Zero-shot Entity Recognition for Polymer Biodegradability Information: GPT-4o on PolyBD

Shanshan Liu[1] Masashi Ishii[2] Yuji Matsumoto[1]
[1]Center for Advanced Intelligence Project, RIKEN
[2]Material Database Group, MaDIS, NIMS
{shanshan.liu, yuji.matsumoto}@riken.jp
ishii.masashi@nims.go.jp

Abstract

To investigate polymer biodegradability information extraction, we constructed PolyBD, a manually annotated dataset containing entity annotations of 100 journal articles. We evaluated the performance of GPT-4o in sentence-level entity recognition under a zero-shot setting on PolyBD. While GPT-4o achieved strong overall results, its performance differed markedly between nested entities (those contained within other entities) and non-nested entities (all others). Specifically, it achieved a recall of 78% for nested entities but only 56% for non-nested entities. These results underscore both the capabilities and limitations of advanced large language models in addressing real-world extraction tasks.

1 Introduction

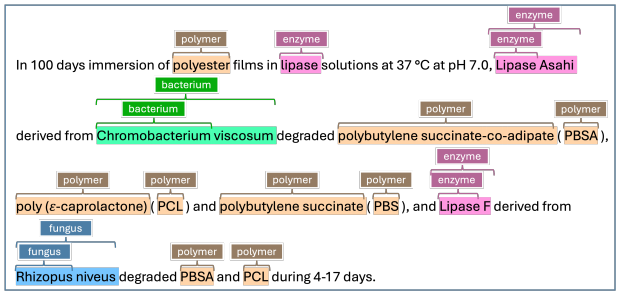

Figure 1 An annotated sentence in PolyBD.

Information extraction techniques have been widely applied across scientific domains [1], including biomedicine [2], chemistry [3], and computer science [4]. However, certain fields remain under-explored despite the critical importance of domain-specific information for societal and environmental advancement. This work sheds light on the polymer biodegradability information, which is essential for advancing material innovation, improving waste management, and informing policy and industrial practices.

The development of automated tools for extracting polymer biodegradability data can greatly improve the efficiency, accessibility, and applicability of existing research, thereby accelerating scientific advancement and practical implementation. This study, conducted in collaboration with material domain experts, aims to foster knowledge integration for enhanced material circulation.

To support the development of accurate, domain-specific models, we present **PolyBD**, an on-progressing dataset focused on polymer biodegradability. The dataset consists of 100 research articles documenting interactions between microorganisms or enzymes and polymers. Each article has been manually segmented into sentences and annotated at the entity level, capturing polymers, bacteria, fungi, and enzymes (see Figure 1).

To improve the utility of annotations for domain experts, entities were annotated at multiple hierarchical levels. For example, as illustrated in Figure 1, both *"Chromobacterium viscosum"* (species) and *"Chromobacterium"* (genus) are annotated. Future annotation efforts will link these bacterium entities to their corresponding ontology entries. The polymer *"polybutylene succinate-co-adipate"* will be associated with genus- and species-level annotations during the relation annotation process, enabling a comprehensive understanding of polymer-bacterium interactions.

PolyBD contains a substantial number of nested entities — entities contained within others, such as *"Chromobacterium"* and *"Rhizopus"* in Figure 1 — presenting considerable challenges for extraction. Methods capable of addressing nested Named Entity Recognition (NER) are scarce, particularly in specialized domains [5]. Given the knowledge-intensive nature of this task — distinguishing

In 100 days immersion of polyester films in lipase solutions at 37 °C at pH 7.0, Lipase Asahi derived from Chromobacterium viscosum degraded polybutylene succinate-co-adipate (PBSA), poly (ε-caprolactone) (PCL) and polybutylene succinate (PBS), and Lipase F derived from Rhizopus niveus degraded PBSA and PCL during 4-17 days.

Attribute	Event 1	Event 2	Event 3	Event 4	Event 5
Polymer	polybutylene succinate-co-adipate / PBSA	poly (ε-caprolactone) / PCL	polybutylene succinate / PBS	PBSA	PCL
Bacterium	Chromobacterium viscosum	Chromobacterium viscosum	Chromobacterium viscosum	-	-
Fungus	-	-	-	Rhizopus niveus	Rhizopus niveus
Enzyme	Lipase Asahi / lipase	Lipase Asahi / lipase	Lipase Asahi / lipase	Lipase F / lipase	Lipase F / lipase
Condition	at 37 °C at pH 7.0 / 100 days immersion	at 37 °C at pH 7.0 / 100 days immersion	at 37 °C at pH 7.0 / 100 days immersion	at 37 °C at pH 7.0 / 4-17 days / immersion	at 37 °C at pH 7.0 / 4-17 days / immersion
Result	degraded	degraded	degraded	degraded	degraded

Figure 2 An example illustrating our ultimate goal in constructing PolyBD involves annotating polymer biodegradability information as events with six attributes given a paragraph.

