ZSEE: A Dataset based on Zeolite Synthesis Event Extraction for Automated Synthesis Platform

Song He[1], Xin Peng[1], Yihan Cai[2], Xin Li[2]*, Zhiqing Yuan[2], Wenli Du[1]*, Weimin Yang[2]*
[1]State Key Laboratory of Industrial Control Technology, East China University of Science and Technology
[2]State Key Laboratory of Green Chemical Engineering and Industrial Catalysis,
Sinopec Shanghai Research Institute of Petrochemical Technology
{songhe, xinpeng}@ecust.edu.cn, {caiyh, lixin, yuanzq}.sshy@sinopec.com
wldu@ecust.edu.cn, yangwm.sshy@sinopec.com

Abstract

Automated synthesis of zeolite, one of the most important catalysts in chemical industries, holds great significance for attaining economic and environmental benefits. Structural synthesis data extracted through NLP technologies from zeolite experimental procedures can significantly expedite automated synthesis owing to its machine readability. However, the utilization of NLP technologies in information extraction of zeolite synthesis remains restricted due to the lack of annotated datasets. In this paper, we formulate an event extraction task to mine structural synthesis actions from experimental narratives for modular automated synthesis. Furthermore, we introduce ZSEE, a novel dataset containing fine-grained event annotations of zeolite synthesis actions. Our dataset features 16 event types and 13 argument roles which cover all the experimental operational steps of zeolite synthesis. We explore current state-of-the-art event extraction methods on ZSEE, perform error analysis based on the experimental results, and summarize the challenges and corresponding research directions to further facilitate the automated synthesis of zeolites. The code is publicly available at https://github.com/Hi-0317/ZSEE.

1 Introduction

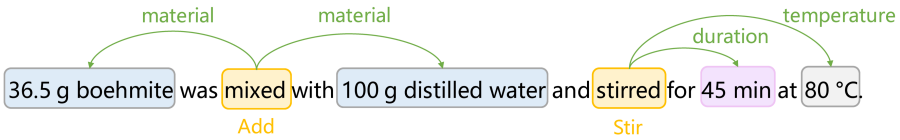
Artificial intelligence is accelerating the autonomous unmannedness of the chemical industry (Burger et al., 2020). As one of the most widely used catalysts in chemical industries, the automated synthesis of zeolite can break the limitations of traditional synthesis process in terms of constrained experimental time of researchers, effectively improving the experimental efficiency (Moliner et al., 2019).

The first step in carrying out automated synthesis of zeolite is to enable the machine to understand the experimental procedures. Recently, there has been a massive increase in the literature and patents of zeolite synthesis experiments, which have documented considerable chemical reaction steps. These synthesis steps can guide the synthesis of specific zeolites and enable the exploration of new zeolites. Natural language processing (NLP) techniques allow automatic mining of these synthesis data from materials science literature on a large scale (Kim et al., 2020; Raccuglia et al., 2016; Kim et al., 2017; Kononova et al., 2019). The purpose of conducting such analyses falls into two categories: (1) to get deeper scientific understanding of materials synthesis (Krallinger et al., 2017); (2) to implement further research on automated synthesis planning, e.g., enabling robots to perform certain experiments (Kim et al., 2019; Rohrbach et al., 2022). But for intelligent machines, there is a huge gap between unstructured records of zeolite synthesis procedures and structured programming languages in terms of semantic understanding. Therefore, it is still a challenge to analyze the unstructured experimental narratives and automatically extract machine-readable structured synthesis steps while implementing automated synthesis.

Previous works have leveraged NLP techniques to extract reaction steps from organic and inorganic chemical synthesis procedures, most of which mostly used named entity recognition (NER) or relationship extraction (RE) to extract chemical entities and inter-entity relationships, ignoring the complete synthesis steps. Aiming to extract structured synthesis information from experimental narratives for automated synthesis platform, Mehr et al. (2020) summarized the chemical synthesis steps and designed a rule-based model to extract the details of these synthesis steps. Vaucher et al. (2019) constructed a sequence-to-sequence deep learning model to convert unstructured experimental narratives into predefined action sequences. However, the machine translation approach failed to yield fine-grained structured experimental actions. To

* Corresponding Author

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1791–1808



ID	Event type	Trigger	Argument role	Arguments
1	Add	mixed	material	36.5 g boehmite
			material	100 g distilled water
2	Stir	stirred	duration	45 min
			temperature	80 °C

Figure 1: Annotation example. Below is the extracted structured information on zeolite synthesis steps.

promote the research on deep learning-based methods for chemical information extraction, Mysore et al. (2019) introduced a dataset of 230 synthesis procedures, where the operations and entities (e.g., materials and conditions) were annotated. He et al. (2020) introduced event extraction task and released a dataset for chemical event extraction that only defined two event types, which failed to distinguish between the different experimental steps. Thus, a dataset with comprehensive fine-grained synthesis information of experimental steps was urgently needed to perform automated and modular synthesis.

In this paper, we formulate an event extraction (EE) task to extract fine-grained structured information of synthesis steps directly from zeolite experimental procedures. Based on this task, we introduce a novel dataset, ZSEE, which contains nearly 5000 sentences extracted from the literature on the synthesis process of different types of zeolites. Specifically, we summarize all experimental steps of zeolite synthesis and define 16 refined synthesis actions (e.g., **Add** and **Stir**) and the corresponding 13 synthesis properties (e.g., *material*, *temperature* and *duration*). The event triggers and arguments of each sentence in the ZSEE are annotated with text spans. Both humans and intelligent machines can easily capture these synthesis details from our annotations (e.g., stir at 80 °C for 45 min). An annotated example is shown in Figure 1.

To evaluate the performance of the state-of-the-art (SOTA) EE methods for the zeolite synthesis event extraction task, we conduct extensive experiments in ZSEE. We implement two main classes of EE methods to evaluate event detection (ED) and event argument extraction (EAE) tasks in ZSEE, namely classification-based and generation-based methods. The classification-based method performs best for ED with the exact match F1 score of 92.46%. The results of these methods are all actually competitive on the ED task. Since triggers of zeolite synthesis events are with single expressions, deep learning-based models can detect these words well. The generation-based method achieves better results for EAE with the exact match F1 score of 68.73%. Observing the disparity between ED and EAE results, we further explore the SOTA EAE methods, which achieve 5.44% gains over the generation-based EE method. Recalling the event arguments contain key information for accurate synthesis planning, we urge the development of a better model for event argument extraction of zeolite synthesis.

Our contributions can be summarized as follows:

- We formulate a novel event extraction task for zeolite synthesis and provide a fine-grained event schema covering all synthesis steps in practical experiments for automated synthesis.
- We present ZSEE, a new zeolite synthesis event extraction dataset. ZSEE consists of nearly 9000 zeolite synthesis events. To the best of our knowledge, ZSEE is the largest dataset for automated synthesis to date.
- We have conducted extensive experiments on ZSEE to evaluate the performance of current SOTA event extraction methods, formed a benchmark for zeolite synthesis event extraction, and presented challenges for future research in this area.

