Overcoming Annotation Scarcity for Shallow Semantic Parsing in Scientific Procedural Text

Anonymous authors

Abstract

Materials science literature contains millions of synthesis routes described in unstructured natural language text. The large scale mining of these synthesis procedures promises to allow a deeper scientific understanding of materials synthesis and the automated planning of synthesis procedures. This however requires the construction of knowledge bases of synthesis procedures from natural language text. A major bottleneck in extraction of these structured synthesis representations from text is the lack of labeled data on which to train or evaluate extraction models. To address this bottleneck, we introduce a dataset of 230 synthesis procedures annotated with the labeled graph structures which express the semantics of the synthesis sentences. The nodes are operations and arguments in the synthesis, while labeled edges specify relations between the nodes. Next, we describe a novel weakly supervised approach to the extraction of unlabeled graph structures from synthesis sentences. The proposed model is framed as a matrix completion model parameterized by a DeepSet neural network [Zaheer et al., 2017]. The proposed model outperforms a strong heuristic baseline by 4 points precision and 2 points F1.

1. Introduction

Systematically reducing the time and effort required to synthesize novel materials remains one of the grand challenges in materials science. Access to massive knowledge bases which tabulate known chemical reactions for organic chemistry [Lawson et al., 2014] has accelerated data-driven synthesis planning and related analyses [Segler et al., 2018, Coley et al., 2017]. Automated synthesis planning for organic molecules has recently achieved human-level planning performance using massive organic reaction knowledge bases as training data [Segler et al., 2018]. There are, however, currently no comprehensive knowledge bases which document the methods by which inorganic materials are synthesized [Kim et al., 2017a,b]. Despite efforts to standardize the reporting of chemical and materials science data [Murray-Rust and Rzepa, 1999], inorganic materials synthesis routes continue to reside in non-standard form as natural language descriptions of synthesis. This data therefore requires
novel techniques to extract the described synthesis in order to study inorganic material synthesis [Kim et al., 2017b]. Fig 1 presents an example synthesis procedure.

A method for extracting action graphs from synthesis procedure text has been presented in Mysore et al. [2017]. One of the main bottlenecks in the extraction of action graphs for complete procedures is the extraction of semantic frame structures for each sentence. To improve the status of this bottleneck we make two contributions in this paper. First we release a dataset of 230 synthesis procedures annotated by materials scientists with the desired semantic frame structures (represented as a graph) to allow development and evaluation of extraction models. Next we describe an approach to the extraction of semantic frames from sentences without access to labeled training data.

Semantic frames express the semantics of text while being invariant to the manner in which the sentence is realized. The extraction of these structures has been an extensively studied problem in the computational semantics and information extraction communities, in the areas of shallow semantic parsing and event extraction [Gildea and Jurafsky, 2002, Das et al., 2014, Nguyen et al., 2016]. Much work has gone into creating annotated data for SRL [Palmer et al., 2005, Fillmore and Baker, 2010] which is used to train shallow semantic parsers. Given the expense of annotating data and the lengthy training for annotators, extraction of shallow semantic structures from un-annotated data is a desirable alternative. This paper presents a step in this direction.

We present a shallow semantic parsing dataset consisting of 230 synthesis procedures annotated with graph representations of the steps in the synthesis. The nodes in the graphs include operations or event triggers (henceforth, used interchangeably), materials, conditions, apparatus and other entities mentioned in the synthesis descriptions. Labeled edges represent relationships between all of the mentioned entities. An example annotation is given in Fig. 2. These annotations are made available to the community.1

In developing a full model for the extraction of semantic structures in the absence of labeled data, we break the problem of extracting a labeled graph into two simpler problems. The first sub-problem consists of making associations of arguments with an operation. We refer to this problem as unlabeled edge placement. In Fig. 2 this task would involve extraction of unlabeled graph structures of the form, placed(Cu, quartz tube furnace) and heated(degC, H2, mTorr). The second sub-problem consists of a clustering problem over operation-argument pairs into role clusters referred to as role induction. For example, the operation-argument pairs (placed, quartz tube furnace) and (crush, agate mortar and pestle) would be placed in a single cluster since the argument plays the role of instrument in both operations. This paper presents a solution for the first of these problems.