Table 1 Type distributions within the PolyBD dataset.

Entity Type	An Example	Count
Polymer	poly(butylenes succinate)	3327
Bacterium	C. acidovorans TB35	4928
Fungus	Rhizopus niveus	1637
Enzyme	Lipase F	3742
All	-	13634

between entities such as bacteria and fungi often requires more than contextual sentence cues — and the superior performance of recent Large Language Model (LLM)-based NER approaches in zero- and few-shot scenarios, we selected GPT-4o [1] as the baseline to evaluate the complexity and challenges of our task.

Our contributions are:

- To the best of our knowledge, we present the first dataset dedicated to information extraction on polymer biodegradability.
- We report experimental results for named entity recognition on the PolyBD dataset using GPT-4o in a zero-shot setting, aiming to advance research and applications in this domain.
- Our analysis reveals a significant performance gap between non-nested and nested entities by GPT-4o, highlighting the need for further investigation.

2 Dataset - PolyBD

We are constructing a dataset of 100 journal articles that investigate the interactions between polymers, microorganisms, and enzymes in the context of polymer biodegradability. This dataset is named PolyBD (**Poly**mer **Bio**d**d**egradability dataset). PolyBD is structured to identify

1) https://openai.com/index/hello-gpt-4o/

```
# Instruction
Analyze the following sentence and identify all named entities corresponding to a polymer, a bacteria, a fungi or an enzyme, including nested entities. For each entity, specify its type (e.g., polymer, bacteria, fungi, enzyme.) and its position in the sentence. If a named entity does not belong to any given types, please do not output that named entity. If there is no valid named entity, please output: {"entity": "None"}
# Sentence
{sentence}
# Output format
Given the sentence, you first need to identify entities of the given types. Then, you need to output entities by filling in the placeholders in [...] in a JSON format:
{ "entity 1": {"Entity": "...", "Type": [polymer/bacteria/fungi/enzyme], "Position": [entity start, entity end]},
"entity 2": {"Entity": "...", "Type": [polymer/bacteria/fungi/enzyme], "Position": [entity start, entity end]}, }
# Answer
```

Figure 3 Prompt template for NER.

```
# Instruction
Analyze the following sentence and identify all named entities corresponding to a polymer, a bacteria, a fungi or an enzyme, including nested entities. Recognizing nested entities means when encountering "Pseudomonas putida HI47" in a sentence, the entities to be identified include "Pseudomonas putida HI47", "Pseudomonas putida" and "Pseudomonas". Similarly, for "Pseudomonas mendocina lipase", the entities "Pseudomonas mendocina lipase", "Pseudomonas mendocina" and "Pseudomonas" also should be recognized. In the case of "PLLA/P(3HB-co-4HB)", "PLLA/P(3HB-co-4HB)", "PLLA" and "P(3HB-co-4HB)" must also be predicted. For each entity, specify its type (e.g., polymer, bacteria, fungi, enzyme.) and its position in the sentence. If a named entity does not belong to any given types, please do not output that named entity. If there is no valid named entity, please output: {"entity": "None"}
...
```

Figure 4 Prompt template with three examples of nested entities for NER.

which polymers are degraded by specific bacterium or fungus, what enzyme is involved, the conditions under which degradation occurs, and the resulting effects.

PolyBD is designed for a task akin to event extraction, where each event comprises six attributes: Polymer, Bacterium, Fungus, Enzyme, Condition, and Result, but lacks an event trigger. To improve reliability and usability, bacterium and fungus entities will be aligned with the NCBI Taxonomy Database[2], and enzyme entities will be aligned with the ENZYME Ontology[3] in future annotation processes. An example of the input and structured output data

2) https://www.ncbi.nlm.nih.gov/taxonomy
3) https://enzyme.expasy.org/

we will construct is shown in Figure 2.

At this stage, we manually segmented sentences (17,357 sentences) and annotated entities (22,777 entities) across all papers. Four entity types were defined and annotated: Polymer, Bacterium, Fungus, and Enzyme. Due to challenges in entity-level annotation, Condition and Result attributes are excluded from the current NER task. 3,638 sentences out of the whole dataset containing at least one Bacterium/Fungus/Enzyme entity are selected for evaluation, covering 13,634 entities. The statistics of entity types are presented in Table 1. PolyBD features a significant proportion (32%) of nested entities.

3 Experiments

3.1 Zero-shot NER

For a given input sentence, we applied zero-shot prompting with GPT-4o to perform sentence-level NER using two distinct prompts, aiming to evaluate whether explicitly highlighting nested entities in the task instruction improves the LLM's performance of recognizing them. Prompt 1 (Figure 3) provides a straightforward instruction to perform NER, specifying the extraction of nested entities and targeted entity types. Prompt 2 (Figure 4) is expanded on Prompt 1 by including three illustrative examples of nested entities, clarifying the definition of nested entities, and detailing the expected output format. GPT-4o was accessed via OpenAI's API with default settings.

