A Unified Framework for N-ary Property Information Extraction in Materials Science

Van-Thuy Phi and Yuji Matsumoto

Center for Advanced Intelligence Project, RIKEN
{thuy.phi, yuji.matsumoto}@riken.jp

Abstract

This paper presents a unified framework for extracting n-ary property information from materials science literature, addressing the critical challenge of capturing complex relationships that often span multiple sentences. We introduce three complementary approaches: RE-Composition, which transforms binary relations into n-ary structures; Direct EAE, which models polymer properties as events with multiple arguments; and LLM-Guided Assembly, which leverages high-confidence entity and relation outputs to guide structured extraction. Our framework is built upon two novel resources: MatSciNERE, a comprehensive corpus for materials science entities and relations, and PolyEE, a specialized corpus for polymer property events. Through strategic synthetic data generation for both NER and EAE tasks, we achieve significant performance improvements (up to 5.34 F1 points). Experiments demonstrate that our combined approaches outperform any single method, with the LLM-guided approach achieving the highest F1 score (71.53%). The framework enables more comprehensive knowledge extraction from scientific literature, supporting materials discovery and database curation applications. We plan to release our resources and trained models to the research community.

1 Introduction

Materials science encompasses diverse entities ranging from broad categories like organic and inorganic materials to specialized subcategories such as polymers. The exponential growth of scientific publications in this field creates significant challenges for researchers attempting to efficiently access and organize critical information. Particularly crucial is the extraction of property information, defined as structured knowledge about materials and their properties, measured values, experimental conditions, and characterization methods. This information is essential for materials discovery, database curation, and accelerating research.

Extracting property information from materials science literature presents unique challenges compared to traditional information extraction (IE) tasks. Property information typically involves complex n-ary relationships connecting multiple entities (e.g., a polymer with a specific property value under certain conditions, measured by a particular method). These relationships frequently span multiple sentences, requiring cross-sentence reasoning. Additionally, specialized terminology, diverse experimental contexts, and varied writing styles further complicate automated extraction.

Previous materials information extraction approaches have focused primarily on Named Entity Recognition (NER) to identify materials (Weston et al., 2019; Shetty et al., 2023) or binary Relation Extraction (RE) between entity pairs (Phi et al., 2024). While promising for specific subtasks, these methods fail to capture the n-ary property information essential for comprehensive materials knowledge bases. Recent Event Argument Extraction (EAE) advancements offer pathways for modeling n-ary relationships, but their application to materials science remains limited. Large Language Models (LLMs) have demonstrated capabilities in extracting structured information, yet their performance on specialized scientific content is inconsistent without appropriate guidance (Kumar et al., 2025).

In this work, we introduce a unified framework for n-ary property IE in materials science that integrates three complementary approaches: (1) RE-Composition, which transforms binary relationship predictions into n-ary structures; (2) Direct EAE, which models property information as

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2369–2388

events with property names as triggers and other elements as arguments; and (3) LLM-Guided Assembly, a novel hybrid approach that leverages high-confidence entity and relation predictions to guide LLMs in generating complete n-ary structures.

The main contributions of this paper are:

- A unified framework integrating three distinct approaches for n-ary property IE, enabling comprehensive coverage and comparative analysis.
- Two novel corpora: MatSciNERE for materials science NER/RE and PolyEE for polymer property event extraction, providing essential resources for developing domain-specific IE systems.
- Strategic synthetic data generation for both NER and EAE tasks, demonstrating significant performance improvements (up to 5.34 F1 points).
- Extensive empirical evaluation showing our combined approaches outperform any single method, with the LLM-guided approach achieving the highest F1 score (71.53%).
- A practical solution for structured knowledge extraction from materials science literature that can adapt to different extraction scenarios.

Our unified framework addresses the critical need for structured knowledge extraction from materials science literature, offering a robust solution that can adapt to different extraction scenarios.

2 Related Work

Resources for Materials Science IE Materials science IE suffers from limited resources despite the field's importance. Early datasets like Matscholar (Weston et al., 2019) and CHEMDNER (Krallinger et al., 2015) provide entity annotations for materials science. Domain-specific resources include Mysore et al.'s (2019) corpus of 230 labeled inorganic synthesis procedures, SC-CoMIcs for superconductive materials (Yamaguchi et al., 2020), and O'Gorman et al.'s (2021) procedural text annotations. Recent corpora have significantly advanced polymer science IE: PolyNERE (Phi et al., 2024) provides a comprehensive corpus of 750 polymer abstracts with 14 entity types and 8 relation types, capturing complex structures including overlapped and discontinuous mentions. However, it only annotates entities that are mainly relevant to polymers, and a system, PolyMinder (Do et al., 2025), has been developed based on that corpus. POLYIE (Cheung et al., 2024) uses a single, coarse "*Material*" entity type, and derives N-ary relations by combining binary relations while discarding those that cannot be combined.

Methods for Structured IE NER approaches fall into three main categories: sequence labeling (Huang et al., 2015; Lample et al., 2016), which struggles with overlapping entities; span-based methods (Shen et al., 2021), which handle overlap but face scalability challenges; and generation-based approaches (Yan et al., 2021; Paolini et al., 2021), which manage flat, overlapped, and discontinuous mentions via sequence generation. Recent work includes unified frameworks like W2NER (Li et al., 2022), which formulates NER as word-to-word relation classification, enabling it to handle these entity types simultaneously.

For RE, common approaches include pipeline systems where RE follows NER (Huang et al., 2021) and joint entity and RE methods (Lu et al., 2022). For more complex n-ary relationships, EAE offers promising solutions, with models like TagPrime (Hsu et al., 2023) using classification-based approaches and PAIE (Ma et al., 2022) employing generation-based methods. These approaches have shown effectiveness in general domains but require adaptation for the specialized terminology and complex relationships in materials science.

LLMs like GPT-4 excel at general NLP tasks but often underperform on domain-specific IE. Kumar et al. (2025) note they may hallucinate and generate conversational rather than precise outputs, limiting effectiveness in tasks like property extraction. Smaller BERT-based models are more efficient, transparent, and often outperform LLMs on specialized scientific tasks.

3 Novel Corpora for Material Science IE

This section introduces the two novel annotated resources that support our unified framework: MatSciNERE for general materials science NER/RE, and PolyEE for event-based polymer property extraction.

We first present **MatSciNERE**, a high-quality corpus representing a significant advancement for materials science IE. This resource expands the PolyNERE corpus (Phi et al., 2024) with a key innovation: a revised annotation assumption that captures all entity mentions in the text, not just those directly relevant to polymers, enhancing both coverage and consistency.

Following the approach in similar work (Phi et al., 2024) and widely used datasets like Matscholar (Weston et al., 2019), a single annotator conducted the primary annotation work to ensure consistency across the corpus, building upon the extensive foundation of existing high-quality entity and relation annotations. A quality assessment was performed where a polymer expert independently annotated a sample set from the corpus, yielding a Cohen's Kappa coefficient of 0.835 and comparative metrics of 95.88% precision, 79.42% recall, and 86.93% F1 score. This validation confirms the high quality of the annotations.

MatSciNERE contains 22,296 entity mentions and 11,935 relation pairs, increasing from 18,930 entities and 11,471 relations in PolyNERE corpus. With 14 entity types and 8 relation types, it includes overlapped (15.65%) and discontinuous (1.26%) mentions crucial for capturing complex scientific expressions, enabling practical extraction systems. Detailed statistics are in Appendix A.

In this work, we also introduce **PolyEE**, a specialized corpus for event-based polymer property extraction that addresses a critical gap in materials science IE. PolyEE reformulates property information as events with property names as triggers and other elements as arguments, enabling the extraction of complete property tuples across multiple sentences.

We define our task based on concepts from PoLyInfo (Otsuka et al., 2011), the largest polymer database. We focus on five key entity types: POLYMER (polymer names like "*polyethylene*"), PROP_NAME (property names such as "*glass transition temperature*"), PROP_VALUE (values with units like "*25 MPa*"), CONDITION (measurement conditions like "*at 25*°C"), and CHAR_METHOD (characterization techniques such as "*DSC*"). These five types represent the core elements needed for polymer property information in PoLyInfo's schema, which will also serve as the primary knowledge source for developing synthetic data later in this work. Our event structure follows a single "PropertyInfo" type with PROP_NAME as the trigger and other types as arguments.

We developed PolyEE through a semi-automated approach using two advanced LLMs (GPT-4o and Claude 3.7 Sonnet Thinking) to generate initial event annotations based on existing entity and relation annotations from MatSciNERE. This dual-LLM approach allowed us to cross-validate annotations and identify potential discrepancies. Out of 750 abstracts in MatSciNERE, 503 contain at least one relevant event with a PROP_NAME trigger and the required POLYMER and PROP_VALUE arguments. When events contained multiple CONDITIONs or CHAR_METHODs, we split them into separate instances. This process yielded 1,601 distinct events reflecting various property measurements and methods. To ensure high-quality annotations, we implemented a thorough validation process. Event tuples generated by the two LLMs were manually compared, with 118 events (7.37%) across 86 abstracts (17.10%) undergoing detailed verification. Additionally, all annotations in development and test sets were manually reviewed. To assess annotation quality, we compared LLM-generated annotations with those produced by the primary corpus annotator on a sample of 10 abstracts containing the highest number of events. This comparison yielded a Cohen's Kappa of 0.87, suggesting strong consistency in the annotation process. Detailed statistics are in Appendix B.

4 Unified Framework for Property IE

Our unified framework integrates three complementary approaches for extracting n-ary property information from materials science literature. As shown in Figure 1, the architecture centers on a foundational NER module that processes scientific text to identify relevant entities. The framework then offers multiple approaches to transform these entities into structured n-ary property information: (1) RE-Composition, (2) Direct EAE, and (3) LLM-Guided Assembly. Each approach has distinct strengths, and they can be deployed individually or in combination depending on specific extraction needs. Two synthetic data generation engines enhance both the NER module and EAE models, improving overall performance.