2 Related work

2.1 Chemical synthesis corpus

The scientific literature has been the key source for researchers to obtain information about the synthesis process of specific materials. Except for constructing structured databases (e.g., Reaxy and

SciFinder), researchers designed some information mining tools to automatically extract chemical information. ChemicalTagger (Hawizy et al., 2011) and ChemDataExtractor (Swain and Cole, 2016) were proposed to capture the entities related to chemical synthesis reactions and the relationships between these entities.

With the help of deep learning-based methods, the efficiency and accuracy of information extraction can be greatly improved, where highly accurate and well-labeled training data is indispensable. Mysore et al. (2019) introduced a dataset of 230 inorganic material synthesis procedures, which annotated synthesis operations, typed arguments and their relationships. Kononova et al. (2019) proposed a dataset for inorganic solid-state synthesis recipes and designed 5 categories of effective operations. CHEMU (He et al., 2020) provided a corpus of labeled synthesis events, which defined experimental events as reaction step events and workup events, representing the conversion of starting materials into products and the separation and purification of products respectively. Although these datasets introduced the concept of events to represent synthesis steps holistically, their constrained event schema failed to distinguish between different synthesis operations to perform modular experiments. We annotate synthesis procedures with more specific and fine-grained event definitions of different synthesis operations based on practical zeolite synthesis experiments. The annotated results are explicitly grouped according to experimental steps, allowing humans and machines to directly perform corresponding modular experiments based on fine-grained events.

2.2 Chemical information extraction

Most current research on chemical information extraction has focused on NER (Wang et al., 2021; Panapitiya et al., 2021; Friedrich et al., 2020), RE (Xu et al., 2023) or the combination of both (Yang et al., 2022), with less research on extracting structured chemical synthesis information through event extraction. ChemRxnExtractor (Guo et al., 2022) formulated the chemical reaction extraction as a structure prediction task, which identified the products through NER and further extracted the reaction roles through RE. Such pipeline chemical information extraction methods fail to provide information about the complete synthesis steps. Thus, we formulate the EE task for zeolite synthesis and support the research with fine-annotated data.

End-to-end extractive and generative approaches achieved better performance in other domains (Song et al., 2023). Li et al. (2020); Du and Cardie (2020) converted EE into a multi-round question-and-answer task by designing different questions to obtain triggers and event arguments. Liu et al. (2022) and Hsu et al. (2022) designed specific templates for each event type and converted EE into conditional generation task, which usually leveraged the pre-trained models (PLM). Introducing prior knowledge of templates allowed PLM to generate the target triggers and arguments more accurately. We mainly explored these extractive and generative event extraction approaches on ZSEE, designing specific questions and templates for different zeolite synthesis events.

3 Dataset Construction

3.1 Task definition and schema

Our proposed zeolite synthesis event extraction task aims to extract information about the synthesis steps, including the experimental actions and corresponding properties. Figure 1 shows an example of the task. The novel task can also be divided into event detection and event argument extraction (Li et al., 2022). ED aims to identify the synthesis actions (i.e., triggers) and specify the action types in the experimental text. EAE aims to extract the properties corresponding to actions (i.e., event arguments) and specify the role relationship between triggers and arguments in a sentence.

We summarize the synthesis actions that occur in the process of zeolite synthesis and regard them as traditional event types, such as **Add** and **Stir**. The action properties are considered as event arguments, such as *temperature* and *duration* corresponding to **Stir**. More specifically, in this task, we design a set of synthesis actions with predefined properties based on zeolite synthesis narratives, which cover all operations of conventional zeolite synthesis. The event schema contains 16 event types and 13 argument roles.

We have detailed the three types of synthesis actions that occur frequently in ZSEE:

Add indicates that some materials are added to the container at a specific temperature, with arguments specifying *material*, *temperature* and *container*.

Stir means that the mixture is stirred with full contact for a while, whose arguments include *duration*, *temperature*, stirring *rate* and the *sample*.

Annotation	Fleiss' Kappa
Span-level labels	0.89
Trigger labels	0.91
Argument labels	0.83

Table 1: Inter-annotator agreements in ZSEE. Span-level labels mean the overall agreements.

	Train	Dev	Test	Total
Sentences	3931	504	526	4961
Events	6966	935	972	8873
Arguments	11353	1475	1503	14331
Tokens	26.44	26.49	26.19	26.27

Table 2: Key statistics of ZSEE. Tokens mean the average number of tokens per sentence.

Wash describes that the product is washed several times with some solvent, with arguments specifying *solvent*, *times*, and the *sample*.

The event arguments are detailed information to the synthesis steps, e.g., the **Stir** action is further supplemented by duration, temperature and stirring speed to form a complete synthesis step. For the filtration and centrifugation actions of the zeolite synthesis, we define them as **Particle Recovery** for the sake of professional presentation. Details of all event types and arguments and their corresponding descriptions are shown in Appendix A.

3.2 Data annotation

3.2.1 Data Collection

To standardize the event extraction task for zeolite synthesis, we collect publicly available English literature containing specific synthesis steps from the database of the University Library, which is created under the agreement with scientific publishers such as Springer and Elsevier. We extract synthesis step-related passages from over 1000 scientific documents and split them into sentences. We manually annotate these sentences and preserve nearly 5000 sentences after removing duplicates. Note that the DOI of each sentence was always recorded to ensure that each annotated text could be traced back to the original data.

3.2.2 Annotation Process

We employed 11 graduate students in chemistry and computer science to carry out the annotation work, three of whom acted as reviewers, checking the quality and consistency of the annotations and determining the final annotation results. Note that the annotators we hired were professionally trained and the reviewers all had extensive experience in zeolite synthesis experiments.

Each sentence in ZSEE was annotated by two annotators and one reviewer. We used DoTAT (Lin et al., 2022) to annotate the zeolite synthesis data. Our annotation process followed strict annotation guidelines to ensure that all experimental steps in each sentence were annotated. The action type was first determined by specifying the span of the trigger. We then analyzed the specific properties associated with the identified event throughout the sentence. Once both annotators had completed their annotations, the review function provided by DoTAT automatically merged the two annotation results. The reviewer could check the inconsistent information and adjust the annotation results to the best. Besides, The reviewers constantly clarified terms that raised questions during the review process to ensure the quality of our annotations. Annotation guidelines and further annotation details are provided in Appendix B.

The annotation results were stored in a JSON file which structurally recorded synthesis action events, including the action type, triggers, and spans and roles of arguments. Based on the annotation results, researchers and intelligent machines could obtain all the structured information about the zeolite synthesis in the experimental procedures.

3.2.3 Data Validation

We report the inter-annotator agreements (IAA) between all three expert reviewers based on a collection of 200 zeolite synthesis sentences in Table 1, where the numbers we report are Fleiss' Kappa scores. The overall agreement on the span-level labels is 0.89, which proves the quality of our annotations. We also observe that the agreements on labels correspond to triggers and arguments are different. The experimental properties (i.e., argument roles) tend to be more ambiguous and the annotators show more subjectivity when annotating these ambiguous arguments such as *material*, which causes a lower agreement of the arguments.