3.2 Evaluation

We evaluated model performance using Precision (P), Recall (R), and Micro F1 (F1), and reported results under two criteria for correct prediction: (i) **Expression**: An entity is considered correct if it matches the expression of a gold entity in the sentence. (ii) **Expression + Type**: An entity is correct if it matches both the expression and type of a gold entity in the sentence.

Under the "Expression" criterion, entities sharing the same expression but differing in type or location were consolidated. Similarly, under the "Expression + Type" criterion, entities with the same expression and type but differing locations were merged. For instance, if two entities, ["PCL", "Polymer", [40, 43]] and ["PCL", "Polymer", [50, 53]], were present in a sentence, they were unified as a single entity ["PCL", "Polymer"] under the "Expression + Type" criterion.

Table 2 NER Results by GPT-4o.

Prompt	Criterion	P(%)	R(%)	F1(%)
1	Expression	85.20	58.10	69.09
	Expression+Type	84.59	57.69	68.60
2	Expression	88.73	67.02	76.36
	Expression+Type	88.25	66.65	75.94
Combined	Expression	83.86	76.94	80.25
	Expression+Type	83.30	76.56	79.78

Table 3 Recalls(%) of nested and non-nested entities by GPT-4o.

Prompt	Criterion	Non-nested	Nested	△R
1	Expression	78.51	14.95	63.56
	Expression+Type	77.93	14.93	63.00
2	Expression	72.24	56.21	16.03
	Expression+Type	71.77	56.09	15.68

Gold entities can be categorized as nested or non-nested. Nested entities are fully contained within other entities, while non-nested entities represent all other gold entities. For instance, in the span *"Chromobacterium viscosum"*, *"Chromobacterium"* is a nested entity, and *"Chromobacterium viscosum"* is a non-nested entity. Recall was reported separately for each category to facilitate comparative analysis.

4 Results

As presented in Table 2, GPT-4o performs adequately when tasked with identifying entity expressions, without considering entity types or positional information. It also demonstrates strong performance in identifying both entity expressions and types.

Huge performance gap between nested and non-nested entities. We report the recalls of nested and non-nested entities in Table 3. When we provide no examples of nested entities, the recall of nested entities is very low (14.95% and 14.93%), much lower than the performance on non-nested entities (78.51% and 77.93%). Despite the Prompt 1 requests predictions for nested entities (*"identify all named entities ... including nested entities"*), GPT-4o still misses most of nested entities. Providing examples in the prompt successfully enhances performance on nested entities, reducing the gap to non-nested entities. However, the recall on non-nested entities decreased from 78.51% to 72.24% when we only ask expressions are aligned with gold entities, indicating that examples obviously brought

Table 4 Results for various entity types were obtained using the "Expression+Type" criterion, combining predictions obtained by two prompts.

Entity Type	P(%)	R(%)	F1(%)
Polymer	80.33	74.20	77.14
Bacterium	87.33	82.46	84.82
Fungus	88.91	83.65	86.20
Enzyme	77.29	67.32	71.96

Table 5 NER results when an entity is correct if it matches the expression, type and location of a gold entity by GPT-4o.

Prompt	P(%)	R(%)	F1(%)
1	4.96	3.27	3.94
2	4.97	3.68	4.23

some negative effects.

Combined the predictions of two prompts. Given that Prompt 1 is effective for non-nested entities and Prompt 2 excels with nested entities, and both prompts demonstrate high precision, we combine their results to improve recall. The combined outcomes are shown in Table 2, resulting in an F1 score of approximately 80%. In the zero-shot scenario, these results are promising, indicating that automatic extraction of polymer biodegradability information is feasible. This sets the stage for further research on event extraction with robust NER performance.

Small differences between two evaluation criteria. Entity expression and type prediction marginally reduces precision and recall compared to scenarios focusing solely on expression prediction, irrespective of the prompt or whether the gold entities are nested or non-nested. As expected, an LLM leverages internal knowledge to perform well in entity type prediction.

