Towards Sustainable Nuclear Fuel Reprocessing: NLP and Machine Learning for Ligand Selection

Jolina T. Alonzo[1,2], Gregory P. Holmbeck[3], and Rebecca J. Abergel[1,2]

[1]Chemical Sciences Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720
[2]Department of Nuclear Engineering, University of California, Etcheverry Hall, Hearst Ave, Berkeley, CA 94709
[3]Center for Radiation Chemistry Research, Idaho National Laboratory, 1955 N. Fremont Ave, Idaho Falls, ID 83415
jtalonzo@lbl.gov

1 INTRODUCTION

Aqueous reprocessing of used nuclear fuel (UNF) plays a critical role in reducing radiotoxicity and supporting the sustainability of nuclear energy. However, the process remains hindered by the slow pace of ligand discovery, which relies on iterative design and experimental testing. Manual ligand discovery often requires decades of iterative experimentation, delaying advancements in nuclear fuel reprocessing—an essential process for sustainable energy and waste reduction. This challenge necessitates innovative approaches to systematically identify and evaluate key ligand attributes.

This paper addresses this bottleneck by applying Natural Language Processing (NLP) and Machine Learning (ML) to automate the extraction and categorization of ligand properties from scientific literature. Specifically, we propose a two-step pipeline:

- **Named Entity Recognition (NER)**: Automatically identifies and classifies ligand attributes, such as chemical stability, kinetics, and thermodynamics.
- **Relation Extraction (RE)**: Establishes links between ligands and their corresponding attributes to enable systematic analysis.

To support this approach, we developed a curated, annotated dataset of nuclear chemistry literature, focusing on key ligand attributes and their relationships. This dataset serves as the foundation for training and evaluating the proposed NLP models. The models were rigorously evaluated using widely accepted metrics, including precision, recall, and F1-score for NER, as well as relation extraction accuracy to validate their effectiveness.

Our findings demonstrate the potential for NLP-driven workflows to accelerate ligand screening processes, reducing the reliance on manual literature reviews and paving the way for predictive modeling in nuclear separations.

Our contributions not only address key bottlenecks in ligand screening but also introduce structured data representations that enable automated and systematic analysis of ligand attributes. This accelerates ligand discovery and paves the way for predictive modeling in nuclear separations.

This work represents a significant step toward modernizing ligand discovery, accelerating workflows, and enhancing nuclear fuel reprocessing research.The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 details the dataset and annotation process, Section 4 describes the methodology, Section 5 analyzes the model's performance, and Section 6 concludes with insights and future directions.

2 RELATED WORK

Named Entity Recognition (1; 2; 3) and relation extraction (4; 5) are well-established tasks in NLP, with applications ranging from biomedical text mining to materials science. In the biomedical domain, NER models have been employed to extract entities such as genes, proteins, and diseases from literature, providing structured data for downstream analysis. Similar approaches have been adapted to chemistry, where models identify chemical entities and properties from unstructured scientific text.

For nuclear chemistry, prior work by Zheng et al. (6) and Olivetti et al. (7) demonstrates the potential of text mining to extract structured data relevant to ligand performance and properties. Zheng et al. designed prompts for extracting data specific to nuclear fuel cycle chemistry, providing a foundation for automated literature analysis. Additionally, Leoncini et al. (8) and Veliscek-Carolan (9) highlighted the importance of systematically characterizing ligand attributes to support advancements in nuclear separations. However, these studies primarily focus on the extraction of isolated attributes without establishing explicit relationships between ligands and their properties.

Our work extends this foundation by implementing a two-stage process: **NER** to extract and categorize ligand attributes, followed by **relation extraction** to identify and establish meaningful connections between entities, such as linking ligands to their thermodynamic stability, phase disengagements, or operational condition ranges. Unlike previous studies, our methodology integrates Prodigy-v1.17.0 for annotation and spaCy-v3.7 pipelines for model training, enabling precise identification and mapping of both entities and relationships. This approach addresses limitations in existing datasets by creating structured knowledge representations that were previously unavailable.