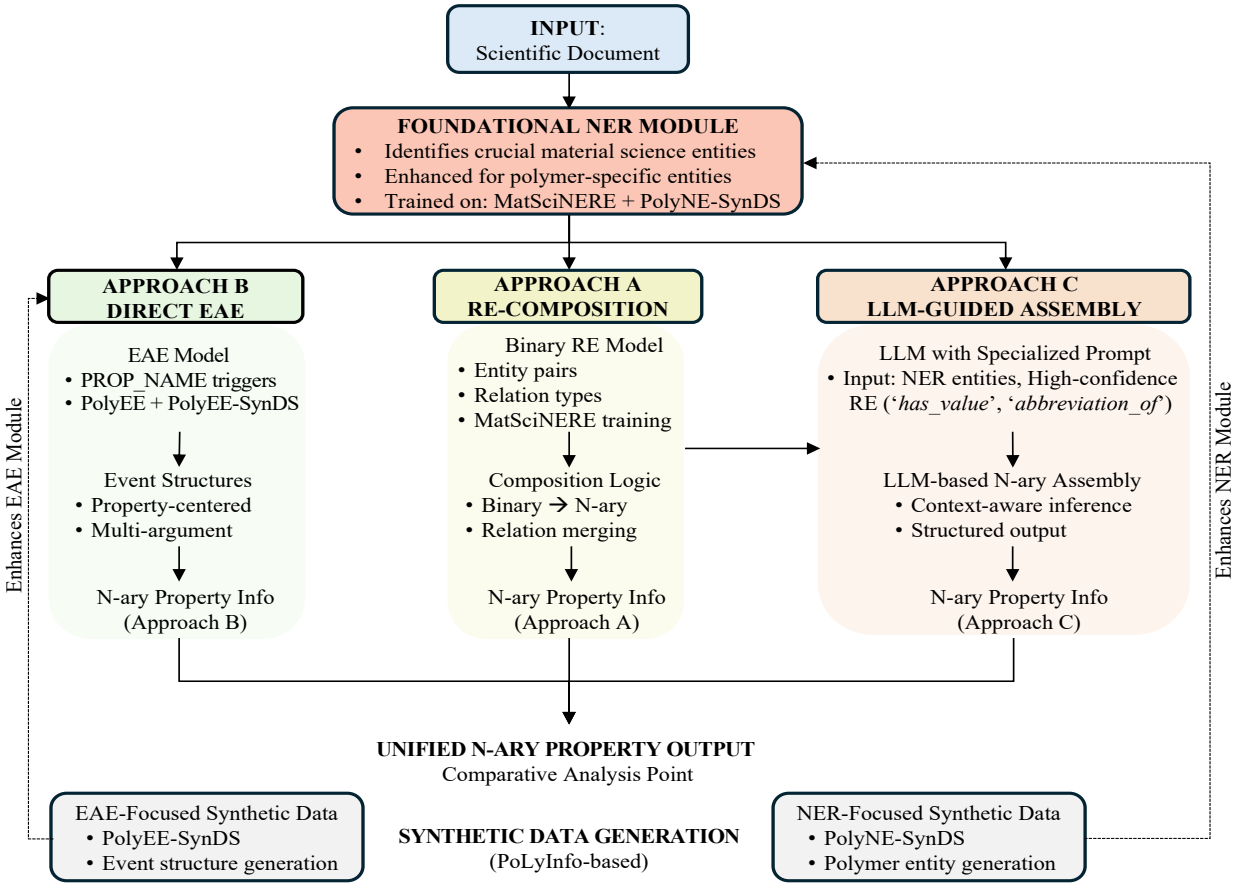


Figure 1: Our unified framework for N-ary property information extraction in materials science

Foundational NER Module The NER module identifies crucial material science entities with emphasis on polymer-specific entities, handling flat, overlapped, and discontinuous mentions. Trained on MatSciNERE and supplemented with synthetic data for polymer entities, this module significantly improves recognition of key entity types (POLYMER, PROP_NAME, PROP_VALUE, CONDITION, CHAR_METHOD). The module serves as the foundation for all extraction approaches: providing entity candidates for RE, identifying PROP_NAME triggers for event extraction, and supplying entities to guide LLM-based extraction.

RE-Composition The RE-Composition approach transforms binary relations into n-ary structures. A document-level RE model predicts binary relationships between entity pairs, trained on MatSciNERE to identify key relations like '*has_property'*, '*has_value*', etc. The composition logic then transforms these binary relations into n-ary structures by identifying patterns of connected relations sharing common entities. For example, POLYMER→has_property→PROP_NAME and PROP_NAME→has_value→PROP_VALUE can be merged into a single n-ary tuple (POLYMER, PROP_NAME, PROP_VALUE).

Direct EAE The Direct EAE approach conceptualizes polymer property information as events with PROP_NAME entities as triggers. The event schema defines a single "*PropertyInfo*" event type with four argument roles: Polymer, Value, Condition, and Char_method. The EAE model identifies all relevant arguments associated with each property name trigger, even when spanning multiple sentences. Trained on PolyEE (derived from MatSciNERE but restructured for event extraction) and enhanced with synthetic data, this approach directly captures complex, multi-argument property information without intermediate binary relation steps.

LLM-Guided Assembly The LLM-Guided Assembly approach leverages LLMs' inferential capabilities while constraining them with high-confidence predictions from the NER and RE modules. Motivated by the observation that certain binary relations (particularly '*has_value*' and '*abbreviation_of*') achieve high F1 scores, this

approach enhances LLM performance on scientific content. The component uses a specialized prompt (detailed in Appendix J) incorporating text, NER-identified entities, and high-confidence relations to guide LLMs in extracting complete n-ary property tuples while ensuring schema consistency.

Synthetic Data Generation Methodology Our framework employs two complementary synthetic data generation approaches that leverage the PoLyInfo database (Otsuka et al., 2011) to enhance both NER and EAE performance. While PoLyInfo contains predominantly lengthy and complex IUPAC nomenclature that scientific literature rarely uses in the exact form, making direct alignment between database entries and research articles challenging, we utilize LLMs and distant supervision to generate realistic linguistic variations that bridge this terminology gap while preserving scientific accuracy.

For NER, we use Llama 3.1 70B Instruct to create paragraphs rich in polymer entities based on strategically selected sample tuples from PoLyInfo. For EAE, we employ advanced LLMs (Claude 3.7 Sonnet Thinking and GPT-4.1) with specialized templates, one mimicking general materials science writing (T2) and another leveraging actual PolyEE abstracts as guides (T3), to generate self-annotated paragraphs with explicit entity tagging and structured event tuples. The generated outputs undergo three-stage distant supervision alignment: (1) entity and event tuple extraction from LLM outputs, (2) abbreviation and relation refinement, and (3) context-sensitive refinement using sentence-level linguistic analysis. Additional details are provided in Appendix F.

Integration and Flexibility A key advantage of our unified framework is its flexibility. During inference, the system can employ any single approach or combine multiple approaches based on the specific extraction needs or document characteristics. Our experimental analysis (Section 5) explores the performance characteristics of each approach and identifies optimal combinations for different scenarios, providing guidance for practical applications of the framework.

5 Experiments

5.1 Experimental Setup

Our experimental evaluation assesses each component of the unified framework both individually and in combination. We conducted comprehensive experiments on the MatSciNERE and PolyEE corpora to evaluate the performance of different pathways for n-ary property extraction.

Method	Encoder	P	R	F1
MaxClique (Wang et al., 2021)	MatSciBERT	78.65	75.50	77.04
	RoBERTa-large	76.37	76.11	76.24
W2NER (Li et al., 2022)	MatSciBERT*	78.05	76.53	77.28
	MatSciBERT	77.47	**79.18**	**78.32**
	RoBERTa-large	77.31	77.60	77.45
TriG-NER (Cabral et al., 2025)	MatSciBERT	**78.85**	74.97	76.86
	RoBERTa-large	77.17	74.89	76.01

Table 1: Results for NER on MatSciNERE test set (*trained and evaluated on PolyNERE).

Implementation Details For our unified framework, we implemented multiple state-of-the-art models for each component. The NER module utilized various advanced methods with different encoders, including domain-specific ones. Similarly, for Approach A (RE-Composition), Approach B (Direct EAE), and Approach C (LLM-Guided Assembly), we implemented several competitive models. For each experiment, the reported results represent the average of five runs. Detailed implementation specifics are provided in Appendix C.

Evaluation Metrics For NER, we report precision (P), recall (R), and F1-score at the entity level. For binary RE, we use similar metrics at the relation level. Following Lin et al. (2020), we evaluate EAE models using both argument classification (AC) and the stricter AC-attached metric, which assesses whether predicted arguments are correctly linked to their appropriate triggers, a critical consideration for property IE where correct trigger-argument associations are essential.

5.2 Results

NER Performance Table 1 presents the best-performing NER models on the MatSciNERE test set. W2NER with MatSciBERT encoder achieved the highest overall F1 score (78.32%), with well-balanced precision (77.47%) and recall (79.18%). TriG-NER with MatSciBERT showed the highest precision (78.85%) but lower recall, while MaxClique with MatSciBERT demonstrated competitive performance. Domain-specific

Method	Encoder	P	R	F1
ATLOP (Zhou et al., 2021)	MatSciBERT*	83.99	82.49	83.23
	MatSciBERT	84.15	83.87	84.01
	RoBERTa-large	86.93	86.64	86.78
	DeBERTa-v3^{L}	**87.93**	86.89	**87.40**
Eider (Xie et al., 2022)	BERT-large	78.11	74.58	76.30
	RoBERTa-large	70.45	75.42	72.85
KD-DocRE (Tan et al., 2022a)	RoBERTa-large	86.93	86.44	86.69
	DeBERTa-v3^{L}	86.59	**87.57**	87.08
PEMSCL (Guo et al., 2023)	RoBERTa-large	87.84	86.22	87.02
	DeBERTa-v3^{L}	86.22	87.31	86.76

Table 2: Results for RE on MatSciNERE test set given gold entities, ^{L}large version (*trained and evaluated on PolyNERE).

MatSciBERT consistently outperformed general-purpose encoders like RoBERTa-large across all architectures, with F1 improvements of 0.80-0.87%. This highlights the importance of domain-specific pre-training for materials science text. Additional experiments with various approaches (span-based, transition-based, generation-based) and other encoders (BERT-large, SciBERT) are detailed in Appendix D.