In the proposed method, the first step is to identify the operations and argument entities mentioned in a sentence. Prior work has found that it is often easier to label entities (manually or automatically) participating in a semantic frame than it is to label structured graph representations which describe relationships between the entities of the frame [Reschke et al., 2014, Ratner et al., 2017, Rooshenas et al., 2018].

Accordingly, our method relies on a supervised approach for the identification of entity spans from text and presents an approach for placement of unlabeled edges between operations and arguments which does not rely on manually labeled data. The proposed model consists

1. Public dataset: The dataset will be released upon acceptance.
In a typical procedure for the synthesis of $\beta$-MnO$_2$ nanowires, $25$ mL of $50$ wt% Mn(NO$_3$)$_2$ solution was diluted to $25.0$ mL, and ozone was fed into the bottom of the solution for $30$ min under vigorous stirring. With the indraught of ozone, black solid appeared gradually and the clear solution turned into black slurry finally. Then the suspension was transferred into an autoclave of $48.0$ ml, sealed and maintained at $200^\circ$C for $8$ h. After this, the autoclave was cooled to room temperature naturally. The resulting solid products were washed with water, and dried at $120^\circ$C for $8$ h. The obtained products were collected for the following characterization.

Figure 1: An example synthesis procedure adapted from Dong et al. [2009]. Colors and underlines were manually added here for clarity; text in bold red indicates the operations involved in the synthesis, bold black indicates arguments and underlines demarcate entity boundaries.

of a DeepSet [Zaheer et al., 2017] based affinity model which assigns a joint score to a whole candidate argument set for attachment to a given operation. The affinity model is trained to rank observed operation-argument sets over unobserved operation-argument sets and is cast as matrix completion model. Positive training operation-argument sets are generated by a heuristic applied on sentences. The heuristic makes reading order assignments of arguments to operations. We also consider this heuristic our baseline model. The proposed model improves upon this baseline by 4 points in precision and 2 points of F1.

The remainder of this paper is organized as follows. We give a description of the dataset (§2), briefly describe our entity extraction model (§3), describe our model for unlabeled edge placement (§4), and present evaluation (§5). Then we presented related work (§6) and conclude with future work (§7).

2. Description of the Annotated Dataset

To address the lack of datasets for the development and evaluation of extraction methods for materials synthesis text we created a dataset consisting of 230 synthesis procedures annotated with the structured graph representations of the individual steps in the synthesis. All annotations were performed by three materials scientists using the BRAT annotation tool.\footnote{BRAT Annotation tool: \url{http://brat.nlplab.org/}}

**Selection of Annotated Synthesis Procedures:** The 230 synthesis procedures annotated were randomly selected from our database of 2.5 million publications describing materials synthesis. The database of 2.5 million was built from agreements with major scientific publication companies. Synthesis paragraphs were obtained by parsing HTML text of publications and the application of a paragraph classifier for identification of synthesis paragraphs. This classifier was trained on a set of manually labeled paragraph examples.
Figure 2: The complete graph structures we would like to extract. Semantic frames generally consist of operations and arguments of these operations as nodes and labeled edges between operation and argument nodes, for e.g. **HEATED**(Condition of: degC, Atmospheric Material: H2, Condition of: mTorr). Our dataset also includes additional structure, labeled edges between argument entities and non-operation entities, for e.g. **Descriptor of** (Cu, foils) and relations between operations, for e.g. **Next Operation**(placed, heated). This work does not deal with extraction of these additional structures.

and had a F1 score of 90.2 on the test set.\(^3\) The paragraphs selected by the classifier were manually verified as containing complete valid synthesis procedures by materials scientists.

**Structures Annotated:** The recipe graph consists of nodes denoting the participants of synthesis steps and edges denoting relationships between the participants of the synthesis. Operation nodes define the main structure of the graph with the different materials, conditions and apparatus serving as the arguments for each operation. In annotating text describing synthesis, we define a set of span-level labels which identify the operations and the different kinds of arguments. We refer to operations and arguments spans together as “entity mentions”. Spans may be viewed as a sequence of tokens or characters which form one entity mention (e.g. ”quartz tube furnace”). Entity mentions are associated with “entity types” which specify a category/kind for the entity mention. The 10 most frequent entity types defined for our dataset are listed in Table 1a. We also measured inter-annotator agreement on the span-labels on set of 5 documents labeled by all three annotators. The annotators had a Cohen Kappa Score [Artstein and Poesio, 2008] of 0.819 averaged across pairs of annotators.