3.3 Dataset analysis

ZSEE contains a total of 4961 sentences, including 8873 annotated events and 14331 zeolite synthesis arguments. To the best of our knowledge, ZSEE is the first and largest dataset in the domain of zeolite

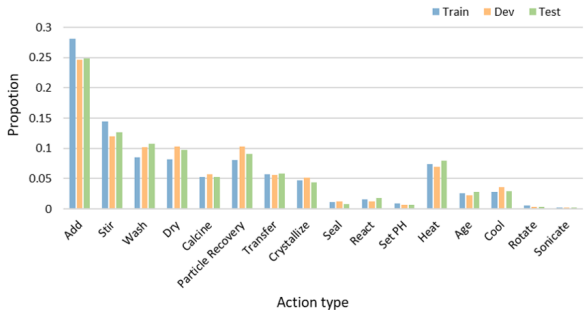


Figure 2: Data distribution of event type on the training, validation and test set.

synthesis. We also compare ZSEE with previous event extraction datasets, including the common domain datasets ACE2005 (Doddington et al., 2004) and ERE-EN, the historical event dataset BRAD (Lai et al., 2021), and the pharmacovigilance event dataset PHEE (Sun et al., 2022). The statistics details are shown in Appendix C, where ZSEE shows strong competitiveness.

We divide the training, validation and test set in a ratio of 8:1:1. Table 2 lists the key statistics, including the number of sentences, events and arguments in subsets. We keep the number of event types balanced across subsets to ensure the consistency of data distributions. Figure 2 shows the proportion of each event type, which indicates that the data distributions are very similar across the three subsets. The exact number of each event type is recorded in Appendix C.

4 Experiments

Inspired by PHEE (Sun et al., 2022), we explore the performance of two mainstream classes of event extraction methods (classification-based and generation-based methods) on ZSEE to reveal the challenges in zeolite synthesis event extraction. Specifically, to achieve better performance, we design questions and templates with experimental logic of zeolite synthesis for these methods respectively.

4.1 Benchmark Methods

Classification-based Method: We primarily evaluate recent classification-based extractive question-and-answer (QA) methods. Inspired by EEQA (Du and Cardie, 2020), we construct a two-stage QA model, where separate questions are designed for the ED and EAE subtasks. The question for ED denotes `Q1`: *What happened in the zeolite synthesis event?* The question for EAE question denotes `Q2`: *What is the <argument> in <trigger>?* Note that the classification-based approach is to identify the trigger first. The arguments are then extracted based on the identified event type. The *<trigger>* is replaced by the identified trigger and the *<argument>* therefore refers to all the argument roles for the predicted event type.

We leverage the pre-trained model BERT (Devlin et al., 2019) to answer the corresponding questions. The model framework is shown in Figure 3, where two separate BERT models are used for the ED and EAE, respectively. The inputs for both models are "`[CLS] <Qi> [SEP] <sentence> [SEP]`", where `[CLS]` and `[SEP]` are placeholds for BERT, `<Qi>` denotes the question defined above, i.e., `Q1` and `Q2`, and `<sentence>` is the source sentence. In the ED task, BERT outputs the probability of the event type for each token in the sentence, thus determining all event triggers and the corresponding event type based on an appropriate threshold. In the EAE task, BERT predicts the start and end offsets of the argument. Previous work proved that different questions will affect the accuracy of the results. Thus, we have conducted experiments with different questions to explore the best setting in Appendix E.

Generation-based Method: The classification-based method could lead to error propagation because the result of EAE depends on the trigger extracted in the first stage. Moreover, the phased extraction of triggers and arguments ignores the potential relationship between them. Thus, We follow DEGREE (Hsu et al., 2022) to construct a joint generation-based model that generates both trigger and event arguments simultaneously. Designing effective prompting templates is the key to the generation-based approach. We design the unified template `Template_trigger` for ED, i.e., *Event trigger is <trigger>*, where *<trigger>* is the placeholder for the trigger to be predicted. To extract the event arguments, specific templates are designed for each event type in the event schema. The templates `Template_args` for the top three most frequently mentioned action types are shown below:

Add: something was added to container at temperature.

Stir: something was stirred at temperature at revolution per minute for some time.

Wash: wash something with something by several times.

The underlined words indicate the arguments

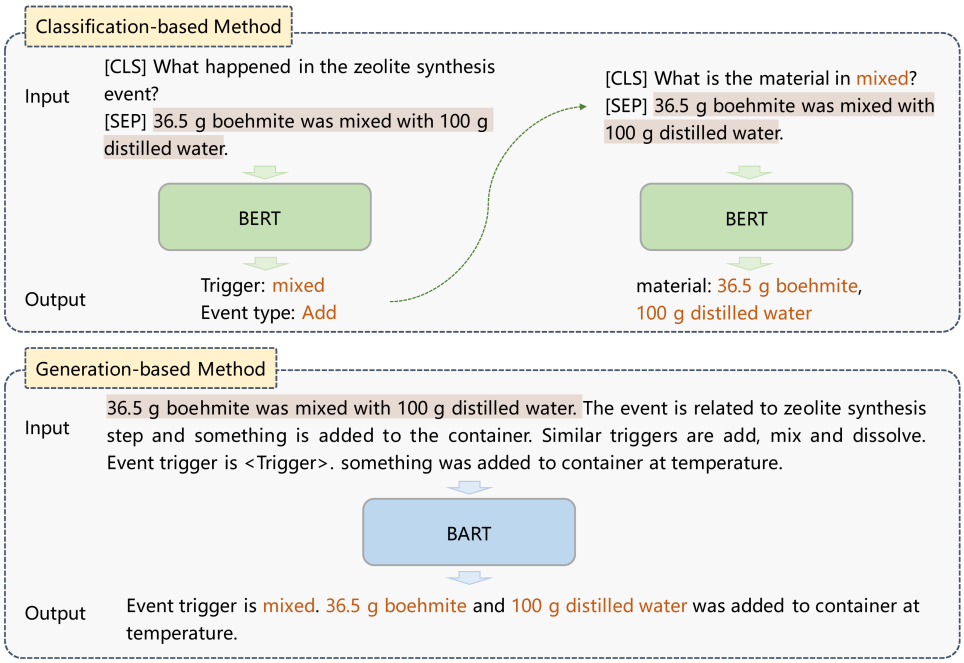


Figure 3: Model framework. The example sentence "36.5 g boehmite was mixed with 100 g distilled water." is highlighted. The illustrations show the question and template settings for classification-based and generation-based methods respectively and their corresponding extraction process.

to be predicted. The model captures the span of text in the source sentence and predicts the specific content to replace these underlined words. The templates of all event types are documented in Appendix E.