The W2NER model trained and evaluated on PolyNERE (denoted with *) achieved a lower F1 score (77.28%) than when trained on our comprehensive MatSciNERE corpus, demonstrating the value of our expanded annotation approach for materials science NER.

RE Performance Binary RE results on MatSciNERE using gold standard entity mentions are shown in Table 2. ATLOP with DeBERTa-v3-large encoder achieved the highest F1 score of 87.40%, outperforming other model configurations. ATLOP, KD-DocRE, and PEMSCL demonstrated robust performance across different encoders, consistently achieving F1 scores above 86% when paired with powerful encoders like RoBERTa-large or DeBERTa-v3-large. Interestingly, we observe that DeBERTa-v3-large generally outperforms domain-specific encoders like MatSciBERT across most models, suggesting that the advanced architecture and larger scale of DeBERTa may compensate for domain specialization in this RE task.

The exception to these strong results comes from our approach based on Eider, where performance was notably lower (best F1 of 76.30% with BERT-large). This is likely due to our implementation constraint of using only one or two sentences containing head and tail entities as evidence sentences (see Appendix C), which limits the model's access to broader contextual information. This limitation highlights a potential area for improvement in our framework. Additional experiments with other approaches (DocuNet) and encoders (BERT-large, SciBERT, MatSciBERT) are in Appendix E.

Method	AC			AC-attached		
	P	R	F1	P	R	F1
TagPrime-C (Hsu et al., 2023)	80.39	**71.93**	**75.93**	72.07	**64.86**	**68.28**
TagPrime-CR (Hsu et al., 2023)	79.84	69.72	74.44	70.46	61.89	65.90
DEGREE (Hsu et al., 2022)	**80.65**	56.31	66.31	**73.53**	53.19	61.73
X-GEAR (Huang et al., 2022)	76.37	62.61	68.81	69.46	58.87	63.72
PAIE (Ma et al., 2022)	73.36	66.90	69.98	66.98	58.65	62.54

Table 3: Results for EAE on PolyEE test set.

Analysis by relation type revealed particularly high F1 scores for '*has_value'* (94.93%) and '*abbreviation_of*' (94.95%) relations. This finding motivated the design of Approach C, which leverages these high-confidence relations to guide LLM-based n-ary extraction.

EAE Performance Table 3 presents the performance of leading EAE models on the PolyEE test set using DeBERTa-v3-large for classification-based models and T5-large for generation-based models. We report both standard Argument Classification (AC) metrics and the stricter AC-attached metric, which evaluates correct linking of arguments to triggers. TagPrime-C achieves the highest F1 score (75.93%) in standard AC evaluation with balanced precision (80.39%) and recall (71.93%), while also leading under the AC-attached metric (68.28%). TagPrime-CR performs competitively (74.44% F1) but sees a larger drop in the AC-attached setting. DEGREE shows the highest precision-recall imbalance, making accurate but fewer predictions, though it maintains more consistent performance across both metrics. Classification-based approaches consistently outperform generation-based methods in our experiments, contrary to some findings in general domain event extraction. This aligns with the characteristics of scientific text where property

Training Strategy	Synthetic Data (paragraphs)	F1 (AC)	F1 (AC-attached)
Gold only	0	75.93	68.28
Combined Training	1,000 (Claude)	77.47	68.94
	1,000 (GPT)	76.77	68.86
	2,000 (GPT)	**78.08**	**73.62**
	5,000 (GPT)	77.86	71.57
Pre-train →Fine-tune	10,000 (GPT)	77.23	70.67
	20,000 (GPT)	77.41	70.59

Table 4: EAE results with synthetic data using GPT-4.1 and Claude 3.7 Sonnet Thinking.

information follows more predictable patterns. Additional experiments with various settings are provided in Appendix G.

EAE-Focused Synthetic Data Impact Table 4 presents the impact of synthetic data on EAE performance using TagPrime-C with DeBERTa-v3-large encoder across different training strategies. The baseline model trained only on gold-standard PolyEE data achieved an F1 score of 75.93% on the AC metric and 68.28% on the AC-attached metric. When combining the gold data with just 1,000 synthetic paragraphs generated by Claude, we observed modest improvements to 77.47% (AC) and 68.94% (AC-attached). Similarly, incorporating 1,000 GPT-generated paragraphs yielded comparable gains. Most notably, the combined training strategy with 2,000 GPT-generated paragraphs produced the optimal results, with F1 scores of 78.08% (AC) and 73.62% (AC-attached). This represents a substantial improvement of 2.15 percentage points for AC and 5.34 percentage points for AC-attached compared to the gold-only baseline.

Interestingly, increasing the synthetic data volume beyond 2,000 paragraphs did not yield further improvements. The performance slightly decreased with 5,000 paragraphs (77.86% AC, 71.57% AC-attached), suggesting that model capacity or data quality factors may limit the benefits of additional synthetic examples. Similarly, pre-training on 10,000 or 20,000 synthetic GPT-generated paragraphs followed by fine-tuning on gold standard data showed comparable but not superior performance to the optimal combined training approach.

These results demonstrate that moderate amounts of high-quality synthetic data can significantly enhance EAE performance. The synthetic data helps address the limitations of the supervised corpus, especially regarding the representation of complex argument patterns. Additional analyses regarding prompt templates and the potential of EAE-focused synthetic data to improve NER or RE tasks are provided in Appendix H.

Entity Type	Synthetic Data (sentences)					
	0	5k	10k	30k	50k	183k
POLYMER	84.42	85.19	84.38	85.63	**85.71**	82.69
PROP_NAME	83.12	83.39	83.51	83.99	84.26	**84.69**
PROP_VALUE	84.28	88.22	86.84	88.96	**89.19**	85.52
CONDITION	68.50	69.46	**71.57**	69.59	68.48	67.08
CHAR_METHOD	**90.25**	89.89	91.43	89.01	89.14	89.77
Overall	82.87	84.07	83.90	84.34	**84.69**	83.33

Table 5: NER results with synthetic data.

NER-Focused Synthetic Data Impact We evaluated the impact of synthetic data on NER performance using our best-performing model configuration (W2NER with MatSciBERT encoder). Since the synthetic data designed for EAE is not optimal for NER improvement, we developed a specialized approach using Llama 3.1 70b Instruct to generate NER-focused synthetic data targeting polymer entities in PoLyInfo. We selected this model for synthetic data generation because NER is relatively less complex than EAE. Table 5 presents the F1 scores across entity types with varying amounts of synthetic data. Adding synthetic data to the baseline MatSciNERE model (F1: 82.87%) yielded consistent improvements, peaking at 84.69% F1 with 50k synthetic sentences. Entity-specific gains varied: POLYMER improved from 84.42% to 85.71%, PROP_NAME from 83.12% to 84.26%, and PROP_VALUE showed the largest gain from 84.28% to 89.19%. The challenging CONDITION entities improved from 68.50% to 71.57%, while high-performing CHAR_METHOD entities (90.25%) maintained strong results despite fluctuations. We believe these improvements are particularly significant for crucial polymer-specific entities in the PoLyInfo database, directly enhancing the framework's ability to extract structured property information from polymer literature. Moderate synthetic data volumes (30k-50k sentences) proved optimal, with larger amounts (183k) showing diminishing returns (F1 decreasing to 83.33%), indicating quality of synthetic data matters more than quantity.

Approach	F1
LLM-based + NER + has_value + abbreviation_of	71.53
EAE + Predicted NER Refinement	69.14
EAE (base)	66.72
LLM only (GPT 4.1)	64.91
EAE + Predicted has_value Constraints	64.35
RE-Composition	58.87
LLM only (Llama 3.1 8B)	36.46

Table 6: F1 scores of n-ary property extraction.

LLM-Guided Assembly Results (Approach C) This approach was motivated by our observation that certain binary relations, particularly '*has_value*' (94.93% F1) and '*abbreviation_of*' (94.95% F1), achieve exceptionally high performance with our RE models. These domain-independent relations serve as critical connectors in property information structures across materials science contexts, allowing our framework to leverage these high-confidence predictions to guide LLMs in extracting structured information for either all 14 entity types or focusing on the 5 key polymer-specific entity types.

For resource-constrained deployment scenarios, we evaluated Llama 3.1 8B Instruct (fp16), which operates efficiently on a single GPU with 16GB memory, a significant advantage over larger models requiring multiple high-capacity GPUs or closed-source API-dependent models.

Complementary Strengths and Comparative Analysis Our framework's pathways exhibit distinct strengths for different extraction scenarios. Approach A (RE-Composition) excels with discontinuous entity mentions and explicit binary relations, offering high interpretability. Approach B (Direct EAE) handles standardized property descriptions and cross-sentence dependencies effectively. Approach C (LLM-Guided Assembly) provides flexibility across diverse phrasing styles while leveraging high-confidence relations and operating efficiently on modest hardware.

Table 6 presents comparative results across our approaches on a diverse test set of 10 real paragraphs. The LLM-guided approach achieves the highest performance for end-to-end n-ary property tuple extraction (F1: 71.53%), followed closely by EAE with predicted NER refinement (F1: 69.14%). The significant gap between guided and unguided LLM approaches (64.91% vs. 36.46% F1) confirms that even powerful models require domain-specific guidance for reliable scientific IE. This creates an efficient hybrid approach balancing performance with practical deployment considerations.

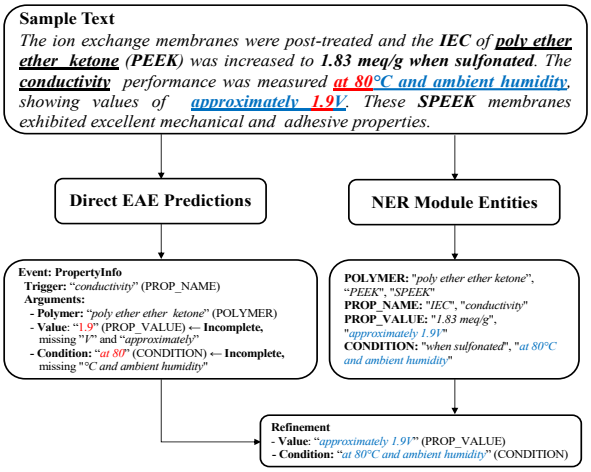


Figure 2: A case study.