Next, we define a set of relationships between entity mentions, which label the edges of the synthesis graph. A subset of these relations describe direct relationships between operations and their arguments, others describe relationships between non-operation entity mentions, and the Next-Operation relation describes relationships between operations so as to step towards annotating full recipe graphs. In the first release of the data, as a placeholder for future annotations, Next-Operation is used simply used to indicate the next operation in text order rather than in true synthesis order.

3. Entity extraction

For automatic graph extraction, we extract entity mentions from the text by casting it as a supervised sequence labeling task trained on the labeled dataset described in §2. Specifically,

\[^{3}\text{Readers are referred to the released dataset for documentation about the paragraph classifier.}\]
<table>
<thead>
<tr>
<th>Entity type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>4843</td>
</tr>
<tr>
<td>Number</td>
<td>4095</td>
</tr>
<tr>
<td>Operation</td>
<td>3786</td>
</tr>
<tr>
<td>Amount-Unit</td>
<td>1659</td>
</tr>
<tr>
<td>Condition-Unit</td>
<td>1621</td>
</tr>
<tr>
<td>Material-Descriptor</td>
<td>1430</td>
</tr>
<tr>
<td>Condition-Misc</td>
<td>535</td>
</tr>
<tr>
<td>Synthesis-Apparatus</td>
<td>490</td>
</tr>
<tr>
<td>Nonrecipe-Material</td>
<td>475</td>
</tr>
<tr>
<td>Brand</td>
<td>348</td>
</tr>
</tbody>
</table>

Table 1: Entity types and relation labels annotated in our dataset. The table (a) depicts the 10 most frequent of the 22 entity types defined in our dataset, and the table (b) includes 14 relation labels among entities. We refer readers to the released dataset for a full entity type statistic.

given a tokenized sentence \( x = [x_1, \ldots, x_L] \) we predict per-token output tags \( y = [y_1, \ldots, y_L] \). With \( y_i \in S \), the label set denoted in Table 1a. We predict BIO5 coded labels for tokens independently using contextualized token features from a dilated-CNN [Strubell et al., 2017]. Token embeddings are initialized with FastText embeddings [Bojanowski et al., 2016] pre-trained on our entire corpus.

4. Unlabeled Edge Placement

In the sections that follow, we describe a general framework for affinity models for edge placement (§4.1) and cast our model as a matrix completion model (§4.2). We next describe the creation of our weakly supervised training dataset (§4.3), followed by the parameterization of our model using DeepSets (§4.4), and finally a simple inference procedure which makes edge assignments given a trained affinity model (§4.5).

4.1 General Framework for Affinity Modeling

We would like to score a given operation \( r \) (eg. HEAT) and argument set \( t \) (eg. (Solution, 120°C, Autoclave)) with a probability \( p(y_{r,t} = 1) \), where \( y_{r,t} \) is a binary random variable that is True iff \( \langle r, t \rangle \) is a valid assignment of the operation with its arguments. Note that \( \langle r, t \rangle \) may be seen as being an n-ary relation tying together the trigger and all its arguments. We denote the set of all operations in the dataset with \( R \), the set of all possible sets of arguments in the training data with \( T \), and the set of positive operation-argument set pairs used for training, \( \{ \langle r, t \rangle \} \) with \( O \). The probability \( p(y_{r,t} = 1) \), is estimated by means of the logistic function parameterized by \( \theta_{r,t} \) as, \( p(y_{r,t} = 1) = \sigma(\theta_{r,t}) \). Here, \( \theta_{r,t} = K(r, t) \) attempts to measure the compatibility between the trigger \( r \) and the argument set \( t \) as a function of

---

4. Sentences segmentation and tokenization was performed with the ChemDataExtractor package: [https://pypi.org/project/ChemDataExtractor/1.2.2/](https://pypi.org/project/ChemDataExtractor/1.2.2/)
5. We append the the Beginning Inside Other (BIO) prefixes to the labels to indicate multi-token entities.
the representations of the event trigger $v_r$ and the argument set $a_t$. §4.4 presents different neural network parameterizations of the function $K$.