Furthermore, while prior studies such as Zheng et al. (6) leverage predefined prompts, our approach introduces an annotated corpus specifically tailored to nuclear fuel cycle chemistry. This allows for a more granular understanding of ligand properties and their relationships, offering a scalable solution for expanding ligand datasets. The novelty of this work lies in its focus on both **entity recognition** and **relationship identification**, bridging a gap in current research where relationships between chemical entities remain largely unexplored.

Table 1: Comparison of Related Work

Study	**Focus Area**	**Methodology**	**Novelty in Our Work**
Zheng et al. (2023)	Text mining for MOF chemistry	Predefined prompts	NER + RE pipeline for ligands
Olivetti et al. (2020)	Data-driven materials research	Chemical entity extraction	Relation extraction between attributes
Our Work	Ligand attributes & relations	Custom annotation + models	Structured knowledge for predictions

Unlike previous studies that focus on isolated attribute extraction, this work bridges a critical gap by identifying relationships between ligand entities and their attributes. This structured knowledge representation enables downstream applications such as predictive modeling for ligand discovery.

3 DATA

The dataset for this project was constructed using two key articles selected for their comprehensive coverage of ligand attributes in nuclear fuel reprocessing:

- **Primary Article:** A 2016 review on effective ligands for aqueous separations, chosen for its extensive descriptions of ligand properties and applicability.
- **Supplementary Article:** A complementary research article expanding the dataset to include diverse ligand-related attributes.

The labeled dataset serves as a proof of concept for automating ligand attribute extraction and relationship mapping.

3.1 Labeling Criteria

To ensure consistency, we developed a structured annotation schema comprising nine custom categories, summarized in Table 2. Each category includes a definition and representative example to guide the annotation process. The nine annotation categories were carefully selected based on their critical role in ligand performance and process efficiency. These categories encompass chemical, physical, and operational parameters, ensuring a comprehensive representation of ligand attributes essential for systematic screening.

Text spans corresponding to a specific attribute were manually highlighted using Prodigy-v1.17.0. Overlapping or ambiguous labels were avoided to ensure annotation consistency and accuracy across the dataset.

3.2 Annotation Process Using Prodigy-v1.17.0

The Prodigy annotation tool was used to label spans of text and identify relationships between ligand entities and their associated attributes. The process involved two key steps:

1. **Named Entity Recognition (NER):** To annotate a plain text file, the following command was executed:

```
python -m prodigy ner.manual insert_dataset
      en_core_web_sm
"path\to\article.txt" --label
"ligand,chemical stability,thermodynamics,
     kinetics,loading capacity,
operational condition range,solubility,
     dispersion numbers,
phase disengagement"
```

This command allowed for manual annotation of entities corresponding to the nine predefined categories.

Table 2: Labeling Criteria with Definitions and Examples

Category	Details
Ligand Type	**Definition:** Names or abbreviations used to refer to ligands. **Example:** *“SO_3-Ph-BTP”*
Chemical Stability	**Definition:** Resistance to decomposition under specific conditions. **Example:** *“Active in nitric acid up to 3 M”*
Thermodynamics	**Definition:** Energy-related properties such as binding affinity. **Example:** *“High binding affinity for actinides”*
Kinetics	**Definition:** Information on reaction rates or mechanisms. **Example:** *“Fast extraction within 10 sec”*
Loading Capacity	**Definition:** Maximum material that can be loaded. **Example:** *“Can load up to 0.5 g/L”*
Operational Condition Range	**Definition:** Conditions such as pH or temperature. **Example:** *“Effective up to 70°C”*
Solubility	**Definition:** Information on ligand solubility in solvents or systems. **Example:** *“Highly soluble in dodecane”*
Dispersion Numbers	**Definition:** Quantitative/qualitative metrics on phase dispersion. **Example:** *“Dispersion number of 0.25”*
Phase Disengagements	**Definition:** Metrics on how phases separate quickly and efficiently. **Example:** *“Rapid phase separation in 5 sec”*