Enzyme and Polymer entities v.s. Bacterium and Fungus entities. GPT-4o demonstrated strong performance in recognizing bacterium and fungus entities (F1 scores > 80%) but showed suboptimal results for enzyme and polymer entities (see Table 4). We analyzed generated outputs and found the potential factors to bring about this phenomenon. Scientific papers typically refer to bacteria and fungi using their scientific names (e.g., Bacillus subtilis) or abbreviated forms (e.g., B. subtilis), facilitating recognition by LLMs if these names are present in the pretrained data. Many false negatives arise from incomplete scientific names. For instance, while GPT-4o successfully recognizes "B. subtilis strain MZA-75", it fails to identify "strain MZA-75" when the species name is omitted. Similarly, sample identifiers (e.g., "H-237") often represent specific bacterial or fungal types, but without sufficient contextual information, they are hard to recognize. On the contrary, the poor performance in enzyme recognition is attributed to two factors. First, general enzyme terms, such as "dehydrogenase", "esterase", and "oxidase" are frequently overlooked. Second, gene-enzyme names are misclassified; for example, "pueA" gene, which encodes the "PueA" enzyme, is incorrectly predicted as an enzyme. Polymer entity recognition exhibits similar challenges. Our future work will extend NER from sentence-level to paragraph-level contexts and incorporate targeted examples in few-shot learning scenarios. These enhancements are expected to improve performance across categories.

Unable to provided precise entity offsets. As shown in Table 5, most predicted positions of entities do not correspond to the actual locations of the entities, even though the expressions and types are correct. This discrepancy is widespread, with only 546 of 10,111 predicted positions (5.4%) accurately corresponding to the predicted expressions. Although our task does not require the location of the entity, it is important to highlight the limitations of GPT-4 in this context. Researchers should take appropriate precautions when employing LLM-based methods for NER tasks that necessitate location information.

What about GPT-4o-mini? Given the higher cost and time requirements of GPT-4o, we provide experimental results achieved by GPT-4o-mini in the Appendix A for reference. These results indicate a significant performance gap on nested-entities between GPT-4o and GPT-4o-mini, with the latter proving insufficient for our task.

5 Conclusion

In this study, we introduced PolyBD, a dataset specifically curated for extracting polymer biodegradability information, and conducted sentence-level named entity recognition using GPT-4o in a zero-shot setting, aiming to advance research and practical applications in this domain. Our findings indicated that GPT-4o demonstrates robust internal knowledge for identifying bacterium and fungus entities but shows limitations in recognizing polymer and enzyme entities involved in polymer biodegradation. Furthermore, a notable performance disparity is observed between non-nested and nested entities, underscoring the need for further investigation.

References

[1] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: a survey. **Frontiers of Computer Science**, Vol. 18, No. 6, p. 186357, Nov 2024.

[2] Eugenio Cesario, Carmela Comito, and Ester Zumpano. A survey of the recent trends in deep learning for literature based discovery in the biomedical domain. **Neurocomputing**, Vol. 568, p. 127079, 2024.

[3] Alain C. Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H. Nair, Philippe Schwaller, and Teodoro Laino. Automated extraction of chemical synthesis actions from experimental procedures. **Nature Communications**, Vol. 11, No. 3601, 2020. https://doi.org/10.1038/s41467-020-17266-6.

[4] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, editors, **Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)**, pp. 546–555, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[5] Meishan Zhang, Bin Wang, Hao Fei, and Min Zhang. In-context learning for few-shot nested named entity recognition. In **ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 10026–10030, 2024.

Table 6 NER Results by GPT-4o and GPT-4o-mini.

Prompt	Criterion	GPT-4o			GPT-4o-mini		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
1	Expression	85.20	58.10	69.09	79.29	60.71	68.77
	Expression+Type	84.59	57.69	68.60	77.21	59.11	66.96
2	Expression	88.73	67.02	76.36	85.02	58.01	68.97
	Expression+Type	88.25	66.65	75.94	83.23	56.89	67.59
Combined	Expression	83.86	76.94	80.25	77.40	71.85	74.52
	Expression+Type	83.30	76.56	79.78	75.02	70.54	72.71

Table 7 Recalls(%) of nested and non-nested entities by GPT-4o and GPT-4o-mini.

Prompt	Criterion	GPT-4o			GPT-4o-mini		
		Non-nested	Nested	△R	Non-nested	Nested	△R
1	Expression	78.51	14.95	63.56	81.92	16.01	65.91
	Expression+Type	77.93	14.93	63.00	79.73	15.76	63.97
2	Expression	72.24	56.21	16.03	68.45	36.37	32.08
	Expression+Type	71.77	56.09	15.68	67.18	35.63	31.55

A NER Results by GPT-4o-mini

In this section, we present the experimental results by GPT-4o-mini in Table 6 and Table 7, obtained under the same settings as GPT-4o.

When prompted with Prompt 1, GPT-4o-mini demonstrated higher recall than GPT-4o for both nested and non-nested entities. However, unlike the huge improvement observed for nested entities and slightly worse performance on non-nested entities when transitioning from Prompt 1 to Prompt 2, GPT-4o-mini exhibited a pronounced decline in recall for non-nested entities, significantly impacted by examples of nested entities added to the prompt. Combining predictions from the two prompts resulted in precision falling below 80%, which is suboptimal for tasks requiring high factual accuracy.