4.2 Training Objective

The proposed model can be considered to be a kernelized matrix completion model that attempts to score an event trigger and the argument set pair. The matrix consists of event triggers $R$, along the rows and argument sets $T$, along the columns. The dataset $O$ which the model is trained with, however, is a subset of all the positive trigger-argument set pairs. Therefore, the unobserved pairs consist of a mix of missing positive pairs (valid events) as well as negative pairs (invalid events). Furthermore, we also desire that rare/less frequent (though valid) events not be penalized by the model as being unlikely. In settings with sparse positive training data, an approach which has been shown to work in collaborative filtering [Rendle et al., 2009], learning word representations [Vilnis and McCallum, 2015, Collobert et al., 2011] and relation factorization [Drumond et al., 2012, Riedel et al., 2013] is one that optimizes a ranking loss. In the current problem setting therefore we choose to optimize the Bayesian Personalized Ranking (BPR) loss proposed in [Rendle et al., 2009].

The BPR loss assumes the observed pairs to be true and aims to rank them above unobserved pairs. This relaxed objective is formulated as:

$$L_{BPR} = - \sum_{(r,t) \in O} \sum_{(r,t) \notin O} \log(\sigma(\theta_{f^+} - \theta_{f^-}))$$ (1)

Here $f^+ = \langle r, t \rangle \in O$ represents the observed pair while $f^- = \langle r, t \rangle \notin O$ represents an un-observed pair. Unobserved pairs are generated by a random permutation of a given trigger with a different argument set in $T$. Note that $\theta_{r,t^+} > \theta_{r,t^-} \implies p(y_{r,t^+} = 1) > p(y_{r,t^-} = 1)$.

4.3 Creation of the Training Dataset

We create a weakly labeled training dataset from a corpus of about 500,000 synthesis procedures. Text extraction from journal articles, identification of synthesis procedure paragraphs and sentence level entity extraction was performed as described in §3. Following this, a heuristic, which we term the Greedy Reading Order (GRO) heuristic was applied on entity tagged sentence text to obtain noisely labeled examples of operation-argument set pairs on which the affinity model is trained. The heuristic associates arguments with operations in the reading order of the sentence. Arguments before and after the first and last operation associate to each in order and all arguments between two operations are split equally between operations. Single operation sentences are trivially extracted. We consider this our baseline model GRO-BASELINE. The described approach gives us a training dataset of 10-million training examples.

4.4 Argument Affinity Model Parameterization

The function aims to model $K(r, t)$ the score, or affinity, of the event trigger $r$ and the argument set $t$. This is formulated as a linear function of the representations of the event trigger $v_r$ and the argument set $a_t$:

$$\theta_{r,t} = K(r, t) = v_r^T a_t = v_r^T \text{encoder}_t(t)$$ (2)
The encoder returns compositional representations for argument sets based on the argument elements. We define three alternative variants for encoder. We call our first model DEEPSET-VS and choose encoder to be a deep-set defined as:

\[ a_t = f(\sum_{i=1}^{N} g(e_i)) \]  

(3)

We define another model, DEEPSET-TYPE-VS, by adding additional argument entity-type information and retain the deep-set formulation of encoder:

\[ a_t = f(\sum_{i=1}^{N} l(s_i, e_i)) \]  

(4)

Here, \( N \) represents the number of elements in the argument set, \( e_i \) represents an argument embedding, \( s_i \) represents an embedding for the entity type of the argument element. The functions \( f \) and \( g \) represent distinct feed-forward neural networks defined as:

\[ f(z) = g(z) = \tanh(Wz + b) \]  

(5)

The function \( l \) is feed-forward network with a bi-linear function and defined as:

\[ l(s_i, e_i) = \tanh(s_i^T W_l e_i + b_l) \]  

(6)

where \( W_f, b_f, W_g, b_g, W_l, b_l \) are shared parameters of the model and \( W_l \) is a 3D tensor.

The function \( l \) aims to return contextualized argument representations. This bilinear-operation allows the the type embedding, \( s_i \), to have operator like semantics on the argument representations \( e_i \). For example, the argument "K" might refer to the unit for temperature Kelvin with entity type Condition_Unit or it might refer to the element Potassium with entity type Material.