We use the pre-trained model BART (Lewis et al., 2020) to generate sentences containing all event information. Specifically, the input of the model is "<Sentence> <Description> <Template_trigger> <Template_args>". Note that <Sentence> denotes the source sentence. <Description> is a complementary description of the given event type, including the event definition and possible trigger words, e.g., the <Description> for **Add** is "The event is related to zeolite synthesis step and something is added to the container. Similar triggers are add, mix and dissolve". The output is then "<Template_trigger> <Template_args>" with the underlined words replaced, as shown in Figure 3, where the event type and argument roles are predicted simultaneously.

EAE Method: Considering the multi-argument characteristic, we further explore the performance of SOTA EAE methods. Specifically, we follow the AMPERE (Hsu et al., 2023) and PAIE (Ma et al., 2022). AMPERE introduces abstract meaning representation (AMR) based on the DEGREE, which generates AMR-aware prefixes for the generative model. PAIE designs extra selectors for the start and end position of the argument span upon the generative model to more accurately identify the arguments. The backbones of the above two methods are the same as the generation-based approach shown in Figure 3.

4.2 Evaluation Metrics

Based on the task definition in Section 3.1, we evaluate the two subtasks ED and EAE separately. (1) For event detection, a trigger is correctly identified if the predicted offset matches the golden offset (Tri-I). If the predicted event type also matches the golden type, the trigger is then correctly classified (Tri-C). (2) For event argument extraction, an argument is correctly identified (Arg-I) if its start and end offset both match the golden offset, and correctly classified (Arg-C) if its role also matches the golden role. We use the same metric that is commonly used in event extraction work (Li et al., 2013), namely the micro-F1 metric.

4.3 Overall Experimental Results

We have designed different questions and templates to discover the most suitable settings for zeolite synthesis event extraction. The results presented in this section are all from the best question and template settings, while the results for the other settings are detailed in Appendix E. Besides, the

Method	Type	PLM	Tri-I	Tri-C	Arg-I	Arg-C
EEQA (Du and Cardie, 2020)	cls	BERT-b	**93.75**	92.43	67.27	66.76
		BERT-l	93.48	**92.46**	68.13	67.86
DEGREE(Hsu et al., 2022)	gen	BART-b	91.08	91.08	68.04	67.70
		BART-l	91.91	91.91	69.21	68.73
AMPERE(Hsu et al., 2023)	gen	BART-b	-	-	71.49	70.94
		BART-l	-	-	72.12	71.70
PAIE(Ma et al., 2022)	gen	BART-b	-	-	74.52	74.01
		BART-l	-	-	**74.58**	**74.17**

Table 3: Overall performance. Note that b denotes the base model and l denotes the large model in column PLM. The best results are highlighted in bold. - indicates no results as AMPERE and PAIE are designed for EAE only.

training details and hyperparameter settings are shown in Appendix D.

Table 3 compares the performance of different SOTA methods on ZSEE. The classification-based method achieves the best performance in event detection, with the exact match F1 score of 92.46%. Although the generation-based method shows a decrease in accuracy, it also still achieves the exact match F1 score of 91.91%. These methods all achieve exciting results on the ED task, which is much higher than their performance on other datasets such as ACE2005 corpus (Doddington et al., 2004). We have analyzed the corpus of zeolite synthesis and found that the representation of trigger for a given synthesis step is relatively single by different authors. In the case of the synthesis event **Dry**, for example, this synthesis action is mostly described by the word "dry" and its different tense and morphological variants. Thus, the large pre-trained language models possess the ability to identify a finite number of triggers accurately. There are definitely also some action types that are documented in a variety of ways, e.g., the synthesis step **Add** is documented as "add, put, charge, dissolve, mix, pour, introduce, etc.".

The generation-based method achieves better results in event argument extraction, with an improvement of 0.87% F1 score compared to the classification-based method with large model. Since the generation-based method introduces more information related to zeolite synthesis through the designed template, the model can focus on the target arguments. However, the results of the current EE method with the best F1 score of 68.73% still cannot support the development of automated synthesis of zeolite because the event arguments contain detailed information about the experimental steps. Thus, we further explore the performance of the SOTA EAE model on ZSEE. Table 3 shows that with the introduction of role-specific selectors and joint prompts, the EAE results achieve promising improvements, with 5.37% and 5.44% F1 gains in Arg-I and Arg-C, respectively.

Event Type	Tri-I	Tri-C	Arg-I	Arg-C
Add	94.55	93.73	55.59	55.42
Stir	97.18	96.37	75.07	75.07
Wash	95.11	94.77	65.09	65.09
Dry	95.03	95.03	83.49	83.16
Set PH*	82.93	82.93	58.33	58.33
Rotate*	87.50	87.50	66.67	66.67
Sonicate*	80.00	80.00	60.00	60.00

Table 4: Results for different event types through the classification-based method with BERT-large. Event types are sorted according to their number. Event types with less than 100 in the training set are marked with *.

Further, in order to explore the performance of the current PLMs on ZSEE, we conduct experiments with pre-trained models of different sizes. The results in Table 3 show that larger PLMs enable better performance than the base PLMs for both ED and EAE tasks.

4.4 Error Analysis and Challenges

We have summarized the common errors of the methods mentioned above on ZSEE and suggested directions that could be investigated and improved in the future.

4.4.1 Challenge of Abstract Expression

We observe that the most frequent error is in the extraction of compounds. Existing methods struggle to achieve accurate extraction of complex chemical entities, especially for abstract representations.

Sentence Type	Tri-I	Tri-C	Arg-I	Arg-C
Short sentences	95.85	94.01	70.21	70.21
Medium sentences	93.89	92.66	69.14	68.79
Long sentences	91.43	90.00	55.85	55.10

Table 5: Results for sentences with different lengths, which are all from the EEQA with BERT-large.

Descriptions of zeolite synthesis often contain additions to the experimental material, such as "deionized water (331.22 g)". The weight of deionized water is supplemented in brackets. Current methods often identify "deionized water" as the material and ignore the information in brackets which is also important for the experiment. Instead, we find that these methods can identify "331.22 g deionized water" very well. When adding dose information or molar mass, some authors also record the company from which the compound is derived. These abstract records cannot be effectively identified. Although there have been calls for uniform writing styles (Kim et al., 2019), descriptions of the zeolite synthesis often vary between authors, with abstract descriptions remaining a challenge for current deep learning-based methods. Considering the powerful semantic understanding performance of PLMs, adding more prompts, such as the description or examples of abstract expressions, might be an effective way to mine the ability of models to address abstract representations.

4.4.2 Challenge of Limited Resource

There is a large imbalance in the sample size for the different event types, as shown in Table 7 in Appendix C. **Add** and **Stir**, the most common chemical synthesis actions, have more than 1000 events in ZSEE. Compared with relatively rare action types such as **Set PH**, the number of **Add** is even tens of times higher. Table 4 presents the extraction results for the top several event types of the highest and lowest number of events with the classification-based method. The results become progressively worse as the amount of data gets smaller. The ED results for event types (such as **Set PH** and **Rotate**) with less than 100 events are all below 90%. Meanwhile, **Sonicate** with the lowest number of events performed 23.16% lower than the best result of **Dry** on the EAE task. Note that although **Add** is the most numerous action type, its results on EAE still need to be improved because the main argument *material* causes the errors in Section 4.4.1.

