ± 4.720	34.6 ± 4.678	43.4 ± 4.075	38.0 ± 1.239
In-domain $promptATE_{Llama2-Chat-7B}$									
OF#1	17.3	7.3	10.3	8.4	11.0	9.5	16.6	23.8	19.6
OF#2	42.9	↑ **63.4**$_{+14.3}$	↓ **51.2**$_{-4.0}$	36.0	↑ **61.6** $_{+16.2}$	↓ **45.4**$_{-10.0}$	40.3	↑ **75.6**$_{+16.0}$	↓ **52.6**$_{-13.4}$
OF#3	↓ **45.0**$_{-18.0}$	32.5	37.7	↓ **52.1**$_{-19.2}$	34.5	41.5	↓ **48.8**$_{-25.2}$	52.3	50.5
In-domain $promptATE_{Llama2-Chat-13B}$									
OF#1	25.9	2.4	4.4	8.4	5.3	6.5	23.5	5.9	9.4
OF#2	38.4	↑ **66.1**$_{+17.0}$	↓ **48.6**$_{-6.6}$	33.8	↑ **60.2**$_{+14.8}$	↓ **43.3**$_{-12.1}$	41.4	↑ **73.6**$_{+16.0}$	↓ **53.0**$_{-13.4}$
OF#3	↓ **40.3**$_{-22.7}$	47.5	43.6	↓ **35.7**$_{-35.6}$	49.4	41.4	↓ **47.9**$_{-26.1}$	49.1	48.5
In-domain $promptATE_{Llama2-Chat-70B}$									
OF#1	21.4	4.7	7.7	7.9	9.5	8.6	18.7	17.5	18.1
OF#2	39.9	↑ **67.2**+18.1	50.1	33.2	↑ **61.8**$_{+16.4}$	43.2	41.1	↑ **74.8**$_{+15.2}$	53.1
OF#3	↓ **48.3**$_{-14.7}$	54.9	↓ **51.4**$_{-3.8}$	↓ **40.8**$_{-30.5}$	57.7	↓ **47.8**$_{-7.6}$	↓ **53.1**$_{-20.9}$	57.9	↓ **55.4**$_{-10.6}$
In-domain $promptATE_{gpt-3.5-turbo}$									
OF#1	10.3	13.1	11.5	10.8	12.0	11.4	14.8	13.2	14.0
OF#2	29.2	↑ **69.2**$_{+20.1}$	41.1	27.9	↑ **66.8**$_{+21.4}$	39.4	39.8	↑ **78.5**$_{+18.9}$	52.8
OF#3	↓ **39.8**$_{-23.2}$	53.1	↓ **45.5**$_{-9.7}$	↓ **44.7**$_{-26.6}$	54.4	↓ **49.1**$_{-6.3}$	↓ **63.6**$_{-10.4}$	60.6	↓ **62.1**$_{-3.9}$

and validated with all possible combinations of the other three domains always having two domains for training and one domain for validation.

We present these combinations' average scores and standard deviations for both benchmarks. We emphasize the settings yielding the most favorable outcomes for each of the three approach types of *promptATE* by rendering them in bold. The arrows are used to compare our proposed methods and best benchmark for each setting, where ↑ is used to show the better performance of our approaches compared to the benchmark, while ↓ denotes the lower performance.

4.1 General Observations

As shown in Table 1, the *iobATE* approach consistently demonstrates a competitive balance between precision and recall, achieving a stable F1-score. This indicates the reliability of the fully supervised token classifier in terms of providing accurate predictions, but the approach requires a manually annotated training set. Comparatively, the *templATE* method showcases a mixed performance. While it can achieve high precision in certain scenarios, its recall lags, implying that it might struggle to identify all relevant examples, potentially resulting in missing important information. Compared to other approaches, there was a significant gap in F1-score performance.

The *promptATE* approach with in-domain few-shot demonstrations exhibits a considerable performance gap depending on the output format. It struggles with low precision and recall for sequence labeling (*OF#1*) compared to the others. This suggests the gap between the semantic labeling task and the text generation one, which open-sourced LLMs (i.e., *Llama 2-Chat*) and close-sourced LLMs (i.e., *gpt-3.5-turbo*) have been trained for. *OF#2* and *OF#3* show much higher scores compared to *OF#1*, even surpassing the *templATE*, and achieving competitive results to *iobATE* classifiers.