Finally we define a simpler non-deep model, SET-VS, defined as:

\[ a_t = \sum_{i=1}^{N} e_i \]  

(7)

In all three variants, we choose to model the argument sets as order invariant sets since early experiments indicated order-invariant representations to perform better than sequential ones which used an LSTM encoder. The structure of the networks in Eqs. 3, 4, 7 are inspired by recent work which aims to learn order invariant representations for sets [Zaheer et al., 2017]. We also normalize the affinity scores by the size of the argument set since experiments indicated that the set based scoring functions tended to score larger sets higher.

4.5 Exhaustive Inference

For placement of edges on a test sentence with the trained affinity model we must define an inference procedure based on which edge assignments may be made on a sentence. Since we present a model which considers entire argument sets for attachment with a trigger, for a
<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>86.04</td>
<td>70.11</td>
<td>77.26</td>
<td>560</td>
</tr>
<tr>
<td>Number</td>
<td>95.39</td>
<td>90.39</td>
<td>92.83</td>
<td>428</td>
</tr>
<tr>
<td>Operation</td>
<td>87.60</td>
<td>75.99</td>
<td>81.38</td>
<td>686</td>
</tr>
<tr>
<td>Amount-Unit</td>
<td>92.86</td>
<td>86.67</td>
<td>89.66</td>
<td>98</td>
</tr>
<tr>
<td>Material-Descriptor</td>
<td>52.81</td>
<td>45.63</td>
<td>48.96</td>
<td>89</td>
</tr>
<tr>
<td>Condition-Unit</td>
<td>93.10</td>
<td>82.65</td>
<td>87.57</td>
<td>87</td>
</tr>
<tr>
<td>Brand</td>
<td>51.85</td>
<td>53.85</td>
<td>52.83</td>
<td>68</td>
</tr>
<tr>
<td>Synthesis-Apparatus</td>
<td>55.88</td>
<td>44.19</td>
<td>49.35</td>
<td>34</td>
</tr>
<tr>
<td>Nonrecipe-Material</td>
<td>28.00</td>
<td>63.64</td>
<td>38.89</td>
<td>25</td>
</tr>
<tr>
<td>Condition-Misc</td>
<td>90.00</td>
<td>58.06</td>
<td>70.59</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2: The span-based performance (%) of entity extraction broken down by the entity types. We include entity types which had more than 20 annotated entity mentions in our test set. Predictions were made with a dilated-CNN neural network with token embeddings initialized with pretrained FastText embeddings.

sentence with M identified argument spans and multiple triggers we would need to evaluate $2^M$ candidate argument sets for attachment with every trigger in the worst case. In practice, due to the nature of the synthesis data and the assumptions of our current annotation⁷, the only arguments which have ambiguous attachment to a trigger are ones which occur between two triggers, such as "quartz tube furnace" between placed and heated in Fig. 2. This allows us to perform an exhaustive search for edge placements. For each trigger in the sentence, we list all argument candidates and select the argument set receiving the highest score for a trigger by the trained affinity model as being the final edge assignment. We observe that on average we score 4 candidates extractions per every trigger.

5. Evaluation

5.1 Entity Extraction

We perform a span-based evaluation of the entity extraction model described in §3. We observe microaveraged precision, recall and F1 of 71.85, 79.61, 75.54 and on our test set. Table 2 presents these results broken down by the entity types.

5.2 Unlabelled Edge Placement

We perform an evaluation of the edge placement model on a set of 200 test documents (consisting of about 3000 event instances). The trained affinity model is used to predict edges for every predicate in a sentence independently. Table 3 presents the results of this evaluation. All of the proposed models DeepSet-VS, DeepSet-Type-VS and Set-VS improve upon the heuristic GRO-Baseline on overall F1 with the DeepSet-VS improving upon the baseline precision by 4.7 points and the overall F1 by 2 points. Interestingly, though

⁷ We disallow argument re-use by triggers which prevents many long-distance trigger-argument edges. We also observe that crossing edges rarely occur in the gold data, hence we do not score candidates with non-contiguous arguments.
Table 3: Evaluation of the un-labeled edge placement models in terms of microaveraged Precision, Recall and F1 (%). For the unlabeled edge placement task, given the spans of arguments and the operation in a sentence edge placement is trivial in a single operation/event sentence. A majority of the events are expressed in multi-operation sentences, we present an evaluation of these multi-operation/event sentences separate from all the sentences. About 60% of our test sentences are multi-operation sentences. In our evaluation setting we also make use of gold argument and operation spans instead of predicted spans.