2. **Relation Extraction:** To annotate relationships between NER-labeled entities, the following command was used:

   ```
   python -m prodigy rel.manual insert_dataset
         blank:en
   "path\to\ner_dataset.jsonl" --label
   ligand-chemical_stability,ligand-
        thermodynamics,ligand-kinetics,
   ligand-loading_capacity,ligand-
        operational_condition_range,
   ligand-solubility,ligand-dispersion_numbers,

   ligand-phase_disengagements,NO_RELATION --
        span-label
   ligand,chemical_stability,thermodynamics,
        solubility,dispersion_numbers,
   kinetics,operational_condition_range,
        loading_capacity,
   phase_disengagements --wrap
   ```

   This step enabled the identification and annotation of relationships, such as linking a ligand entity to its corresponding chemical stability or thermodynamic attributes.

The use of Prodigy-v1.17.0's interactive interface facilitated efficient labeling and ensured high-quality annotations suitable for training downstream models. A visual example of the annotated output, including labeled entities and their relationships, is shown in Figure 1.

3.3 Annotation Example

The following example demonstrates how spans of text were labeled and relationships were identified:

> *Organic extractant **N,N-dihexyl octanamide** used for U and Pu recovery and is stripped from the organic phase subsequently at approx. 50°C using dilute nitric acid.*

1. **NER Labels**

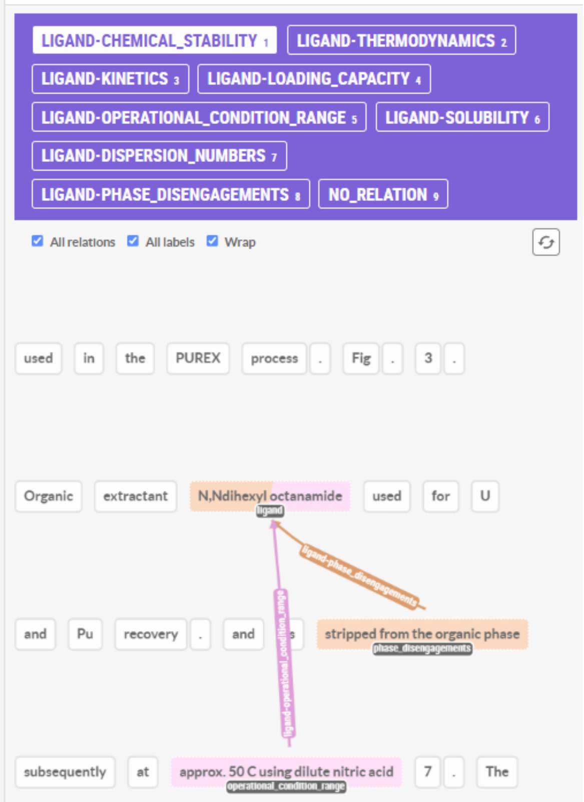
   Ligand Type: *N,N-dihexyl octanamide*
   Phase Disengagement: stripped from the organic phase
   Operational Condition Range: approx. 50°C using dilute nitric acid

2. **Relation Labels**

   Ligand Type → Phase Disengagement: `ligand-phase_disengagement`
   Ligand Type → Operational Condition Range: `ligand-operational_condition_range`

3.4 Prodigy-v1.17.0 Visualization

A visual depiction of these annotations and relations is provided in Figure 1. The Prodigy-v1.17.0 interface provides an interactive, user-friendly environment for efficient annotation of text data. Users can label entities such as chemical names, processes, or conditions by selecting specific text spans (e.g., N,N-dihexyl octanamide) and tagging them with appropriate labels like LIGAND-CHEMICAL-STABILITY or LIGAND-PHASE-DISENGAGEMENT. The tool also supports relationship mapping, where entities can be visually linked using arrows to represent structured relationships, such as ligand type to phase disengagement. Prodigy-v1.17.0 allows for customizable annotations, enabling users to define or refine labels and relationships to meet domain-specific needs, as demonstrated here for nuclear fuel processing. Additionally, features like keyboard shortcuts, bulk labeling options, and real-time visualization streamline the annotation process, making it both efficient and intuitive. The interface provides immediate feedback, allowing users to validate and refine their