Event extraction in low-resource scenarios has always been a worthwhile research direction, and zeolite synthesis event extraction is no exception. The amount of each event type can be increased in the future through data augmentation strategies. Besides, introducing contrastive learning or leveraging existing large language models is also the potential direction to improve the few-shot learning ability.

4.4.3 Challenge of Long Sentences

Table 2 shows that the average number of tokens per sentence in ZSEE is approximately 26. However, there are many excessively long sentences in the zeolite synthesis narratives. We classify all sentences into short, medium and long sentences according to the number of tokens, as shown in Table 9 in Appendix C. Short sentences have less than 15 tokens, medium sentences have between 15 and 40 tokens, and long sentences have more than 40 tokens. Table 5 presents the performance of the classification-based method on these three types of sentences. As the sentences become longer, the model performance decreases significantly. In particular, on the EAE task, the results for long sentences decreased by 15.11% compared to short sentences. It is worthwhile to explore how to accurately identify different events from long and complex sentences and capture the correlation between event triggers and arguments across long distances. Reasonably truncating long sentences into short ones might be feasible, and introducing some extra attention mechanisms or graph-based information might improve the ability of the model to capture the long sentence dependencies.

To address these errors, we have conducted further experiments with the Large Language Model (LLM) (Ma et al., 2023), as shown in Appendix F. We observe that LLM suffers from extraction hallucinations on ZSEE, mainly in confusing the argument roles of different events. Therefore, we still look forward to further research on ZSEE to effectively address the above challenges, both on large and small language models.

5 Conclusion

In this paper, we present the **Z**eolite **S**ynthesis **E**vent **E**xtraction dataset, ZSEE, which contains nearly 5000 sentences collected from the literature. We design a comprehensive event schema including 16 event types and 13 event arguments, which cover all experimental steps of zeolite synthesis. Fine-grained event annotations for each sentence are further provided. We have performed extensive experiments on ZSEE and analyzed the strengths and weaknesses of current state-of-the-art methods. We also explore the limitations of the large language model on ZSEE, which highlights the necessity of ZSEE. Furthermore, we summarize the challenges and research directions in ZSEE, which can effectively drive the development of zeolite synthesis event extraction for automated synthesis.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Basic Science Center Program: 61988101), the Shanghai Committee of Science and Technology, China (Grant No.22DZ1101500), National Natural Science Foundation of China (62173145), the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017 and Shanghai AI Lab.

Limitation

We present the largest dataset of zeolite synthesis event extraction to our knowledge. Nevertheless, our dataset has several limitations. First, in terms of annotation quality, although all texts are annotated by two annotators and reviewed by an experienced reviewer, the reviewers mainly check for inconsistencies between two annotation results. The reviewers usually do not add new annotations that both annotators might have missed, resulting in some event information being missed. Second, although we have collected nearly 5000 sentences, ZSEE may still not meet the data requirements of the current deep learning methods. It is essential to provide more data with high-quality annotations. Third, the event schema that we design mainly covers the operational steps in the zeolite experimental procedure. But the information on the order of the experiments for each sentence is not annotated, which is also important for automated synthesis. One solution is to determine the order of events by developing rules (e.g., capturing words such as "after" and "before").

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015.

B. Burger, Phillip M. Maffettone, Vladimir V. Gusev, Catherine M. Aitchison, Yang Bai, Xiao yan Wang, Xiaobo Li, Ben M. Alston, Buyin Li, Rob Clowes, Nicola Rankin, Brandon Harris, Reiner Sebastian Sprick, and Andrew I. Cooper. 2020. A mobile robotic chemist. *Nature*, 583:237–241.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ace) program – tasks, data, and evaluation. In *International Conference on Language Resources and Evaluation*.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683.

Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Marusczyk, and Lukas Lange. 2020. The SOFC-exp corpus and neural approaches to information extraction in the materials science domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1255–1268. Association for Computational Linguistics.

Jiang Guo, A. Santiago Ibanez-Lopez, Hanyu Gao, Victor Quach, Connor W. Coley, Klavs F. Jensen, and Regina Barzilay. 2022. Automated chemical reaction extraction from scientific literature. *Journal of Chemical Information and Modeling*, 62(9):2035–2045.

Lezan Hawizy, David M. Jessop, Nico Adams, and Peter Murray-Rust. 2011. Chemicaltagger: A tool for

semantic text-mining in chemistry. *Journal of Cheminformatics*, 3:17–17.

Jiayuan He, Dat Quoc Nguyen, Saber Ahmad Akhondi, Christian Druckenbrodt, Camilo Thorne, Ralph Hoessel, Zubair Afzal, Zenan Zhai, Biaoyan Fang, Hiyori Yoshikawa, Ameer Albahem, Lawrence Cavedon, Trevor Cohn, Timothy Baldwin, and Karin M. Verspoor. 2020. Overview of chemu 2020: Named entity recognition and event extraction of chemical reactions from patents. In *Conference and Labs of the Evaluation Forum*.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908.

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023. AMPERE: AMR-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 10976–10993.

Edward Kim, Kevin Huang, Olga Kononova, Gerbrand Ceder, and Elsa Olivetti. 2019. Distilling a materials synthesis ontology. *Matter*, 1:8–12.

Edward Kim, Kevin Huang, Alexander C. Tomala, Sara Matthews, Emma Strubell, Adam Saunders, Andrew McCallum, and Elsa A. Olivetti. 2017. Machine-learned and codified synthesis parameters of oxide materials. *Scientific Data*, 4:170127.

Edward Kim, Zach Jensen, Alexander van Grootel, Kevin Huang, Matthew Staib, Sheshera Mysore, Haw-Shiuan Chang, Emma Strubell, Andrew McCallum, Stefanie Jegelka, and Elsa Olivetti. 2020. Inorganic materials synthesis planning with literature-trained neural networks. *Journal of Chemical Information and Modeling*, 60:1194–1201.

Olga Vitalievna Kononova, Haoyan Huo, Tanjin He, Ziqin Rong, Tiago Botari, Wenhao Sun, Vahe Tshitoyan, and Gerbrand Ceder. 2019. Text-mined dataset of inorganic materials synthesis recipes. *Scientific Data*, 6.

Martin Krallinger, Obdulia Rabal, Anália Lourenço, Julen Oyarzábal, and Alfonso Valencia. 2017. Information retrieval and text mining technologies for chemistry. *Chemical reviews*, 117:7673–7761.

Viet Dac Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. 2021. Event extraction from historical texts: A new dataset for black rebellions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2390–2400.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, and Philip S. Yu. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Yupian Lin, Tong Ruan, Ming Liang, Tingting Cai, Wen Du, and Yi Wang. 2022. DoTAT: A domain-oriented text annotation tool. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–8.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601.

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.

S. Hessam M. Mehr, Matthew Craven, Artem I. Leonov, Graham Keenan, and Leroy Cronin. 2020. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370(6512):101–108.