<table>
<thead>
<tr>
<th>Multi-operation Sentences</th>
<th>All Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
</tr>
<tr>
<td>GRO-Baseline</td>
<td>86.54</td>
</tr>
<tr>
<td>Set-VS</td>
<td>91.32</td>
</tr>
<tr>
<td>DeepSet-VS</td>
<td>91.28</td>
</tr>
<tr>
<td>DeepSet-Type-VS</td>
<td><strong>91.34</strong></td>
</tr>
</tbody>
</table>

Table 4: Nearest neighbours of operation embeddings learned in training the DeepSet-VS affinity model. Operations likely to describe the same event and likely to have a similar semantic frames seem to be grouped together.

<table>
<thead>
<tr>
<th>operation</th>
<th>nearest neighbours</th>
</tr>
</thead>
<tbody>
<tr>
<td>pyrolyzed</td>
<td>carbonize, heat - treated, fire, thermally treated, heat treated</td>
</tr>
<tr>
<td>vortexed</td>
<td>centrifuge, vortexing, shake, agitate, vortex - mixed, sonicated</td>
</tr>
<tr>
<td>cold pressed</td>
<td>press, uniaxially pressed, cold - pressed, dry - pressed, compact</td>
</tr>
<tr>
<td>etherification</td>
<td>esterification, acetalization, epoxidation, Reaction, Oxidation</td>
</tr>
<tr>
<td>purchase</td>
<td>buy, procure, receive, purchased from, supply, used</td>
</tr>
<tr>
<td>clean</td>
<td>degrease, pre-cleaned, pre-cleaned, etch, rinse, cleanse</td>
</tr>
<tr>
<td>award</td>
<td>thank, grateful, supervise, indebted, grant, acknowledge</td>
</tr>
</tbody>
</table>

DeepSet-Type-VS makes use of additional entity type information for the arguments, DeepSet-VS seems to perform better on F1. Additionally we also note that the DeepSet models only improve performance over a non-deep model only marginally. In the presented evaluation of the unlabeled edge placement model we make use of gold argument and operation spans in a sentence, and gold entity types. All the reported numbers are averages over 3 separate runs of the the models. We also expect operation embeddings learned by our models to form meaningful clusters. Table 4 lists some qualitative results depicting this.

6. Related Work

**Extraction of Semantic Structures from Text:** Prior work in the NLP community has defined and annotated semantic structures for non-scientific text as done in, PropBank [Palmer et al., 2005], FrameNet [Fillmore and Baker, 2010], AMR [Banarescu et al., 2013] and ACE event schemas [Doddington et al., 2004]. The GENIA project has defined event structures for biomedical data [Kim et al., 2003]. There has, recently, also been an interest in labeling scientific wetlab protocol text, with semantic structures and training supervised models for the extraction of these structures [Kulkarni et al., 2018]. These structured representations often seek to generalize about a predicate (often a verb) and its arguments'
semantic roles; abstracting away from the surface nuances of natural language and representing its semantics. We refer readers to work by Abend and Rappaport for a lucid coverage of semantic representations for text [Abend and Rappaport, 2017].