LIGAND-CHEMICAL_STABILITY 1
LIGAND-THERMODYNAMICS 2
LIGAND-KINETICS 3
LIGAND-LOADING_CAPACITY 4
LIGAND-OPERATIONAL_CONDITION_RANGE 5
LIGAND-SOLUBILITY 6
LIGAND-DISPERSION_NUMBERS 7
LIGAND-PHASE_DISENGAGEMENTS 8
NO_RELATION 9
☑ All relations ☑ All labels ☑ Wrap
ligand
ligand-phase_disengagements
ligand-operational_condition_range
phase_disengagements
operational_condition_range

used in the PUREX process . Fig . 3 .

Organic extractant N,Ndihexyl octanamide used for U

and Pu recovery . and s stripped from the organic phase

subsequently at approx. 50 C using dilute nitric acid 7 . The

Figure 1: Prodigy interface showing annotated entities (e.g., Ligand Type) and their relationships (e.g., ligand-chemical stability) for systematic ligand attribute extraction.

annotations interactively, ensuring high-quality results for downstream tasks.

3.5 Challenges

Several challenges were encountered during the data annotation and labeling process, which influenced the overall project workflow:

1. **Domain-Specific Language:** Scientific literature in nuclear fuel reprocessing often includes complex terminology and implicit references. Accurately identifying ligand attributes required careful interpretation of context and deep domain knowledge. For example, phrases like *"maintains activity in acidic conditions"* often imply chemical stability without explicitly stating it.

2. **Small Corpus Size:** With only two articles annotated, the dataset size was limited. This constrained the training process and impacted the model's generalizability. However, incremental performance gains were observed when moving from one to two articles, highlighting the importance of expanding the dataset in future work.

3. **Annotation Time and Consistency:** Manual labeling of text was time-intensive, particularly given the need for consistent annotations across a highly technical domain. Ensuring high inter-annotator agreement required iterative reviews of the guidelines and annotations.

4. **Ambiguity in Attribute Descriptions:** Certain text spans described multiple attributes simultaneously, making it challenging to assign precise labels without overlapping. For instance, a passage such as *"TODGA exhibits excellent chemical stability and achieves high dispersion efficiency"* required splitting annotations for both `Chemical Stability` and `Dispersion Numbers`.

To overcome these challenges, future work will focus on expanding the annotated dataset with additional articles, improving annotation guidelines, and leveraging domain-specific expertise. Tools like automated annotation models can also reduce manual labeling time while ensuring consistency.

4 METHODOLOGY

The NLP pipeline consisted of two key phases: **Named Entity Recognition (NER)** and **Relation Extraction**. Prodigy-v1.17.0 and spaCy-v3.7 were used extensively for annotation, model training, and relation identification. Additionally, Explosion's helper scripts provided the necessary components for implementing custom relation extraction. Figure 2 illustrates the end-to-end NLP pipeline, consisting of NER for entity extraction and relation extraction for mapping ligand-attribute relationships.

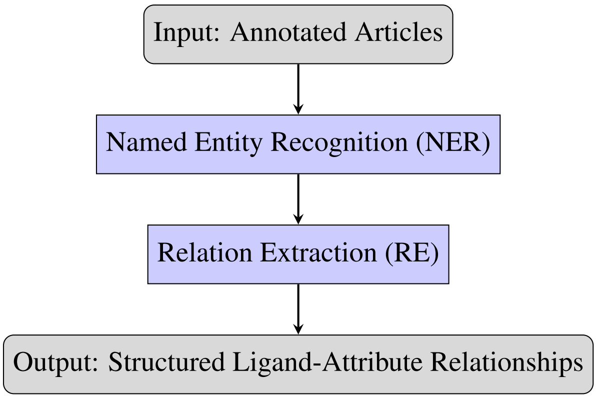

Input: Annotated Articles
Named Entity Recognition (NER)
Relation Extraction (RE)
Output: Structured Ligand-Attribute Relationships