Figure 2 illustrates this with a case study of the EAE + NER refinement process. Additionally, our framework's key advantage is its adaptability to different extraction requirements, allowing users to select approaches based on their specific constraints: EAE-based methods for limited computational resources, LLM-guided approaches for maximum accuracy, or RE-Composition when focusing on specific relation types.

6 Conclusion

We presented a unified framework for n-ary property IE in materials science integrating three approaches: RE-Composition, Direct EAE, and LLM-Guided Assembly. Built upon four distinct corpora, two standard (MatSciNERE and PolyEE) and two synthetic (NER-focused and EAE-focused), our framework shows synthetic data enhances model performance across components, with optimal gains at moderate volumes.

Experimental results confirm different extraction scenarios benefit from different approaches, with the LLM-guided approach achieving highest performance when constrained, while specialized models offer competitive results with deployment advantages. Our work addresses a critical need by advancing structured property IE in materials science.

Future work will expand to additional scientific domains, improve pathway integration, and explore techniques to reduce LLM hallucination in scientific contexts.

Limitations

The LLM-Guided Assembly approach, despite achieving the highest performance, relies on external LLM infrastructure and remains susceptible to hallucination even with our guiding mechanisms in place. Additionally, the full framework implementation demands substantial computational resources, especially when utilizing larger encoders such as DeBERTa-v3-large or employing the LLM-Guided Assembly approach. These resource requirements may be excessive in computationally constrained environments. Although we offer more efficient alternatives (such as W2NER+MatSciBERT for NER), these options inevitably involve performance trade-offs.

References

Bajan, C., & Lambard, G. (2025). Exploring the expertise of large language models in materials science and metallurgical engineering. Digital Discovery.

Cabral, R. C., Han, S. C., Alhassan, A., Batista-Navarro, R., Nenadic, G., & Poon, J. (2025, April). TriG-NER: Triplet-Grid Framework for Discontinuous Named Entity Recognition. In Proceedings of the ACM on Web Conference 2025 (pp. 2824-2837).

Dai, X., Karimi, S., Hachey, B., & Paris, C. (2020). An effective transition-based model for discontinuous NER. arXiv preprint arXiv:2004.13454.

Do, T. D., Trieu, A. H., Phi, V. T., Le Nguyen, M., & Matsumoto, Y. (2025, January). PolyMinder: A Support System for Entity Annotation and Relation Extraction in Polymer Science Documents. In Proceedings of the 31st International Conference on Computational Linguistics: System Demonstrations (pp. 1-8).

Guo, J., Kok, S., & Bing, L. (2023). Towards integration of discriminability and robustness for document-level relation extraction. arXiv preprint arXiv:2304.00824.

Hsu, I., Huang, K., Boschee, E., Miller, S., Natarajan, P., Chang, K., & Peng, N. (2022). DEGREE: A Data-Efficient Generation-Based Event Extraction Model. North American Chapter of the Association for Computational Linguistics.

Huang, K. H., Hsu, I., Natarajan, P., Chang, K. W., & Peng, N. (2022). Multilingual generative language models for zero-shot cross-lingual event argument extraction. arXiv preprint arXiv:2203.08308.

Huang, K. H., Tang, S., & Peng, N. (2021). Document-level entity-based extraction as template generation. arXiv preprint arXiv:2109.04901.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2023. TAGPRIME: A Unified Framework for Relational Structure Extraction. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12917–12932, Toronto, Canada. Association for Computational Linguistics.

Jerry Cheung, Yuchen Zhuang, Yinghao Li, Pranav Shetty, Wantian Zhao, Sanjeev Grampurohit, Rampi Ramprasad, and Chao Zhang. 2024. POLYIE: A Dataset of Information Extraction from Polymer Material Scientific Literature. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2370–2385, Mexico City, Mexico. Association for Computational Linguistics.

Krallinger, Martin, et al. "The CHEMDNER corpus of chemicals and drugs and its annotation principles." Journal of cheminformatics 7 (2015): 1-17.

Kumar, P., Kabra, S., & Cole, J. M. (2025). MechBERT: Language Models for Extracting Chemical and Property Relationships about Mechanical Stress and Strain. Journal of Chemical Information and Modeling.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

Li, F., Lin, Z., Zhang, M., & Ji, D. (2021). A span-based model for joint overlapped and discontinuous named entity recognition. arXiv preprint arXiv:2106.14373.

Li, J., Fei, H., Liu, J., Wu, S., Zhang, M., Teng, C., Ji, D. and Li, F., 2022, June. Unified named entity recognition as word-word relation classification. In proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 10, pp. 10965-10973).

Lu, Yaojie, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. "Unified structure generation for universal information extraction." arXiv preprint arXiv:2203.12277 (2022).

Mysore, S., Jensen, Z., Kim, E.J., Huang, K., Chang, H., Strubell, E., Flanigan, J., McCallum, A., & Olivetti, E.A. (2019). The Materials Science Procedural Text Corpus: Annotating Materials

Synthesis Procedures with Shallow Semantic Structures. LAW@ACL.

O'Gorman, T.J., Jensen, Z., Mysore, S., Huang, K., Mahbub, R., Olivetti, E.A., & McCallum, A. (2021). MS-Mentions: Consistently Annotating Entity Mentions in Materials Science Procedural Text. Conference on Empirical Methods in Natural Language Processing.

Otsuka, S., Kuwajima, I., Hosoya, J., Xu, Y., & Yamazaki, M. (2011, September). PoLyInfo: Polymer database for polymeric materials design. In 2011 International Conference on Emerging Intelligent Data and Web Technologies (pp. 22-29). IEEE.

Paolini, G., Athiwaratkun, B., Krone, J., Ma, J., Achille, A., Anubhai, R., Santos, C.N.D., Xiang, B. and Soatto, S., 2021. Structured prediction as translation between augmented natural languages. arXiv preprint arXiv:2101.05779.

Phi, V. T., Teranishi, H., Matsumoto, Y., Oka, H., & Ishii, M. (2024, May). PolyNERE: A Novel Ontology and Corpus for Named Entity Recognition and Relation Extraction in Polymer Science Domain. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 12856-12866).

Shen, Y., Ma, X., Tan, Z., Zhang, S., Wang, W., & Lu, W. (2021). Locate and label: A two-stage identifier for nested named entity recognition. arXiv preprint arXiv:2105.06804.

Shetty, P., Rajan, A.C., Kuenneth, C., Gupta, S., Panchumarti, L.P., Holm, L., Zhang, C., & Ramprasad, R. (2023). A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. Npj Computational Materials, 9.

Tan, Q., He, R., Bing, L. and Ng, H.T., 2022. Document-level relation extraction with adaptive focal loss and knowledge distillation. arXiv preprint arXiv:2203.10900.

Wang, Y., Yu, B., Zhu, H., Liu, T., Yu, N., & Sun, L. (2021). Discontinuous named entity recognition as maximal clique discovery. arXiv preprint arXiv:2106.00218.

Weston, L., Tshitoyan, V., Dagdelen, J., Kononova, O.V., Trewartha, A., Persson, K.A., Ceder, G., & Jain, A. (2019). Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature. Journal of chemical information and modeling.

Xie, Y., Shen, J., Li, S., Mao, Y., & Han, J. (2021). Eider: Empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. arXiv preprint arXiv:2106.08657.

Yamaguchi, K., Asahi, R., & Sasaki, Y. (2020, May). SC-CoMIcs: A superconductivity corpus for materials informatics. In Proceedings of the Twelfth Language Resources and Evaluation Conference (pp. 6753-6760).

Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., & Qiu, X. (2021). A unified generative framework for various NER subtasks. arXiv preprint arXiv:2106.01223.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

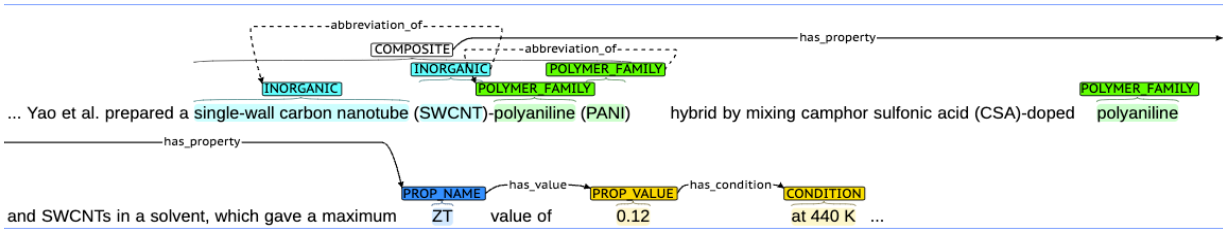
Zhang, N., Chen, X., Xie, X., Deng, S., Tan, C., Chen, M., Huang, F., Si, L. and Chen, H., 2021. Document-level relation extraction as semantic segmentation. arXiv preprint arXiv:2106.03618.

Zhou, W., Huang, K., Ma, T., & Huang, J. (2021, May). Document-level relation extraction with adaptive thresholding and localized context pooling. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 16, pp. 14612-14620).

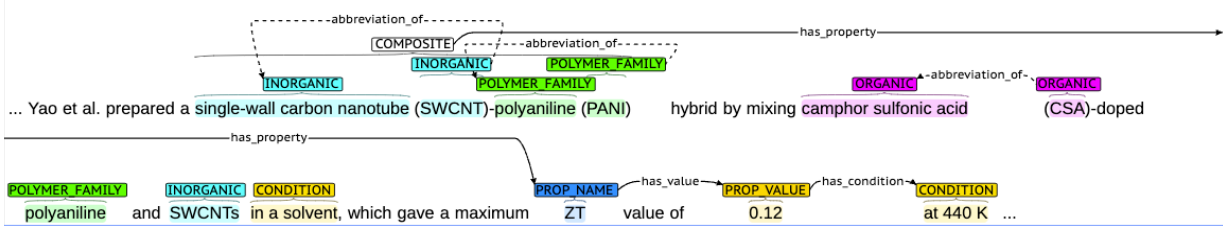
A Construction of MatSciNERE Corpus

In the PolyNERE corpus, only entities and relations relevant to our target domain were annotated. As a result, in Figure 3a, even though the second '*SWCNTs*' mention is an inorganic material, it was not provided a label. We argue that it greatly affects the overall performance of NER and RE systems trained on those annotations, and limits the practical usage of these systems. Therefore, in this work, we made an important change in the annotation assumption by considering all entity mentions in the text, aiming at a practical RE system for both general material science and its subdomains like polymer science. Figure 3b illustrates our new annotation assumption applied to our newly developed MatSciNERE corpus.