Results show variations given the language and model size, however, *Llama 2-Chat* with fewer parameters demonstrates to be better suited for listing candidate terms (*OF#2*) while *gpt-3.5-turbo* and the largest version of *Llama 2-Chat* show to be to a good option for generative output with specific delimiting tokens (*OF#3*). An interesting behavior is present for the *OF#2*, independently of the model, all recall scores are equal or higher than 59.2%, even reaching 79.6% in the case of *gpt-3.5-turbo* for Dutch. These scores outperform *iobATE* in the recall by an important margin, nevertheless in terms of precision, *promptATE* present a limited performance. We explain this by the complexity of the ATE task regarding the definition of a "term" inside a sentence, which is closely related to a specific domain. Generative models (i.e., *Llama 2-Chat* and *gpt-3.5-turbo*) can retrieve what can behave like a term but this leads to a big amount of false positives reflecting a high recall but limited precision.

4.2 Error Analysis

Impact of Output Formats As ATE is a sequence-labeling rather than a generative task, it is not readily suited for the ICL paradigm by default. Ad-

ditionally, the expected candidate terms to be extracted depend not only on the role that words occupy inside the sentence but also on a specific domain. *gpt-3.5-turbo* tends to generate outputs that exhibit a broader range of variability, often producing results that can be less predictable. However, *Llama 2-Chat* stands out for its remarkable ability to consistently adhere to the desired output structure and maintain a high level of reliability in generating content, especially *Llama 2-Chat-70B*. This contrast underscores the importance of choosing the right model for specific applications, where predictability and adherence are critical factors in decision-making processes and content generation.

IOB format (OF#1) contains the information for each word label and can be easily transformed into the term sequence. However, three main obstacles led to the poor performance in this format for all tested LLMs: (1) The model needs to learn the alignment between each position in the input sequence and the output labels, which naturally adds to the difficulty of the generation task; (2) It is difficult for the model to generate the output with the same length as the input sentence, especially when the input sentence is long, a case where the model is more likely to exhibit *hallucinations*; (3) The model either added an extra explanation per label of the input sequence or failed to provide the labels.

Despite reducing the obstacles from the previous format design, *List of candidate terms format (OF#2)* faces the following challenges: (1) The model failed to finish their predictions for elongated sentences containing multiple terms due to their limited amount of tokens as inputs and outputs by default; and (2) The model generated the predictions for candidate terms that do not appear in the original sentence (*hallucination*), which is mostly found in the Dutch corpus.

Text generation format (OF#3) solves to a certain degree the obstacles faced by the two previous formats. As the model only needs to mark the position of the terms and make copies for the rest, it can (1) significantly decrease the difficulty in generating text that fully encodes label information (as in *OF#1*) of the input sequence, (2) avoiding self-explanation and repetition of the few-shot demonstration, and (3) preventing the wrong output formats.

Impact of Term Length Variants To determine whether the term length affects the models' performance, we calculate the precision, recall, and F1-score of *promptATE* for terms of length $k = \{1, 2, 3, 4, \geq 5\}$ over the Htfl domain with both ANN and NES gold standards. Generally, as the term length increases, precision and recall decrease across most output formats and languages. This suggests that longer terms are more challenging to predict accurately, since they may have more variability and complexity, making them harder for the models to capture effectively. Independently of the approach, the highest F1-scores are achieved on the Dutch dataset. The proportion of different word lengths for the gold standard terms is shown in Table 2. It can be seen that, for the Htfl domain, the number of terms with more than one word ($k \geq 2$) is considerably smaller compared to French and English, which facilitates their extraction.

Besides, different output formats have also varying effects on model performance across term lengths and languages. *OF#1* consistently exhibits lower precision and recall compared to the other formats across all the term lengths,

indicating that it might not be as effective for ATE. For the English ANN version, *OF#2* has higher recall but lower precision in comparison with other formats across most term lengths. *OF#3* shows the best F1-score, striking a better balance between precision and recall. In the French and Dutch corpora, precision and recall are generally lower compared to the English ones, suggesting potential challenges in term extraction for these languages.