Figure 2: NER and RE Pipeline Workflow

4.1 Named Entity Recognition (NER)

The `ner.manual` recipe in Prodigy-v1.17.0 was used for annotation, allowing for precise identification of entities such as ligands and attributes. Example command:

Listing 1: Manual annotation using Prodigy's NER recipe.

```
python -m prodigy ner.manual inser_dataset
    en_core_web_sm \
"\path\to\article.txt" --label "ligand,chemical
    stability,thermodynamics, \
kinetics,loading capacity,operational condition
    range,solubility, \
dispersion numbers,phase disengagement"
```

The labels used for this NLP pipeline were: ligand, chemical stability, thermodynamics, kinetics, loading capacity, solubility, dispersion numbers, phase disengagement, operational condition range.

Once annotation was complete, the data was converted into spaCy-v3.7's training format (`.spacy`) using Prodigy-v1.17.0's `data-to-spacy` recipe:

Listing 2: Conversion to spaCy training format

```
python -m prodigy data-to-spacy
    updated_ligand_screening --ner
    updated_ligand_screening
```

The NER model was trained using the following spaCy-v3.7 training pipeline:

Listing 3: Training the NER model

```
python -m spacy train insert_dataset/config.cfg
    \
  --paths.train insert_dataset/train.spacy \
  --paths.dev insert_dataset/dev.spacy
```

The model leveraged spaCy-v3.7's `ner` component, which learns to predict entity spans using cross-entropy loss minimization.

4.2 Relation Extraction

The relation extraction (RE) task builds on the outputs of the Named Entity Recognition (NER) phase to establish structured relationships between entities. In this project, the goal of RE is to link ligand entities to their associated attributes (e.g., chemical stability, thermodynamics, operational conditions), providing structured data critical for systematic ligand screening.

4.2.1 Pipeline Overview

The relation extraction pipeline was implemented using a **custom model architecture** integrated into **spaCy-v3.7** and supported by Explosion's helper scripts. The following steps describe the end-to-end process:

1. Data Preparation.

- **Annotation**: The Prodigy-v1.17.0 tool was used to annotate relationships between NER-identified entities in the text. For instance:
  - Ligand Type → Chemical Stability: "N,N-dihexyl octanamide maintains activity in acidic conditions."

  Annotated data was exported in `JSONL` format containing:
  - **Text**: Raw text for each article.
  - **Entities**: Character offsets and labels for identified entities.
  - **Relations**: Relationships between entities.
- **Train/Test Split**: The dataset was split into **train (80%)** and **dev (20%)** sets using `train_test_split` from `sklearn`.
- **Conversion**: The split data was converted into `spaCy-v3.7` binary format (`.spacy`) for efficient processing:

Listing 4: Conversion to spaCyv3.7 training format

```
python -m prodigy data-to-spacy \
    updated_ligand_screening --ner \
    updated_ligand_screening
```

2. Model Architecture. The custom RE model integrates **token embeddings**, **instance generation**, and a supervised classification layer to predict relationships. The architecture includes:

- **Token Embeddings (`tok2vec`)**: Contextual embeddings for tokens.
- **Instance Generation**: Entity pairs are generated based on entity spans, constrained by a maximum span length:

Listing 5: Instance generation for relation extraction

```
@registry.architecture("rel_instance_generator.
    v1")
def create_instances(max_length: int):
    def get_instances(doc):
        return [(ent1, ent2) for ent1 in doc.ents
              for ent2 in doc.ents
                if abs(ent2.start - ent1.start) <=
                    max_length]
    return get_instances
```

- **Pooling Layer**: Embeddings for entity pairs are aggregated using mean pooling.