Our MatSciNERE corpus consists of a total of 750 abstracts, divided into three sets: 637 for training, 38 for development, and 75 for testing. Table 7 displays the statistics for our corpus, presenting details about the annotation type, and the number of annotations across various categories within MatSciNERE. Overall, our MatSciNERE corpus provides a rich source of



(a) Annotations specific to the field of polymer science



(b) Annotations applicable to the broader domain of materials science

Figure 3: Annotation assumption (a) In PolyNERE corpus: only entities and relations relevant to the target domain were annotated; the first '*SWCNTs*' was labeled as it is part of a composite with another polymer class. (b) In our MatSciNERE corpus: other labeled entities include '*camphor sulfonic acid*', '*CSA*'.

#tokens/sentence	11.87
#entities/abstract	29.73
#relations/abstract	15.91
Overlapped entities	3,490 mentions (15.65%)
Discontinuous entities	281 mentions (1.26%)
ENTITY (14)	Total: 22,296 mentions
POLYMER	4,053 (582/750 abstracts)
POLYMER_FAMILY	1,159 (315)
PROP_NAME	3,882 (717)
PROP_VALUE	1,829 (587)
MONOMER	1,600 (320)
ORGANIC	1,855 (435)
INORGANIC	1,939 (393)
MATERIAL_AMOUNT	539 (267)
COMPOSITE	398 (172)
OTHER_MATERIAL	258 (120)
CONDITION	1,376 (552)
SYN_METHOD	381 (231)
CHAR_METHOD	1,752 (435)
REF_EXP	1,275 (460)
RELATION (8)	Total: 11,935 pairs
has_property	3,502 (661/750 abstracts)
has_value	1,903 (582)
has_amount	424 (225)
has_condition	1,104 (406)
synthesised_by	282 (193)
characterized_by	1,347 (391)
abbreviation_of	2,033 (627)
refers_to	1,340 (459)

Table 7: Statistics of our MatSciNERE corpus.

information for training and evaluating models in the field of polymer science, particularly for tasks related to NER and RE.

B PolyEE Corpus Details

We selected five types (POLYMER, PROP_NAME, PROP_VALUE, CONDITION, CHAR_METHOD) because they represent the core elements needed for polymer property information in PoLyInfo's schema and because annotations for other entity types are relatively sparse. Our PolyEE corpus comprises 503 abstracts and 1,601 events, with a balanced split into training (396 abstracts, 1,253 events), development (50 abstracts, 177 events), and test (50 abstracts, 171 events). A few abstracts were excluded from model training because certain event arguments exceeded the 512-token limit.

C Implementation Details

Model Selection and Implementation For NER tasks, we experimented with multiple architectures capable of handling flat, overlapped, and discontinuous mentions: Span-based (Li et al., 2021), Transition-based (Dai et al., 2020), MaxClique (Wang et al., 2021), BARTNER (Yan et al., 2021), W2NER (Li et al., 2022) and TriG-NER (Cabral et al., 2025).

For RE tasks, we implemented several document-level models: DocuNet (Zhang et al., 2021), ATLOP (Zhou et al., 2021), KD-DocRE (Tan et al., 2022a), PEMSCL (Guo et al., 2023), and Eider (Xie et al., 2022). With Eider, we

Method	P	R	F1
Span-based (Li et al., 2021)	59.56	26.66	36.84
Transition-based (Dai et al., 2020)	73.49	72.03	72.75
MaxClique (Wang et al., 2021)	77.35	71.86	74.51
BARTNER (Yan et al., 2021)	74.80	73.25	74.02
W2NER (Li et al., 2022)	**78.27**	74.39	76.28
TriG-NER (Cabral et al., 2025)	77.77	**75.34**	**76.54**

Table 8: Results for NER on MatSciNERE test set with BERT-large encoder.

Method	Encoder	P	R	F1
MaxClique (Wang et al., 2021)	BERT-large	77.35	71.86	74.51
	SciBERT	**80.64**	71.96	76.05
	MatSciBERT	78.65	75.50	77.04
	RoBERTa-large	76.37	76.11	76.24
W2NER (Li et al., 2022)	BERT-large	78.27	74.39	76.28
	SciBERT	75.23	76.84	75.85
	MatSciBERT	77.47	**79.18**	**78.32**
	RoBERTa-large	77.31	77.60	77.45
TriG-NER (Cabral et al., 2025)	BERT-large	77.77	75.34	76.54
	SciBERT	73.64	72.51	73.07
	MatSciBERT	78.85	74.97	76.86
	RoBERTa-large	77.17	74.89	76.01

Table 9: Results for NER on MatSciNERE test set with other encoders.

restricted the context to only one or two sentences containing both head and tail entities as evidence, due to memory constraints with this model's evidence extraction mechanism.

For EAE experiments, we evaluated both classification-based and generation-based approaches: TagPrime-C and TagPrime-CR (Hsu et al., 2023), DEGREE (Hsu et al., 2022), X-Gear (Huang et al., 2022), and PAIE (Ma et al., 2022).

Training Configuration For NER and RE models, we used Adam optimizer (Kingma and Ba, 2015) with linear warmup and decay learning rate schedules. All models were trained for 30 epochs with a batch size of 8, maintaining consistent hyperparameter settings across baselines where applicable.

For EAE models, we extended training to a maximum of 90 epochs. For generation-based methods, we set the maximum output text length to 200 tokens.

Entity Type	F1	Entity Type	F1
POLYMER	84.42	PROP_NAME	83.12
MONOMER	75.58	PROP_VALUE	84.28
POLYMER_ FAMILY	69.43	MATERIAL_ AMOUNT	80.00
ORGANIC	59.38	CONDITION	68.50
INORGANIC	82.85	SYN_ METHOD	76.92
COMPOSITE	54.32	CHAR_ METHOD	90.25
OTHER_ MATERIAL	35.29	REF_EXP	74.25
Overall	78.32		

Table 10: NER performance on MatSciNERE test set using W2NER with a MatSciBERT encoder.

Evaluation Metrics Following Lin et al. (2020), we primarily report argument classification (AC) metrics for EAE performance. Additionally, we include the more stringent AC-attached metric (Huang et al., 2024), which evaluates whether arguments are correctly linked to their appropriate triggers. For example, in the text "*poly(p-diethynylbenzene) has a density of 1.097 g/cm³ at 25°C and a density of 1.082 g/cm³ at 50°C*", if a model associates "*1.082 g/cm³*" with the first "*density*" mention, this would be considered correct under the AC metric but incorrect under the AC-attached metric.

Computing Infrastructure All experiments were conducted on a single NVIDIA A100 40GB GPU, except for NER-focused synthetic data generation with Llama 3.1 70b Instruct (FP16), which required 4 × NVIDIA A100 40GB GPUs.

D NER Performance

Table 8 compares the performance of different NER methods on the MatSciNERE test set with BERT-large encoder, showing that W2NER and TriG-NER achieve the highest F1 scores of 76.28% and 76.54% respectively.

Table 9 demonstrates that domain-specific encoders, particularly MatSciBERT, consistently outperform general-purpose encoders across different NER architectures, with W2NER+MatSciBERT achieving the highest overall F1 score of 78.32%.

Performance Analysis by Entity Type Analysis by entity type revealed varying performance levels

Method	Encoder	P	R	F1
DocuNet (Zhang et al., 2021)	BERT-large	78.42	81.07	79.72
	SciBERT	77.03	77.68	77.36
	MatSciBERT	75.81	79.66	77.69
	RoBERTa-large	65.69	75.71	70.34
	DeBERTa-v3^{L}	77.22	61.30	68.35
ATLOP (Zhou et al., 2021)	BERT-large	84.67	74.73	79.39
	SciBERT	83.96	82.37	83.16
	MatSciBERT	84.15	83.87	84.01
	RoBERTa-large	86.93	86.64	86.78
	DeBERTa-v3^{L}	**87.93**	86.89	**87.40**
Eider (Xie et al., 2022)	BERT-large	78.11	74.58	76.30
	SciBERT	71.59	71.17	71.39
	MatSciBERT	71.89	68.64	70.23
	RoBERTa-large	70.45	75.42	72.85
KD-DocRE (Tan et al., 2022a)	BERT-large	82.34	77.68	79.94
	SciBERT	82.43	86.16	84.25
	MatSciBERT	83.35	87.21	85.24
	RoBERTa-large	86.93	86.44	86.69
	DeBERTa-v3^{L}	86.59	**87.57**	87.08
PEMSCL (Guo et al., 2023)	BERT-large	83.24	75.55	79.21
	SciBERT	83.62	82.35	82.98
	MatSciBERT	83.39	85.21	84.29
	RoBERTa-large	87.84	86.22	87.02
	DeBERTa-v3^{L}	86.22	87.31	86.76

Table 11: Results for RE on MatSciNERE test set given gold entities; ^{L}DeBERTa-v3-large.

across different categories as shown in Table 10. Core entities for property extraction performed well, with F1 scores of 84.42% for POLYMER, 83.12% for PROP_NAME, and 84.28% for PROP_VALUE. CHAR_METHOD achieved the highest F1 score (90.25%), while MATERIAL_AMOUNT (80.00%) and INORGANIC (82.85%) also showed strong results. Challenges remain for CONDITION (68.50%), POLYMER_FAMILY (69.43%), and COMPOSITE (54.32%). The lowest performance was observed for OTHER_MATERIAL (35.29%). Overall, the model achieved a 78.32% F1 score across all entity types.