Table 2: Term proportion of different word lengths in each domain and language

Language	Domain	ANN version					NES version				
		k = 1	**k = 2**	**= 3**	**k = 4**	**k ≥ 5**	**k = 1**	**k = 2**	**= 3**	**k = 4**	**k ≥ 5**
English	**Corp**	389	377	117	30	14	502	419	146	52	54
	Equi	646	418	69	18	4	884	540	100	36	15
	Wind	319	527	198	39	8	565	639	245	58	24
	Htfl	1,064	767	358	118	54	1,170	801	377	142	91
French	**Corp**	440	326	131	51	31	550	356	158	75	68
	Equi	579	203	111	49	19	712	253	137	58	21
	Wind	315	232	122	65	39	446	265	128	74	55
	Htfl	1,207	604	264	79	74	1,309	620	266	91	88
Dutch	**Corp**	682	246	67	30	22	803	287	96	44	65
	Equi	1,091	185	65	37	15	1,181	224	82	40	17
	Wind	701	186	35	9	9	881	263	67	17	17
	Htfl	1,587	368	87	20	12	1,687	391	108	35	33

Impact of Language Distribution in pretraining The study by [19] pointed out that having a training dataset predominantly in English could potentially limit the model's effectiveness when used in languages other than English[10]. Despite French and Dutch not conventionally falling into the category of low-resourced languages, they are relatively under-resourced in the context of training LLMs, where English dominates (the pretraining distribution of French and Dutch accounts for 0.16% and 0.12% in *Llama 2-Chat*, 1.82% and 0.34% in *gpt-3.5-turbo* while English accounts for 89.70% in *Llama 2-Chat* and 92.65% in *gpt-3.5-turbo*. Our results indicate that our LLM prompting can indeed potentially enhance term extraction performance for under-represented languages.

5 Conclusions

In this paper, we evaluated the applicability of RLHF models toward ATE through an empirical study on different prompt designs in comparison with classical sequence labeling and the seq2seq approach. Although the RLHF models have achieved SOTA performances on various NLP tasks, there is still a gap between their performance in ATE and the fully supervised sequence-labeling baselines. We bridge the gap between the text generation and the sequence labeling task inherent in the ATE task by guiding the RLHF models to produce predictions with three designed formats.

Our empirical inquiry unveils that RLHF models exhibit a greater ability in the few-shot setups when the amount of training data is scarce and surpasses

[10] *"A training corpus with a majority in English means that the model may not be suitable for use in other languages." [19]*

the performance of *templATE* in all scenarios with the last two output-designed formats: (1) as a list of candidates terms, (2) encapsulating the candidate terms using specialized tokens. Its performance is not only close to fully supervised sequence-labeling baselines, but it offers a valuable trade-off by eliminating the need for extensive data annotation efforts as well. These findings demonstrate the capabilities of RLHF models' prompting to ATE tasks within pragmatic, real-world applications characterized by the constricted availability of labeled examples. Nevertheless, RLHF models, which are built upon LLMs, are pre-trained with an enormous amount of general data, making them agnostic to the specific domain of a term. This leads to an over-extraction of terms, resulting in good coverage but poor precision. In consequence, when a complete training dataset is accessible, opting for a fully-supervised ATE system remains the optimal choice.

Acknowledgements

The work was partially supported by the Slovenian Research and Innovation Agency (ARIS) core research program Knowledge Technologies (P2-0103) and projects Linguistic Accessibility of Social Assistance Rights in Slovenia (J5-50169) and Embeddings-based techniques for Media Monitoring Applications (L2-50070). The work has also been supported by the ANNA (2019-1R40226) and TERMITRAD (2020-2019-8510010) projects funded by the Nouvelle-Aquitaine Region, France. Besides, the work was supported by the project Cross-lingual and Cross-domain Methods for Terminology Extraction and Alignment, a bilateral project funded by the program PROTEUS under the grant number BI-FR/23-24-PROTEUS006.

References

1. Amjadian, E., Inkpen, D., Paribakht, T., Faez, F.: Local-Global Vectors to Improve Unigram Terminology Extraction. In: Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016). pp. 2–11 (2016)
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
3. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
4. Damerau, F.J.: Evaluating computer-generated domain-oriented vocabularies. Information processing & management **26**(6), 791–801 (1990)
5. El-Kishky, A., Song, Y., Wang, C., Voss, C.R., Han, J.: Scalable topical phrase mining from text corpora. Proc. VLDB Endow. **8**(3), 305–316 (nov 2014). https://doi.org/10.14778/2735508.2735519, `https://doi.org/10.14778/2735508.2735519`
6. Frantzi, K.T., Ananiadou, S., Tsujii, J.: The c-value/nc-value method of automatic recognition for multi-word terms. In: International conference on theory and practice of digital libraries. pp. 585–604. Springer (1998)