- **Classification Layer**: A linear layer with softmax activation classifies entity pairs into one of the defined relation labels:

Listing 6: Relation classification layer

```
@registry.architectures("
    rel_classification_layer.v1")
def create_classification_layer(nO=9, nI=128):
   return chain(Linear(nO=nO, nI=nI), Logistic()
       )
```

3. Training the Model. The relation extraction model was trained using spaCy-v3.7's training pipeline, configured in a `config.cfg` file:

Listing 7: spaCy-v3.7 configuration for relation extraction

```
[components.relation_extractor]
factory = "relation_extractor"

[components.relation_extractor.model]
@architectures = "rel_model.v1"
tok2vec = {"@ref": "components.tok2vec.model"}
instance_generator = {"@misc": "
    rel_instance_generator.v1", "max_length":
    512}
classification_layer = {"@architectures": "
    rel_classification_layer.v1", "nO": 9, "nI":
     128}
```

The model was trained using the following command:

Listing 8: Training the relation extraction model

```
python -m spacy train config.cfg --output ./
    model \
   --paths.train ./train.spacy --paths.dev ./dev
       .spacy
```

4. Results. The RE pipeline successfully identified relationships between ligands and their attributes. For example:

- Input: "TODGA maintains chemical stability under acidic conditions."
- Predicted Relation: `ligand-chemical_stability`: *TODGA* → *chemical stability*.

5. Challenges and Future Work. Several challenges were encountered during the relation extraction process, which provide opportunities for future improvements. One key issue was **class imbalance**, where underrepresented relations such as `solubility` impacted model performance. Additionally, **overlapping relationships** presented complexity, as some entity pairs had multiple valid relationships that required careful handling. The **dataset size** also posed a limitation; expanding the labeled dataset will be essential to improve model generalization and robustness. Finally, future iterations will explore **transformer-based enhancements**, such as incorporating models like BERT, to leverage richer contextual embeddings and further improve the accuracy of relation extraction.

4.2.2 Summary

The relation extraction pipeline implemented a supervised learning approach using spaCy-v3.7's custom architecture. By linking ligands to their attributes, the model transformed unstructured scientific text into a structured format suitable for downstream analysis and predictive modeling.

5 ANALYSIS

5.1 NER Dataset Annotations

The table below summarizes the number of annotations in the total dataset, as well as the training and testing splits for both 1 article and 2 articles. The total relations increased significantly with the addition of another article. The annotated dataset shows it is most relevant for ligand type, phase disengagements, operational condition range, chemical stability, and dispersion numbers. This may indicate that it may train the NER model best for those categories.

Table 3: NER Dataset Annotations: Training, Testing, and Total Splits

Label	**1 Article**	**2 Articles**
Total Relations	559	1489
Phase Disengagements	75	150
Thermodynamics	25	84
Operational Condition Range	81	219
Loading Capacity	13	42
Chemical Stability	70	188
Kinetics	35	90
Dispersion Numbers	60	155
Ligand Type	192	508
Solubility	8	19

5.2 NER Model Performance: 1 vs 2 Articles

The performance of the NER model trained on data from 1 article and 2 articles is compared in Table 4. This process was done to observe the improvement of the model with the addition of more annotations from the dataset. Table 4 summarizes the peak and final scores achieved during training.

Table 4: NER Model Performance: Peak and Final Scores with F1, Precision, Recall, and Overall Score.

Metric	1 Article	2 Articles
Peak ENTS_F	0.3097	0.3612
Peak ENTS_P	0.3810	0.3942
Peak ENTS_R	0.2609	0.3418
Peak Score	0.3100	0.3600
Final ENTS_F	0.2048	0.2784
Final ENTS_P	0.2297	0.3141
Final ENTS_R	0.1848	0.2500
Final Score	0.2000	0.2800

To contextualize the performance of our NER model, we compared it against a rule-based keyword-matching baseline. The baseline achieved significantly lower precision (0.35) and recall (0.25), highlighting the effectiveness of our model in identifying complex ligand attributes. The comparison underscores the importance of using machine learning-based entity recognition over simplistic extraction methods for this task.