E RE Performance

Table 11 presents the performance of various RE models (DocuNet, ATLOP, Eider, KD-DocRE, and PEMSCL) with different encoders, showing that

Relation Type	F1
has_property	86.72
has_value	94.93
has_amount	74.29
has_condition	79.50
synthesised_by	75.86
characterized_by	88.12
abbreviation_of	94.95
refers_to	82.58
Overall	87.40

Table 12: Results for RE on MatSciNERE test set using ATLOP with a DeBERTa-v3-large encoder.

DeBERTa-v3-large achieves the highest F1 score (87.40%) with ATLOP architecture, while MatSciBERT remains competitive across most model configurations. Table 12 breaks down RE performance by relation type, revealing particularly high F1 scores for '*has_value*' (94.93%) and '*abbreviation_of*' (94.95%) relations, which subsequently motivated the LLM-Guided Assembly approach in the framework.

Material Science Event Extraction Task

Paragraph:
INSERT HERE

Entity and Relation Annotations:
INSERT HERE

Instructions:
Extract all event tuples in the form of (POLYMER, PROP_NAME, PROP_VALUE, CONDITION, CHAR_METHOD) from the paragraph using the provided entity and relation annotations.

Entity Types Explanation:
- POLYMER: The polymer material being described
- PROP_NAME: Property name (e.g., molecular weight, conductivity, etc.)
- PROP_VALUE: Property value (the measured or reported value of the property)
- CONDITION: Experimental or measurement conditions (multiple mentions allowed)
- CHAR_METHOD: Characterization or measuring method used (multiple mentions allowed)

Requirements:
1. Each tuple must contain at minimum: POLYMER entity, PROP_NAME entity, and PROP_VALUE entity
2. CONDITION and CHAR_METHOD entities are optional fields and can have multiple mentions
3. Use EXACTLY the same entity mentions as provided in the annotations - do not combine two or more mentions
4. Only include tuples where PROP_NAME and PROP_VALUE have a "has_value" relation in the annotations (e.g., "R3 has_value Arg1:T4 Arg2:T5")
5. Format each tuple on a new line with clear labeling
6. Return ONLY the event tuples without any explanations or additional text

Example Format:
1. (Polymer: "X", Property: "Y", Value: "Z", Condition: ["A", "B"], Method: ["C", "D"])
2. (Polymer: "P", Property: "Q", Value: "R", Condition: [], Method: ["S"])

Note: Empty fields for condition or method should be represented as empty lists as shown in example 2.

Figure 4: Specialized prompt template used for generating the PolyEE corpus.

You are an expert in polymer science and materials engineering. Your task is to generate a detailed, realistic paragraph discussing multiple polymers and their properties, and then extract relevant event tuples from that paragraph.

Use the following guidelines:

1. **Context**: Choose a random context from this list: research paper, industry report, blog post, conference presentation, classroom lecture, patent application, product development meeting, or materials database entry.
2. **Polymers**: Include the provided polymers. Compare and contrast their properties, structures, and characteristics.
3. **Properties**: Discuss the provided properties for each polymer, ensuring to differentiate between polymers that share property names but have different values.
4. **Measurement Methods**: Describe the provided measurement techniques or instruments used to determine the properties, ensuring clarity when multiple methods are involved.
5. **Conditions**: Mention the provided conditions under which properties were measured or observed, and be prepared to include multiple conditions for different properties.
6. **Relationships**: Establish complex relationships between polymers, properties, conditions, and measurement methods, ensuring that the context is clear and logical.
7. **Technical Details**: Include specific numerical values, units, chemical formulas, and polymer classifications where appropriate.

Before writing the paragraph:

1. For conditions, create natural prepositional phrases that start with words like "under", "at", "in", etc. Try to randomly generate some concrete details based on the original data in the provided sample information (e.g., values like "at a heating rate of 10°C/min and a frequency of 0.1 Hz").
2. For measuring methods, use complete noun phrases that indicate the measuring or characterization method. Only the actual method name/instrument will be tagged.

Use the following data as a basis for your paragraph, but feel free to extrapolate or invent additional coherent details:

-----Sample Information-----

[INSERT SAMPLE INFORMATION HERE]

Generate a paragraph of 250-350 words that incorporates these elements in a natural, flowing manner. Ensure that the information is presented in a way that would challenge an ML model to correctly identify and relate entities, properties, conditions, and measurement methods. Vary the writing style and complexity to create diverse and realistic content. Provide only the paragraphs without any additional explanation.

After generating the paragraph, extract and list all relevant event tuples from the text in the following format:

Event Tuples:

1. (Polymer: "...", Property: "...", Value: "...", Conditions: "...", Method: "...")
2. (Polymer: "...", Property: "...", Value: "...", Conditions: "...", Method: "...")

...

Each tuple should contain the exact tagged text from your paragraph, including the complete phrases used for conditions and methods.

TAGGING RULES:

- Polymer names: <P>exact polymer name as provided</P>
- Property names: <PROP>exact property name as provided</PROP>
- Property values: <VAL>exact value with unit as provided</VAL>
- Conditions: <COND>entire prepositional phrase</COND>
- Measuring methods: <M>only the method/instrument name</M>

IMPORTANT: Ensure that every property measurement mentioned in the paragraph is represented in the event tuples, and that the text in the tuples exactly matches the tagged text in the paragraph.

Figure 5: Base prompt template for generating synthetic EAE data using Templates 2 and 3.

F LLM Prompt Templates for Event Annotation and Data Generation

Figure 4 illustrates the specialized prompt template used to guide advanced LLMs (GPT-4.1 and Claude 3.7 Sonnet Thinking) in generating event annotations from existing entity and relation data to create the PolyEE corpus, while Figure 5 presents the base prompt structure employed for synthetic EAE data generation that either mimics general materials science writing (Template 2) or leverages actual PolyEE abstracts as contextual guides (Template 3).

Important instructions for Prompt Template 2 include:

***Natural Scientific Writing**:*
- Use natural scientific writing style with proper sentence case (not copying capitalization from the sample information)
- Introduce and use appropriate abbreviations for polymers, properties, and methods (e.g., "glass transition temperature (Tg)" then later just "Tg")
- Vary how you refer to properties and methods rather than always using the exact terms from the sample information
- Use domain-specific jargon and conventions as real scientists would

***Linguistic Complexity**: Incorporate these challenging linguistic patterns:*
- Coreference and anaphora (using "it", "this polymer", "the latter compound", etc.)
- Discontinuous mentions (separating polymer names from their properties)
- Nested relationships (properties that depend on other properties)
- Comparative statements between polymers
- Negation patterns ("unlike X, polymer Y does not exhibit...")
- Hedging language ("appears to have", "approximately", "estimated to be")

***Domain Challenges**: Include these domain-specific complexities:*
- Properties mentioned for polymer blends or composites
- Conditional properties (that only appear under specific circumstances)
- Properties that change over time or processing conditions
- Implicit relationships that domain experts would understand
- Abbreviated or alternative names for the same polymer

Critical instructions for Prompt Template 3 include:

***YOU MUST strictly imitate the exact sentence structure, flow, and technical presentation format of the provided sample text.** Pay careful attention to:*
- Sentence length and complexity
- How information is sequenced
- Paragraph structure and flow
- Technical term introduction patterns
- Transitions between ideas
- Use of supporting details
- Types of clauses and grammatical constructions
- How data and measurements are presented

Sample text:
[INSERT HERE]

Your generated paragraph should be nearly indistinguishable in structure from this sample - as if written by the same author, but about your assigned polymers.

Synthetic Data Generation Details

Our **NER-focused synthetic data generation approach** follows a structured multi-stage pipeline:

1. Strategic Sample Selection: For each synthetic instance, we select sample information tuples from PoLyInfo, combining them based on common polymer names and property names when possible. We extract polymer names (POLYMER), associated property names (PROP_NAME), corresponding property values (PROP_VALUE), experimental conditions (CONDITION), and characterization techniques (CHAR_METHOD). This strategic selection creates complex relationships that challenge ML models to learn effectively. We focus on these polymer-relevant entities because resources for other entity types are limited and lack sufficient quality for effective data generation. CONDITION and CHAR_METHOD can sometimes be absent in the database for certain property entries.

2. Contextual Prompt Engineering: We transform the selected polymer information into carefully engineered prompts for the Llama 3.1 70B model. These prompts incorporate explicit instructions for generating scientifically coherent text with natural variations of formal chemical nomenclature, encouraging the model to use abbreviated forms, common names, or alternative notations typically found in research publications.

3. Entity Variation Control: The prompt design encourages linguistic variation in entity mentions while preserving scientific precision. This produces diverse representations of the same underlying entities, such as using full chemical names alongside abbreviations or alternative nomenclature systems, mirroring the variability found in authentic scientific literature.

4. Distant Supervision for NER: For NER-focused generation, we save the initial sample information tuples selected from PoLyInfo and then align entities in these tuples to the text generated by Llama. This alignment process

locates where each entity from the original tuples appears in the generated text, creating annotations based on these alignments rather than using dictionaries or rule-based entity extraction.

5. Generation Scale: We generated up to 20,000 prompts for this task, resulting in approximately 183,000 synthetic sentences. Our experiments show that moderate volumes (30k-50k sentences) yield optimal improvements, with larger quantities showing diminishing returns.

For EAE-focused synthetic data generation, we developed an enhanced methodology incorporating explicit entity tagging:

1. Strategic Sample Selection: We use the same approach as in NER generation, selecting and combining sample information tuples from PoLyInfo based on common polymer names and property names when available. This creates more complex relationships for the model to learn, though samples without common elements are still utilized. Our focus on polymer-relevant entities addresses the limitations in available high-quality resources for other entity types in materials science. As with NER generation, CONDITION and CHAR_METHOD information may be absent for some property entries.

2. Advanced LLM Selection: We employed state-of-the-art models (Claude Sonnet 3.7 Thinking and GPT-4.1) specifically for this task due to their superior instruction-following capabilities and domain knowledge representation.