7. Gao, Y., Yuan, Y.: Feature-less End-to-end Nested Term extraction. In: CCF International Conference on Natural Language Processing and Chinese Computing. pp. 607–616. Springer (2019)
8. Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., Wu, Y.: How close is chatgpt to human experts? comparison corpus, evaluation, and detection (2023)
9. Hazem, A., Bouhandi, M., Boudin, F., Daille, B.: TermEval 2020: TALN-LS2N System for Automatic Term Extraction. In: Proceedings of the 6th International Workshop on Computational Terminology. pp. 95–100 (2020)
10. Kessler, R., Béchet, N., Berio, G.: Extraction of terminology in the field of construction. In: 2019 First International Conference on Digital Data Processing (DDP). pp. 22–26. IEEE (2019)
11. Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., Bielaniewicz, J., Gruza, M., Janz, A., Kanclerz, K., Kocoń, A., Koptyra, B., Mieleszczenko-Kowszewicz, W., Miłkowski, P., Oleksy, M., Piasecki, M., Łukasz Radliński, Wojtasik, K., Woźniak, S., Kazienko, P.: Chatgpt: Jack of all trades, master of none (2023)
12. Kucza, M., Niehues, J., Zenkel, T., Waibel, A., Stüker, S.: Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks. In: INTERSPEECH. pp. 2072–2076 (2018)
13. Lang, C., Wachowiak, L., Heinisch, B., Gromann, D.: Transforming term extraction: Transformer-based approaches to multilingual term extraction across domains. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 3607–3620 (2021)
14. Le Serrec, A., L'Homme, M.C., Drouin, P., Kraif, O.: Automating the compilation of specialized dictionaries: Use and analysis of term extraction and lexical alignment. Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication **16**(1), 77–106 (2010)
15. Lingpeng, Y., Donghong, J., Guodong, Z., Yu, N.: Improving retrieval effectiveness by using key terms in top retrieved documents. In: European Conference on Information Retrieval. pp. 169–184. Springer (2005)
16. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
17. Rigouts Terryn, A., Hoste, V., Drouin, P., Lefever, E.: TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset. In: 6th International Workshop on Computational Terminology (COMPUTERM 2020). pp. 85–94. European Language Resources Association (ELRA) (2020)
18. Tang, Y., Tran, C., Li, X., Chen, P.J., Goyal, N., Chaudhary, V., Gu, J., Fan, A.: Multilingual translation with extensible multilingual pretraining and finetuning. arXiv preprint arXiv:2008.00401 (2020)
19. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
20. Tran, H.T.H., Martinc, M., Caporusso, J., Doucet, A., Pollak, S.: The recent advances in automatic term extraction: A survey. arXiv preprint arXiv:2301.06767 (2023)
21. Tran, H.T.H., Martinc, M., Doucet, A., Pollak, S.: Can cross-domain term extraction benefit from cross-lingual transfer? In: Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, Proceedings. pp. 363–378. Springer (2022)

22. Tran, H.T.H., Martinc, M., Pelicon, A., Doucet, A., Pollak, S.: Ensembling transformers for cross-domain automatic term extraction. In: From Born-Physical to Born-Virtual: Augmenting Intelligence in Digital Libraries: 24th International Conference on Asian Digital Libraries, ICADL 2022, Hanoi, Vietnam, November 30–December 2, 2022, Proceedings. pp. 90–100. Springer (2022)
23. Tran, H.T.H., Martinc, M., Repar, A., Ljubešić, N., Doucet, A., Pollak, S.: Can cross-domain term extraction benefit from cross-lingual transfer and nested term labeling? Machine Learning pp. 1–30 (2024)
24. Tran, H., Martinc, M., Doucet, A., Pollak, S.: A transformer-based sequence-labeling approach to the slovenian cross-domain automatic term extraction. In: Slovenian Conference on Language Technologies and Digital Humanities (2022)
25. Vilar, D., Freitag, M., Cherry, C., Luo, J., Ratnakar, V., Foster, G.: Prompting palm for translation: Assessing strategies and performance. arXiv preprint arXiv:2211.09102 (2022)
26. Zhang, Z., Gao, J., Ciravegna, F.: Semre-rank: Improving automatic term extraction by incorporating semantic relatedness with personalised pagerank. ACM Transactions on Knowledge Discovery from Data (TKDD) **12**(5), 1–41 (2018)