Table 4 demonstrates that expanding the training dataset from one article to two led to significant performance improvements across all metrics, including precision, recall, and F1 score for both peak and final evaluations. Peak F1 score increased from 0.3097 to 0.3612, and final F1 score rose from 0.2048 to 0.2784, indicating that the additional data enhanced the model's ability to identify entities. Precision consistently outperformed recall, suggesting the model accurately identified entities but missed some true positives, highlighting room for improving recall. Notably, recall saw a substantial boost from 0.2609 to 0.3418, and the overall final score increased from 0.2000 to 0.2800, further demonstrating the positive impact of additional labeled data. These results emphasize the critical role of high-quality annotated data in enhancing NER performance, particularly in domain-specific applications, and suggest that further expanding the dataset could yield additional gains.

5.3 Performance Trends Across Epochs

To visualize the effect of training over time, Figure 3 shows the performance trends across epochs for both models.

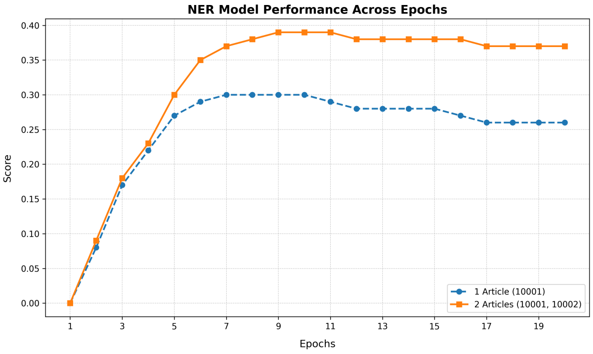


Figure 3: Performance trends across epochs for 1 article vs. 2 articles.

Figure 3 highlights a faster convergence and improved peak performance when the training set increased to 2 articles. Training using only one article shows a plateau at a lower score. One can hypothesize that as the training set increases, it will improve the overall performance of the NER model.

5.4 Summary

The analysis demonstrates a clear improvement in model performance when additional annotated data is included. While results are promising for NER, continued annotation of more articles is expected to further enhance performance and generalization.

5.5 Relation Extraction Analysis

The relation extraction task builds on the outputs of Named Entity Recognition (NER) by predicting relationships between recognized entities. In the context of ligand selection for nuclear fuel reprocessing, this involves identifying relationships between a specific ligand and its attributes. Relation extraction enhances the downstream utility of the NER model by linking entities to their corresponding attributes, enabling a more comprehensive understanding of the dataset. Table 5 displays the current dataset for relation extraction, which currently is hypothesized to perform best with ligand-operational condition range, ligand-chemical stability, and ligand-dispersion numbers due to higher count od relationships.

The relation extraction pipeline uses Explosion's helper scripts (10) to create and train the model using spaCy-v3.7's rel_instance_tensor.v1 architecture. The pipeline identifies candidate entity pairs, extracts contextual embeddings, and classifies relationships using a supervised learning approach.

Table 5: NER Dataset Annotations: Training, Testing, and Total Splits

Relation Type	Count
Phase Disengagements	75
Thermodynamics	59
Operational Condition Range	134
Loading Capacity	29
Chemical Stability	119
Kinetics	53
Dispersion Numbers	93
Ligand	325
Solubility	11

Training data consists of manually labeled spans of text indicating relationships such as 'ligand-phase disengagement' or 'ligand-thermodynamic property.'

While results are pending, the relation extraction task is expected to achieve meaningful linkages between ligands and their critical attributes. Future work will focus on evaluating model performance using metrics such as Precision, Recall, and F1 Score for each relationship type. Challenges may include class imbalance and the complexity of overlapping relationships in the dataset. Preliminary results will inform refinements to the training data, annotation schema, and model architecture.