3. Multi-Template Approach: We created three complementary prompt templates with increasing levels of sophistication:

- Template 1 (T1): Basic structured template focusing on clear entity relationships
- Template 2 (T2): Enhanced template designed to mimic general materials science writing patterns, incorporating domain-specific terminology, discourse structures, and citation styles commonly found in materials science literature
- Template 3 (T3): Template that leverages actual PolyEE training corpus abstracts as structural guides, replacing original entities with new content while preserving the linguistic patterns, rhetorical structures, and argumentative flows of real-world polymer science text

We generated up to 20,000 prompts for each template variant. Templates T2 and T3 proved particularly effective at generating text that closely resembles authentic scientific writing while maintaining explicit entity tagging.

4. Explicit Entity Tagging and Tuple Generation: The LLMs apply XML-style tags to entities directly in the generated text along with structured event tuples and abbreviation pairs.

5. Three-Stage Distant Supervision Alignment: We transform these self-annotated paragraphs into standardized BRAT format annotations through:

a. Initial Entity and Event Extraction: Parsing tagged entities and event tuples from LLM outputs

b. Abbreviation and Relation Refinement: Handling abbreviations, entity relationships, and event rewiring

c. Context-Sensitive Refinement: Applying linguistic analysis to optimize argument assignments based on sentence-level proximity and domain constraints

This alignment process transforms LLM outputs into high-quality, consistent annotations that adhere to BRAT format specifications and domain-specific constraints, effectively bridging the gap between formal database knowledge and realistic scientific text representation.

G EAE Performance

Table 13 presents the results of TagPrime-C with various encoders for EAE on the PolyEE test set. DeBERTa-v3-large achieves the highest performance with 75.93% F1 score for Argument Classification (AC) and 68.28% F1 score for AC-attached. Domain-specific encoders like MatBERT (74.81% AC F1) and PureMechBERT variants show competitive performance, while general language models like BERT-large exhibit lower scores (65.76% AC F1). The results demonstrate the significant impact encoder selection has on EAE performance.

Table 14 compares different EAE methods on the PolyEE test set using gold triggers. Classification-based approaches like TagPrime-C and TagPrime-CR consistently outperform generation-based methods such as DEGREE, X-GEAR, and PAIE. TagPrime-C with DeBERTa-v3-large achieves the best results with 75.93% F1 on AC and 68.28% on AC-attached, while DEGREE models show higher precision but notably lower recall. The performance gap between AC and AC-attached metrics highlights the challenge of correctly linking arguments to their appropriate triggers in materials science text.

Method	Architecture	AC			AC-attached		
		P	R	F1	P	R	F1
TagPrime-C (Hsu et al., 2023)	**DeBERTa-v3-large**	80.39	**71.93**	**75.93**	72.07	**64.86**	**68.28**
	SciBERT	79.61	71.48	75.32	70.12	64.05	66.95
	MatBERT	80.24	70.07	74.81	**72.24**	61.89	66.67
	PureMechBERT-cased-squad	**80.49**	69.72	74.72	67.06	62.16	64.52
	MechBERT-cased-squad2	78.52	70.77	74.44	65.27	62.97	64.10
	MatSciBERT	77.61	70.77	74.03	67.88	60.54	64.00
	PureMechBERT-cased-squad2	77.65	69.72	73.47	66.57	61.89	64.15
	MechBERT-cased-squad	76.54	70.07	73.16	60.74	61.89	61.31
	MaterialsBERT	78.57	65.85	71.65	70.36	58.38	63.81
	RoBERTa-large	76.05	63.73	69.35	66.35	57.03	61.34
	BERT-large	72.96	59.86	65.76	61.13	52.70	56.60

Table 13: Results for EAE on PolyEE test set using TagPrime-C with different encoders.

Method	AC			AC-attached		
	P	R	F1	P	R	F1
TagPrime-C (DeBERTa-v3-large) (Hsu et al., 2023)	80.39	**71.93**	**75.93**	72.07	**64.86**	**68.28**
TagPrime-CR (DeBERTa-v3-large) (Hsu et al., 2023)	79.84	69.72	74.44	70.46	61.89	65.90
DEGREE (BART-large) (Hsu et al., 2022)	76.92	57.25	65.65	68.95	51.04	58.66
DEGREE (T5-large) (Hsu et al., 2022)	**80.65**	56.31	66.31	**73.53**	53.19	61.73
X-GEAR (BART-large) (Huang et al., 2022)	74.32	62.98	68.18	65.99	57.91	61.69
X-GEAR (T5-large) (Huang et al., 2022)	76.37	62.61	68.81	69.46	58.87	63.72
PAIE (BART-large) (Ma et al., 2022)	72.59	66.20	69.24	66.67	58.38	62.25
PAIE (T5-large) (Ma et al., 2022)	73.36	66.90	69.98	66.98	58.65	62.54

Table 14: Results for EAE on PolyEE test set given gold triggers.

H EAE Synthetic Data

We selected GPT-4.1 and Claude 3.7 Sonnet Thinking as the primary LLMs for EAE synthetic data generation based on the demonstrated superior performance of their closely related versions in scientific domain tasks. This selection is supported by recent research from Bajan et al. (2025), who conducted a comprehensive evaluation of 15 different LLMs on the MatSciQA benchmark. Their study revealed that GPT-4o consistently achieved superior accuracy across four distinct question categories specifically designed to assess specialized materials science domain knowledge. Similarly, Claude models demonstrated exceptional performance in scientific reasoning tasks, making these advanced LLMs particularly well-suited for generating high-quality synthetic data in specialized scientific domains like materials science and polymer property extraction.

Table 15 presents a comprehensive comparison of model performance when trained using different combinations of gold standard and synthetic data for EAE. The table evaluates various training strategies, including gold-only baseline, combined training with different amounts of synthetic data (1,000-5,000 examples), and pre-training followed

Training Strategy	Synthetic Data	Template	AC			AC-attached		
			P	R	F1	P	R	F1
Gold only (baseline)	0	N/A	80.39	71.93	75.93	72.07	64.86	68.28
Combined Training 'train' + synthetic	1,000 (Claude)	T2	80.47	72.28	76.06	**74.62**	**65.95**	**70.01**
		T3	**82.54**	**72.98**	**77.47**	72.54	65.68	68.94
	1,000 (GPT)	T2	80.24	69.82	74.67	71.82	64.05	67.71
		T3	**82.33**	71.92	76.77	73.03	65.14	68.86
	2,000 (GPT)	T2	82.17	**74.39**	**78.08**	**76.99**	**70.54**	**73.62**
		T3	81.54	74.38	77.80	76.22	67.57	71.63
	5,000 (GPT)	T2	79.62	72.62	75.96	73.89	67.31	70.45
		T3	82.10	74.04	77.86	75.07	68.38	71.57
Pre-train → Fine-tune	1,000(GPT)	T2	79.01	72.64	75.69	71.26	65.67	68.35
		T3	80.61	**74.37**	77.36	**76.85**	67.30	**71.76**
	2,000 (GPT)	T2	80.38	73.32	76.69	72.84	65.94	69.22
		T3	80.31	72.97	76.46	74.12	68.10	70.98
	5,000 (GPT)	T2	81.64	73.32	77.26	74.70	67.84	71.10
		T3	82.93	71.58	76.84	76.03	65.13	70.16
	10,000 (GPT)	T2	82.73	72.27	77.15	74.13	**68.92**	71.43
		T3	82.47	72.61	77.23	75.08	66.76	70.67
	20,000 (GPT)	T2	81.96	73.33	**77.41**	73.26	68.11	70.59
		T3	**83.33**	71.91	77.20	75.00	64.84	69.55
	42,000 (All)	T2&T3	**83.13**	72.63	**77.53**	75.95	64.85	69.96

Pre-train: 90% train/10% val, 10 epochs; Fine-tune (FT): 80% train/10% val/10% test, 90 epochs
GPT refers to GPT-4.1; Claude refers to Claude 3.7 Sonnet Thinking; All refers to GPT & Claude
Prompt template 2 (T2): designed to mimic general materials science writing
Prompt template 3 (T3): leverages actual abstracts from PolyEE as templates to better replicate the style of real-world text

Table 15: Impact of synthetic data on n-ary property extraction performance across different training strategies and data volumes.

Entity Type	Gold only			Combined Training		
	P	R	F1	P	R	F1
POLYMER	81.13	**88.00**	**84.42**	**82.32**	84.55	83.42
PROP_NAME	78.39	**88.48**	**83.12**	**85.20**	79.16	82.07
PROP_VALUE	81.21	**87.58**	**84.28**	**85.19**	76.16	80.42
CONDITION	66.27	**70.89**	**68.50**	**70.16**	55.41	61.92
CHAR_METHOD	85.26	**95.86**	**90.25**	**89.16**	87.57	88.36

Table 16: Impact of EAE-focused synthetic data on NER task.

Relation Type	Combined Training		
	P	R	F1
has_property	77.46	71.55	74.39
has_value	90.14	94.12	**92.09**
has_condition	81.44	61.24	69.91
characterized_by	75.40	74.80	75.10
abbreviation_of	96.09	93.18	**94.61**

Table 17: Impact of EAE-focused synthetic data on RE task.

by fine-tuning approaches (1,000-42,000 examples).

The results demonstrate that incorporating moderate amounts of synthetic data (particularly 2,000 GPT-generated examples) yields the best performance, with a significant improvement over the gold-only baseline, achieving AC F1 scores of 78.08% (vs. 75.93%) and AC-attached F1 scores of 73.62% (vs. 68.28%). Interestingly, the table shows that increasing synthetic data beyond 2,000 examples doesn't yield further improvements, suggesting that data quality is more important than quantity. The templates (T2 and T3) represent different approaches to synthetic data generation, with T2 generally showing stronger performance in the combined training setting.

Entity Type / No. of sent.	MatSciNERE (train) 4,878 sent	5k sent	10k sent	15k sent	30k sent	50k sent	100k sent	150k sent	~183k sent
POLYMER	84.42	85.19	84.38	86.42	85.63	85.71	86.61	83.70	82.69
PROP_NAME	83.12	83.39	83.51	83.93	83.99	84.26	84.61	84.52	84.69
PROP_VALUE	84.28	88.22	86.84	87.54	88.96	89.19	88.14	87.58	85.52
CONDITION	68.50	69.46	71.57	67.09	69.59	68.48	68.52	69.93	67.08
CHAR_METHOD	90.25	89.89	91.43	90.96	89.01	89.14	88.33	89.71	89.77
Overall	82.87	84.07	83.90	84.19	84.34	**84.69**	84.26	83.66	83.33

Table 18: Impact of synthetic data amount on NER performance (F1 scores) by entity type.