Manuel Moliner, Yuriy Román-Leshkov, and Avelino Corma. 2019. Machine learning applied to zeolite synthesis: The missing link for realizing high-throughput discovery. *Accounts of Chemical Research*, 52(10):2971–2980.

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey

Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64.

Gihan Panapitiya, Fred Parks, Jonathan Sepulveda, and Emily Saldanha. 2021. Extracting material property measurement data from scientific articles. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5393–5402. Association for Computational Linguistics.

Paul Raccuglia, Katherine C. Elbert, Philip Adler, Casey Falk, Malia B. Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A. Friedler, Joshua Schrier, and Alexander J. Norquist. 2016. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533:73–76.

Simon Rohrbach, Mindaugas Šiaučiulis, Greig Chisholm, Petrisor-Alin Pirvan, Michael Saleeb, S. Hessam M. Mehr, Ekaterina Trushina, Artem I. Leonov, Graham Keenan, Aamir Khan, Alexander Hammer, and Leroy Cronin. 2022. Digitization and validation of a chemical synthesis literature database in the chempu. *Science*, 377:172–180.

Yu Song, Santiago Miret, and Bang Liu. 2023. MatSci-NLP: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3621–3639, Toronto, Canada. Association for Computational Linguistics.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5571–5587.

Matthew C. Swain and Jacqueline M. Cole. 2016. Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904.

Alain C. Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H. Nair, Philippe Schwaller, and Teodoro Laino. 2019. Automated extraction of chemical synthesis actions from experimental procedures. *Nature Communications*, 11.

Xuan Wang, Vivian Hu, Xiangchen Song, Shweta Garg, Jinfeng Xiao, and Jiawei Han. 2021. ChemNER: Fine-grained chemistry named entity recognition with ontology-guided distant supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5227–5240.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Benfeng Xu, Quan Wang, Yajuan Lyu, Dai Dai, Yongdong Zhang, and Zhendong Mao. 2023. S2ynRE: Two-stage self-training with synthetic data for low-resource relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8186–8207. Association for Computational Linguistics.

Xianjun Yang, Ya Zhuo, Julia Zuo, Xinlu Zhang, Stephen Wilson, and Linda Petzold. 2022. PcMSP: A dataset for scientific action graphs extraction from polycrystalline materials synthesis procedure text. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6033–6046. Association for Computational Linguistics.

Hanzhang Zhou, Junlang Qian, Zijian Feng, Hui Lu, Zixiao Zhu, and Kezhi Mao. 2023. Heuristics-driven link-of-analogy prompting: Enhancing large language models for document-level event argument extraction. *ArXiv*, abs/2311.06555.

Appendix

A Event Schema

The definitions of all 16 action events are as follows.

Add indicates that some materials are added to the container at a specific temperature, with arguments specifying *material*, *temperature* and *container*.

Stir means that the mixture is stirred with full contact for a while, whose arguments includes *duration*, *temperature*, *revolution* and the *sample*.

Age means waiting a period of time for the reaction, with arguments specifying *duration*, *temperature*, *revolution* and the *pressure*.

Wash describes that the product is washed several times with some solvent, with arguments specifying *solvent*, *times*, and the *sample*.

Dry indicates that the product is dried in the container for a while at a specific temperature. The corresponding arguments contain *duration*, *temperature*, *container* and the specific *condition*.

Calcine indicates that the product is calcined at high temperature. The corresponding arguments contain *duration*, *temperature*, *container*, *sample* and the specific *condition*.

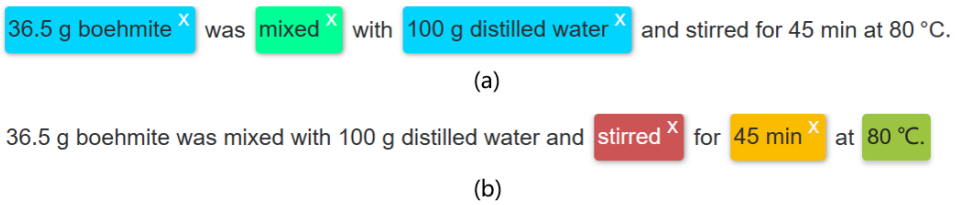


Figure 4: Annotation example with DOTAT (Lin et al., 2022).

Particle Recovery indicates that experimental operations such as filtration are carried out to recover the clean product. The corresponding arguments contain *material*, *duration* and *revolution*.

Set PH means that the product is brought to a specific pH value by adding material, with arguments specifying *material* and *PH*.

Cool means that the temperature of the product is reduced to a specific value, with arguments specifying *duration*, *temperature*, *container*, *sample* and the specific *condition*.

Heat means that the temperature of the product is increased to a specific value, with arguments specifying *duration*, *temperature*, *container*, *sample*, *pressure*, *revolution* and heating *rate*.

Crystallize is the key experimental step in zeolite synthesis, where the amorphous compound is converted to a crystalline state, with arguments specifying *duration*, *temperature*, *container*, *pressure* and *revolution* rate (or *revolution text*).

Transfer means that the product is transferred from one container to another, with arguments specifying *sample* and *container*.

Seal indicates that the product is kept in a sealed container, with arguments specifying *sample* and *container*.

Sonicate means that the product is washed by ultrasound, with arguments specifying *sample* and *solvent*.

React refers to ordinary reactions not specifically described in zeolite synthesis corpus, such as the reaction of materials at a specific temperature. The corresponding arguments contain *duration*, *temperature*, *material* and the specific *condition*.

Rotate refers to the direct rotation of a container, with arguments specifying *duration*, *temperature*, *container*, and *revolution*.

We list the definitions of all 13 event arguments to clarify the important role they play in the synthesis steps.

duration, ***temperature*** and ***pressure*** indicate the duration, temperature and pressure of the experiment respectively.

materials are compounds, both liquid and solid, which are added during experimental operations.

container indicates the container where the synthesis action is carried out.

sample is the subject of the reaction, which is different from the *material*.

solvent indicates the solvent to which the washing product is added.

times refers to the number of washings.

condition indicates the specific conditions under which the reaction is operated, e.g., performing the synthesis step in air.

revolution indicates the reaction is carried out at a specific revolution per minute, which is a common property of zeolite synthesis action.

revolution text indicates an abstract textual representation of the rotation, which indicates the presence or absence of the attribute rotation, while *revolution* refers to a specific value.

rate indicates that the temperature increases at a certain rate to a specific value.

PH indicates the specific pH value of the product.

B Annotation Guide

We provide an example to illustrate our annotation process in detail. Figure 4 shows an example of the annotation of the sentence "36.5 g boehmite was mixed with 100 g distilled water and stirred for 45 min at 80 °C." through DoTAT (Lin et al., 2022). The first synthesis event **Add** with the trigger "mixed" is annotated and then corresponding arguments are determined including *material*, as shown in Figure 4 (a). We then annotate the next synthesis event **Stir** with the trigger "stirred" and arguments *temperature* and *duration*, as shown in Figure 4 (b).