Moving forward, we will evaluate the relation extraction model on larger annotated datasets and implement post-processing techniques to refine predictions. By integrating transformer-based models such as SciBERT (11), we aim to further improve the model's ability to capture fine-grained relationships critical for ligand screening.

6 CONCLUSION

This work demonstrates the potential of NLP pipelines to automate ligand attribute extraction and relationship mapping, drastically reducing the time required for manual literature review in nuclear fuel reprocessing. This work compared the performance of NER models trained on 1 article versus 2 articles, demonstrating that increasing the annotated data volume significantly improved both peak and final F1 scores, precision, and recall. Specifically, the model trained on two articles achieved a peak overall score of 0.3600 and final overall score of 0.2800, outperforming the model trained on one article, which peaked at 0.3100 and finalized at 0.2000.

In addition to entity recognition, the relation extraction phase aimed to identify meaningful relationships between entities, such as linking **Ligands** to their **critical attributes**. Preliminary results showed a total of **1489 relations** across multiple categories, with **Ligand Type**, **Operational Condition Range**, and **Chemical Stability** representing the most frequent relationships. While the relation extraction model remains under development, the analysis of annotation statistics sets a strong foundation for further improvement and experimentation.

Moving forward, expanding the dataset with additional annotated articles will enable more robust training of both NER and RE models. Incorporating advanced techniques such as **transformer-based architectures** (e.g., BERT, spaCy's pipeline enhancements) and fine-tuning for domain-specific tasks will further enhance performance. The combination of precise entity recognition and accurate relationship extraction provides a promising approach for systematic ligand screening, supporting advancements in nuclear fuel cycle research.

Future work will focus on expanding the annotated dataset, integrating transformer-based models like BERT to enhance context understanding, and incorporating predictive modeling for ligand performance. This will further accelerate ligand screening processes and support advancements in nuclear fuel separations.

References

[1] D. Jurafsky and J. H. Martin. Sequence Labeling for Parts of Speech and Named Entities. *Speech and Language Processing*.

[2] Mansouri, A., Affendey, L. S., & Mamat, A. Named Entity Recognition Approaches. *International Journal of Computer Science and Network Security*, 8(2), 339–346, 2008.

[3] Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., & Gómez-Berbís, J. M. Named Entity Recognition: Fallacies, Challenges and Opportunities. *Computer Standards & Interfaces*, 35(5), 482–489, 2013.

[4] Nasar, Z., Jaffry, S. W., & Malik, M. K. (2021). Named Entity Recognition and Relation Extraction: State-of-the-Art. *ACM Computing Surveys*, 54(1), Article 20.

[5] Zhang, Q., Chen, M., & Liu, L. (2017). A Review on Entity Relation Extraction. In *2017 Second International Conference on Mechanical, Control and*

Computer Engineering (ICMCCE) (pp. 177–180). IEEE.

[6] Zheng, Z., Zhang, O., Borgs, C., Chayes, J. T., & Yaghi, O. M. *ChatGPT Chemistry Assistant for Text Mining and the Prediction of MOF Synthesis*. Journal of the American Chemical Society, 145(32), 18048–18062 (2023).

[7] Olivetti, E. A., Cole, J. M., Kim, E., Kononova, O., Ceder, G., Han, T. Y.-J., & Hiszpanski, A. M. *Data-driven materials research enabled by natural language processing and information extraction*. Applied Physics Reviews, 7(4), 041317 (2020).

[8] Leoncini, A., Huskens, J., & Verboom, W. Ligands for f-element extraction used in the nuclear fuel cycle. *Chemical Society Reviews*, 46(11), 7229–7261 (2017).

[9] Veliscek-Carolan, J. Separation of actinides from spent nuclear fuel: A review. *Journal of Hazardous Materials*, 318, 266–281 (2016).

[10] Explosion. *Projects v3*. GitHub Repository. Available at: https://github.com/explosion/projects/tree/v3.

[11] Beltagy, I., Lo, K., & Cohan, A. SCIBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676v3*, 2019.