Additional Investigations Tables 16 and 17 present the impact of EAE-focused synthetic data on NER and RE tasks as additional investigations to better understand data characteristics.

As additional investigation, Table 16 compares "Gold only" versus "Combined Training" approaches for NER across five key entity types. Using the W2NER+MatSciBERT model trained on supervised data combined with synthetic examples, the results show that precision consistently improves across all entity types while recall generally decreases. This precision-focused improvement is particularly beneficial for systems that prioritize high-confidence predictions over coverage, making the approach valuable for applications requiring high precision.

As additional investigation, Table 17 displays the performance of the Combined Training approach on RE tasks. Using ATLOP+DeBERTa-v3-large trained on supervised plus synthetic data, the recall is notably decreased compared to gold-only training. This decline is reasonable since LLMs introduce numerous new entities and potentially new relations in the synthetic data. However, the F1 scores for two relation types '*has_value*' (92.09) and '*abbreviation_of*' (94.61) remain exceptionally strong. These high-performing relation types directly motivated the development of Approach C (LLM-Guided Assembly) in the unified framework, which leverages these reliable relation predictions to guide structured extraction.

I NER Synthetic Data

Table 18 presents the detailed results of NER Synthetic Data experiments, showing F1 scores for various entity types across different synthetic data volumes. The table demonstrates how performance changes when training with MatSciNERE's baseline 4,878 sentences versus adding increasing amounts of synthetic data (from 5k to ~183k sentences). Performance for most entity types improves with moderate synthetic data volumes, with the best overall F1 score (84.69%) achieved at 50k sentences. PROP_VALUE shows the most dramatic improvement (from 84.28% to 89.19%), while POLYMER and PROP_NAME show modest gains. Performance peaks at 50k sentences and slightly declines with larger synthetic datasets (~183k), suggesting that optimal synthetic data size is around 30k-50k sentences (approximately 3k-5k paragraphs) for materials science NER tasks.

J Prompt Design for LLM-Guided Extraction

Figure 6 illustrates the prompt template used in our LLM-Guided Assembly approach. The prompt integrates a scientific paragraph with pre-identified entity mentions and their character offsets from the NER module. It includes "*Known abbreviations*" based on high-confidence '*abbreviation_of*' relation predictions and "*Entity mapping for shortened entities*" to handle lengthy mentions that might challenge LLM processing. The prompt targets a specific property-value pair identified through the '*has_value*' relation and provides structured instructions for generating complete 5-argument tuples. This approach effectively constrains the LLM while leveraging its inferential capabilities to extract complex n-ary property information from scientific text.

Paragraph:
In a recent study on thermoplastic elastomers, we investigated the properties of two distinct copolymers: poly[(acrylic acid)-co-(butyl acrylate)] (PAA-BA) with a chemical formula C3H4O2/C7H12O2, and poly{[4-(octadecyloxy)benene-1,3-diamine;4,4'-sulfonyldianiline]-alt-(5,5'-biisobenzofuran-1,1',3,3'-tetrone)} (ODA-PMDA-BTDA) with a chemical formula C40H46N2O5/C28H14N2O6S. A notable difference between these copolymers lies in their interfacial properties. PAA-BA exhibited an interfacial tension of 2.915 dyn/cm, measured using the pendant drop method at a temperature of 25°C. In contrast, ODA-PMDA-BTDA displayed a significantly higher surface energy due to its aromatic and heterocyclic constituents, resulting in a surface tension of 22.90 N/m, as determined by the Wilhelmy plate technique under identical temperature conditions. Thermal analysis revealed distinct differences in their glass transition temperatures (Tg) and thermal decomposition behavior. PAA-BA exhibited a Tg of -21.00 K, measured using thermomechanical analysis (TMA) at a heating rate of 10°C/min and a frequency of 0.1 Hz. Conversely, ODA-PMDA-BTDA demonstrated exceptional thermal stability, with an onset decomposition temperature of 401.9°C and a weight loss of only 5.000% during thermogravimetric analysis (TG) under nitrogen atmosphere at a heating rate of 20°C/min in vacuum conditions. These findings suggest that the incorporation of rigid aromatic moieties in ODA-PMDA-BTDA enhances its thermal stability, whereas PAA-BA's aliphatic backbone contributes to its lower Tg and increased flexibility. The observed differences in their interfacial properties and thermal behavior underscore the importance of tailoring copolymer composition for specific applications.

Known entity mentions (non-exhaustive list):
POLYMER: ODA-PMDA-BTDA [312, 325] (ID: T5)
…
POLYMER: PAA-BA [965, 971] (ID: T26)
POLYMER: ODA-PMDA-BTDA [594, 607] (ID: T17)
CHAR_METHOD: pendant drop method [535, 554] (ID: T3)
CONDITION: at a temperature of 25°C [555, 579] (ID: T13)
…

Known abbreviations (non-exhaustive list):
PAA-BA: poly[(acrylic acid)-co-(butyl acrylate)]
…
TG: thermogravimetric analysis

Entity mapping for shortened entities:
poly{[4-(octadecyloxy)benene-1,3-diamin...f6ff0199: poly{[4-(octadecyloxy)benene-1,3-diamine;4,4'-sulfonyldianiline]-alt-(5,5'-biisobenzofuran-1,1',3,3'-tetrone)}
at a heating rate of 10°C/min and a fre...f6708722: at a heating rate of 10°C/min and a frequency of 0.1 Hz
under nitrogen atmosphere at a heating ...4bfdaa67: under nitrogen atmosphere at a heating rate of 20°C/min in vacuum conditions

For the following PROP_NAME and PROP_VALUE pair, provide the completed event tuple(s) (POLYMER, PROP_NAME, PROP_VALUE, CONDITION, CHAR_METHOD) using the correct entity mentions and their offsets from the list above, or by identifying new relevant entities from the text. Use the shortened versions of entities where applicable. You may also modify existing CONDITION or CHAR_METHOD entities if a more appropriate version is found in the text. Maintain the given PROP_NAME and PROP_VALUE relationship without changes. Include all relevant POLYMER, CONDITION, and CHAR_METHOD entities that apply to the given PROP_NAME and PROP_VALUE pair based on the text. Use the known abbreviations to help identify and link related concepts, but be aware that there may be other abbreviations in the text not listed.

PROP_NAME: interfacial tension [479, 498] (ID: T10), PROP_VALUE: 2.915 dyn/cm [502, 514] (ID: T11)

Instructions:
1. Provide a separate event tuple for each unique combination of POLYMER, CONDITION, and CHAR_METHOD that applies to the given PROP_NAME and PROP_VALUE pair.
2. Each event tuple must have exactly 5 arguments: (POLYMER, PROP_NAME, PROP_VALUE, CONDITION, CHAR_METHOD).
3. Use the exact entity mentions and offsets from the list above when applicable. If a more appropriate entity is found in the text but not listed, add it with its correct offsets.
4. If no relevant entity is found for POLYMER, CONDITION, or CHAR_METHOD, use an empty list [] for that argument.
5. Do not combine multiple entities of the same type into a single event. Instead, create separate event tuples for each combination.
6. For each relevant POLYMER, CONDITION, or CHAR_METHOD, create a separate event tuple, even if other arguments remain the same.
7. Do not add explanations or additional text.
8. Maintain the given PROP_NAME and PROP_VALUE relationship without changes.
9. Verify the correctness of each tuple based ONLY on explicit expressions in the raw text that are directly related to the given PROP_NAME and PROP_VALUE pair.
10. Only include information that is directly stated and clearly associated with the specific PROP_NAME and PROP_VALUE pair in the text.
11. Be aware of both listed and unlisted abbreviations in the text, using them to help identify relationships between concepts.
12. Ensure that you have considered all possible POLYMER, CONDITION, and CHAR_METHOD entities mentioned in the text that are relevant to the given PROP_NAME and PROP_VALUE pair.
13. Generate all possible combinations of relevant POLYMER, CONDITION, and CHAR_METHOD entities, including cases where some entity types may be empty ([]).
14. When referring to entities, use the shortened versions provided in the entity mentions list.

Example response format:
(['POLYMER1 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION1 [start, end]'], ['CHAR_METHOD1 [start, end]'])
(['POLYMER1 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION1 [start, end]'], ['CHAR_METHOD2 [start, end]'])
(['POLYMER1 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION2 [start, end]'], ['CHAR_METHOD1 [start, end]'])
(['POLYMER1 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION2 [start, end]'], ['CHAR_METHOD2 [start, end]'])
(['POLYMER2 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION1 [start, end]'], ['CHAR_METHOD1 [start, end]'])
(['POLYMER2 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION1 [start, end]'], ['CHAR_METHOD2 [start, end]'])
(['POLYMER2 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION2 [start, end]'], ['CHAR_METHOD1 [start, end]'])
(['POLYMER2 [start, end]'], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', ['CONDITION2 [start, end]'], ['CHAR_METHOD2 [start, end]'])
([], 'PROP_NAME [start, end]', 'PROP_VALUE [start, end]', [], [])

Figure 6: Example prompt for LLM-Guided Assembly showing how high-confidence entity and relation predictions guide the extraction of n-ary property tuples.

K Practical Applications

Rather than presenting a single best method, our framework acknowledges the diverse requirements of real-world extraction scenarios and offers multiple viable pathways. This flexibility allows users to select the approach that best aligns with their specific constraints and priorities:

1. When computational resources are limited or offline deployment is required, the EAE-based 3approaches offer strong performance without external dependencies.

2. When maximum accuracy is crucial and LLM resources are available, the LLM-guided approach provides most effective results.

3. When extraction needs to focus on specific relation types or entity categories, the RE-Composition approach offers more granular control and interpretability.

The framework also enables ensemble methods and hybrid approaches that can further enhance performance. For instance, high-confidence extractions from multiple pathways could be combined, or different approaches could be deployed based on document characteristics or extraction requirements.