During the annotation, to bridge the potential expertise gap between chemistry and computer science annotators, we design a multi-round error correction. In the beginning, all the annotators are

Dataset	Split	Sents	Events	Event Types	Args	Arg Roles
ACE2005	Train	17172	4202	33	4859	22
	Dev	923	450	21	605	22
	Test	832	403	31	576	20
ERE-EN	Train	14736	6208	38	8924	21
	Dev	1209	525	34	730	21
	Test	1163	551	33	822	21
BRAD	Train	3847	2720	12	6057	6
	Dev	925	606	12	1219	6
	Test	866	933	12	2570	6
PHEE	Train	2898	3006	2	7230	3
	Dev	961	1003	2	2428	3
	Test	968	1010	2	2377	3
ZSEE (Ours)	Train	3931	**6966**	16	**11353**	13
	Dev	504	935	16	1475	13
	Test	526	972	16	1503	13
	Total	4961	8873	16	14331	13

Table 6: Key statistics of ZSEE and other EE datasets. Sents, Events, Event Types, Args and Arg Roles denotes the number of sentences, events, event types, arguments and argument role types, respectively. Note that the PHEE designs a hierarchical event schema and we report the number of main arguments here.

trained to ensure that they hold a common understanding of the event schema and annotation guild. The annotators would record the problem sentences. After a fixed number of annotations, all annotators would meet to discuss and analyze these problems. Annotators from computer science would propose solutions from an NLP perspective, while chemistry annotators might provide solutions based on their synthesis experimental experience. Based on the above discussion, the unified knowledge of all annotators are gradually refined and the accuracy of the annotation is also guaranteed.

C Event statistics

Table 6 shows the details and differences between ZSEE and other common or domain-specific datasets. Although ACE2005 provides a larger number of sentences than ZSEE, with 17,172 sentences in the training set, there are only 3,136 sentences containing events. We provide the largest number of annotated events and event arguments compared to these datasets. Besides, each sentence in ZSEE contains an average of three event arguments. Thus, performing event argument extraction on ZSEE, where more argument dependencies need to be considered, is more challenging than on other datasets.

In our proposed dataset ZSEE, different event types occur with different frequencies. We count the number of each event type in the training, validation and test sets, as shown in Table 7. Note that the division of training, test and validation sets in section 3.3 is mainly based on the consistency of event types. Besides, Table 8 shows the statistics of event argument roles. Table 9 presents the distribution of short, medium and long sentences.

D Experiment Details

The training details and hyperparameter settings for the experiments that we implement are as follows.

Classification-based method: We fine-tune the EEQA (Du and Cardie, 2020) to conduct experiments on ZSEE. We explore the performance of BERT-base and BERT-large (Devlin et al., 2019) through the Huggingface package (Wolf et al., 2020). The batch size and learning rate of both models for event detection are 32 and 4×10^{-5}, respectively. When training the event argument extraction models, we set the batch size to 16. The maximum training epoch is 30 for each experiment. Note that experiments are conducted on NVIDIA GeForce RTX 3090 GPUs.

Generation-based method: We follow DEGREE (Hsu et al., 2022) to achieve end-to-end

Event Type	Train	Dev	Test	Total
Add	1959	230	242	2431
Stir	1005	112	123	1240
Wash	591	95	105	791
Dry	570	96	95	761
Particle Recovery	558	96	88	742
Heat	517	65	77	659
Transfer	399	52	57	508
Calcine	368	53	51	472
Crystallize	324	48	42	414
Cool	193	33	28	254
Age	183	21	27	231
React	113	11	17	141
Seal	76	12	8	96
Set PH	59	6	7	72
Rotate	39	3	3	45
Sonicate	12	2	2	16

Table 7: Statistics of event types on ZSEE.

Argument Role	Train	Dev	Test	Total
material	3802	455	480	4737
temperature	2258	301	306	2865
duration	2110	279	283	2672
container	1019	121	121	1261
sample	949	132	128	1209
solvent	568	90	98	756
condition	248	40	40	328
revolution	147	17	14	178
times	77	15	10	102
PH	63	6	7	76
rate	45	8	11	64
pressure	51	8	3	62
revolution_text	16	3	2	21

Table 8: Statistics of event arguments on ZSEE.

generative event extraction of zeolite synthesis. We fine-tune BART (Lewis et al., 2020) with different sizes, i.e., BART-base and BART-large. We set the batch size to 32. The model is trained for 40 epochs with a learning rate of 1×10^{-5}. We set the number of negative examples for each sample to 15.

EAE method: To evaluate the performance of the SOTA EAE methods on ZSEE, we fine-tune the code of PAIE (Ma et al., 2022), where we also train two models based on BART-base and BART-large. We set the batch size to 16. The maximum training epoch and learning rate are 40 and 2×10^{-5}, respectively. Besides, AMPERE (Hsu et al., 2023) keep the same settings as DEGREE. We use the SOTA AMR-to-text model AMRBART (Bai et al., 2022) to introduce the AMR information[1].

Sentence Type	Train	Dev	Test	Total
Short sentences	613	85	85	783
Medium sentences	2862	359	381	3602
Long sentences	456	60	60	576

Table 9: Statistics of sentences with different lengths on ZSEE.

E Detailed Experiment Results

We design different question templates for classification-based method and generative templates for generation-based method to explore the best settings for current methods.

Classification-based method: We design six question templates for event detection, as shown in Table 10. Table 11 presents five question templates for event argument extraction. The descriptions of synthesis process added to the questions help the model to achieve better results.

Generation-based method: Table 12 show the templates we design for each event type in DEGREE and AMPERE. We express the arguments in a more natural way and aggregate them in a single sentence. For instance, we use "several times" to represent the argument *times*. Note that in the inference stage, the model enumerates all 16 templates of each event type and generates the templates filled with the target event content present in the source sentence. The structured event triggers and arguments can be parsed directly through comparing the output with the initial template.

EAE method: Inspired by PAIE (Ma et al., 2022), we explore the performance of three types of templates. The concatenation template just concatenates all argument roles of the given event type, while the soft template connects different roles with learnable, role-specific pseudo tokens. The manual template designs natural language to connect all argument roles for specific types. We show an example of the above three templates in Table 13. The results of different templates are shown in Table 14. The manual template with the base model outperforms other template settings, which is consistent with the research intuition that well-designed templates are more semantically coherent and provide

[1]https://github.com/goodbai-nlp/AMRBART

Questions	PLM	Tri-I	Tri-C
What is the trigger in the event?	BERT-b	93.69	92.14
	BERT-l	94.00	92.36
What happened in the event?	BERT-b	93.60	92.28
	BERT-l	93.81	**92.58**
What happened in the zeolite synthesis event?	BERT-b	<u>93.75</u>	<u>92.43</u>
	BERT-l	93.48	92.46
action	BERT-b	93.34	91.71
	BERT-l	93.67	92.23
synthesis action	BERT-b	92.85	91.53
	BERT-l	**94.11**	92.38
null	BERT-b	93.14	91.61
	BERT-l	93.94	92.42

Table 10: Results of different questions in ED task. The best result of large model is highlighted in bold and the best result of base model is underlined. Note that b denotes the base model and l denotes the large model in column PLM.