Prior approaches to the extraction of semantic frames from text in settings of limited resources have been along a handful of different directions; one line of work aims to extend the coverage of frame annotations to unlabeled triggers by label propagation and weak supervision [Fürstenau and Lapata, 2009, Das and Smith, 2011, Reschke et al., 2014]. Closely related to this is the line of work which learns to map one relatively resource rich frame structure to a different target frame structure which is poorer in annotations [Huang et al., 2018, Rao et al., 2017]. A large line of work, variously termed role/schema/frame induction, attempts to induce individual or groups of roles in a completely unsupervised manner and often employs latent variable models or representation learning models followed by a clustering step [Cheung et al., 2013, Titov and Khoddam, 2015, Woodsend and Lapata, 2015, Lang and Lapata, 2014, Huang et al., 2016]. There has been significantly lesser work on prediction of unlabeled graph structures in the unsupervised frame extraction literature beyond simple baselines. The closest work on the problem we present comes from Abend et al. [Abend et al., 2009], who propose a argument span and trigger-argument edge placement method for PropBank arguments using a set of heuristics applied on an unsupervised syntactic parse and mutual information considerations. Another line of work which appears close to the one we present, is the one on extraction of n-ary relations across sentence boundaries presented by Peng et al. [2017]. Peng et al. present an approach to extract ternary relations of drug-gene-mutation interaction triples from a group of sentences with a DAG-LSTM model trained on sentences distantly labelled from a knowledge base of interaction triples. Peng et al. also highlight the strong closeness of n-ary relation extraction to that of extracting semantic frames, an approach that we follow in this paper.

Materials Science & Chemistry: The focus of existing large-scale inorganic materials knowledge bases has primarily been on on materials structures and properties [Jain et al., 2013, Kirklin et al., 2015], rather than reactions and synthesis. Comprehensively extracting the knowledge contained within written inorganic materials syntheses, without the use of significant human effort, is a key step towards reducing the overall discovery and development time for novel materials [Butler et al., 2018].

Prior work by Raccuglia et al. [2016] and Ghadbeigi et al. [2015] have shown that manual extraction and subsequent text mining can be an effective approach to analysis of synthesis routes for specific materials. However, such approaches are fundamentally limited in scale, as manually extracting data from more than a few hundred journal articles is impractical. In pursuit of more scalable methods for materials synthesis data extraction, Young et al. [2018] have made use of automated methods for extracting specific categories of materials synthesis parameters, while Mysore et al. [2017] and Kim et al. [2017a] have both presented various methods for automated text extraction from materials science literature across a variety of material types and synthesis methods.

7. Conclusion and Future Directions

In this work we present a shallow semantic parsing dataset consisting of 230 synthesis procedures. This dataset presents progress for training robust supervised entity tagging
models and is suitable for the evaluation of models trained to extract shallow semantic structures, although not quite large enough for training supervised semantic parsers. We believe un/semi-supervised approaches to shallow semantic parsing provide a valuable route to extraction of semantic structures in resource limited settings. Our model for extraction of unlabeled semantic structures represents a promising step in this direction, which remains relatively un-explored in the NLP community. Future work will explore approaches to refining and scaling up our annotation effort, development of un/semi-supervised models for extraction of labeled semantic structures, efficient and generic inference procedures for models with global factors, and development of models for construction of full synthesis graphs. We believe we are at high enough extraction performance levels to meaningfully consider large scale recipe graph extraction for synthesis mining, and future work plans to explore this direction as well.

References


Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, 
Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning 
representation for sembanking. In Proceedings of the 7th Linguistic Annotation Workshop 
and Interoperability with Discourse, 2013.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie 
Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, 

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. Genia corpusa semantically 

Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. An annotated corpus for 
machine reading of instructions in wet lab protocols. NAACL, 2018.

Omri Abend and Ari Rappoport. The state of the art in semantic representation. In ACL, 
2017.

Hagen Fürstenau and Mirella Lapata. Semi-supervised semantic role labeling. In EACL, 
2009.

Dipanjan Das and Noah A Smith. Semi-supervised frame-semantic parsing for unknown 

Lifu Huang, Heng Ji, Kyunghyun Cho, and Clare R Voss. Zero-shot transfer learning for 
event extraction. ACL, 2018.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. Biomedical event extraction 

NAACL, 2013.

Ivan Titov and Ehsan Khoddam. Unsupervised induction of semantic roles within a 
reconstruction-error minimization framework. NAACL, 2015.

Kristian Woodsend and Mirella Lapata. Distributed representations for unsupervised 
semantic role labeling. In EMNLP, 2015.

Joel Lang and Mirella Lapata. Similarity-driven semantic role induction via graph partition-

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and 

Omri Abend, Roi Reichart, and Ari Rappoport. Unsupervised argument identification for 

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-
sentence n-ary relation extraction with graph lstms. TACL, 2017.