Questions	PLM	Arg-I	Arg-C
<argument role>	BERT-b	55.72	55.47
	BERT-l	56.36	55.76
<argument role> in <trigger>	BERT-b	67.20	<u>66.82</u>
	BERT-l	66.92	66.49
What is the <argument role>?	BERT-b	56.42	56.06
	BERT-l	56.38	56.07
What is the <argument role> in <trigger>?	BERT-b	<u>67.27</u>	66.76
	BERT-l	**68.13**	**67.86**

Table 11: Results of different questions in EAE task. The best result of large model is highlighted in bold and the best result of base model is underlined. Note that b denotes the base model and l denotes the large model in column PLM.

more information about zeolite synthesis. The results of the large model show that the soft template achieves best performance with the introduction of role-specific pseudo tokens, which can significantly reduce the effort to design the template.

F Results of LLMs

Currently, large language models exhibit strong few-shot learning and even zero-shot learning capabilities for information extraction (Agrawal et al., 2022). Following the prompt format in Ma et al. (2023), we explore the performance of LLMs in ZSEE. Specifically, we investigate the performance of LLMs on difficult samples that are difficult to handle by current SOTA methods. We leverage the ChatGPT (gpt-3.5-turbo-0301)[2] by giving the task definition and five demonstrations for each event type. The results on difficult samples are shown in Table 15. We observe that when performing the EAE task, the large language model can extract several event arguments that are difficult to identify by PAIE (the best EAE model on ZSEE). Inevitably, however, there are hallucinations in the results of LLM, where arguments that are irrelevant to the task definition are extracted. The LLM will confuse argument roles between different events, such as predicting the *solvent* of **Wash** in the **Add** event in Sample 1. The results of LLM are also not stable enough, where arguments of other events in the same sentence are predicted, as shown in Sample 2 and 3.

Thus, it remains difficult to obtain available structured information on zeolite synthesis for subsequent automated synthesis through LLMs directly. We assume that LLMs fine-tuned with knowledge from the chemical synthesis domain can

[2]https://openai.com/blog/openai-api

Event Type	EAE Template
Add	something was added to container at temperature.
Stir	something was stirred at temperature at revolution per minute for some time.
Age	wait for some time at temperature at revolution per minute under pressure.
Wash	wash something with something by several times.
Dry	something was dried in container at temperature for some time under condition.
Calcine	something was calcined in container at temperature for some time under condition.
Particle Recovery	something was recovered for some time at revolution per minute by adding something.
Set PH	something was set to PH by adding something.
Cool	something was cooled in container at temperature for some time under condition.
Heat	something was heated in container at heating rate to temperature for some time at revolution per minute under pressure.
Crystallize	crystallize in container at temperature for some time at revolution per minute under pressure under condition.
Transfer	something was transferred to container.
Seal	something was sealed in container.
Sonicate	something was sonicated with something.
React	something was treated at temperature for some time under condition.
Rotate	something was rotated in container at temperature for some time at revolution per minute.

Table 12: All EAE templates we designed for ZSEE when training the DEGREE (Hsu et al., 2022) and AMPERE (Hsu et al., 2023).

achieve better performance, but the fine-tuning process is costly. Notably, Ma et al. (2023) and Zhou et al. (2023) have discussed that LLMs would reach a performance stagnation with increasing sample size, whereas the performance of small language models can often be enhanced. In summary, we encourage further research on ZSEE to address the existing challenges to achieve more accurate extraction of experimental information for modular automated synthesis.

Types	Template Examples
CT	sample duration temperature container condition
ST	<sample_left_0> sample <sample_right_0> <duration_left_0> duration <duration_right_0> <temperature_left_0> temperature <temperature_right_0> <container_left_0> container <container_right_0> <condition_left_0> condition <condition_right_0>
MT	Calcine sample in container at temperature for duration under condition

Table 13: Different templates for ZSEE when training the PAIE (Ma et al., 2022). CT means the concatenation template, ST indicates the soft template, and MT indicates the manual template. We present the templates of event type **Calcine**.

Templates	PLM	Arg-I	Arg-C
Concatenation Template	BART-b	73.79	73.28
	BART-l	73.90	73.39
Soft Template	BART-b	74.30	73.80
	BART-l	**75.03**	**74.39**
Manual Template	BART-b	<u>74.52</u>	<u>74.01</u>
	BART-l	74.58	74.17

Table 14: Results of different templates of PAIE (Ma et al., 2022). The best result of large model is highlighted in bold and the best result of base model is underlined. Note that b denotes the base model and l denotes the large model in column PLM.

Sanple	PAIE	LLM
Sample1: Sodium aluminate (2.16 g) was dissolved in 10 ml of distilled water in a plastic beaker .	(material, Sodium aluminate (2.16 g)) (material, None) (container, plastic beaker)	(material, Sodium aluminate (2.16 g)) (material, 10 ml of distilled water) (container, plastic beaker) (solvent, distilled water)
Sample2: 8.51 g (2 mmol) of R2+(OH-)2 solution (0.235 mmol g-1) was mixed with 0.1 g (0.84 mmol) of a 32 wt% aqueous solution of NaOH and stirred for 15 min followed by the addition of 1.5 g colloidal silica (Snowtex1 40, Nissan Chemical Co.), which consists of 0.6 g (10.0 mmol) of SiO2, and the mixture was stirred for 30 min.	(material, 0.1 g (0.84 mmol)) (material, None)	(material, 0.1 g (0.84 mmol)) (material, 8.51 g (2 mmol) of R2+(OH-)2 solution (0.235 mmol g-1)) (duration, 15 min) (duration, 30 min)
Sample3: The relative molar composition of the starting mixture was Ga2O3: P2O5: HF: 70 H2O: 1.7 amine, obtained by successive addition with vigorous stirring of 0.61 g of orthophosphoric acid (85 wt% in water, Fisher) , 3.18 g of deionized water , 0.5 g of Ga2O3 , 0.133 g of hydrofluoric acid (40 wt% in water, Fluka) , and finally 0.36 g of 1-methylimidazole (Fisher, 99 wt% in water) or the equivalent amount of pyridine (0.35 g) was added with continuous stirring.	(material, 3.18 g of deionized water) (material, None) (material, None) (material, None)	(material, 0.61 g of orthophosphoric acid (85 wt% in water, Fisher)) (material, 3.18 g of deionized water) (material, 0.5 g of Ga2O3) (material, 0.133 g of hydrofluoric acid (40 wt% in water, Fluka)) (material, 0.36 g of 1-methylimidazole (Fisher, 99 wt% in water)) (material, equivalent amount of pyridine (0.35 g)) (action, continuous stirring) (container, Not explicitly mentioned)

Table 15: EAE results of LLMs on difficult samples. The triggers of the given event type are marked in orange font and the ground-truth arguments are highlighted in light blue. The purple font indicates the extraction errors of PAIE and LLM.