Device Fabrication Knowledge Extraction from Materials Science Literature

Neelanshi Wadhwa, Sarath S, Sapan Shah, Sreedhar Reddy,
Pritwish Mitra, Deepak Jain, Beena Rai

TCS Innovation Labs, Tata Consultancy Services,
TRDDC, Pune, India 411013
{neelanshi.w, sarath.s8, sapan.hs, sreedhar.reddy, pritwish.mitra, deepak.jain3, beena.rai}@tcs.com

Abstract

Devices like solar cells, batteries etc. often comprise of a host of material types including organic, inorganic and hybrid materials. The fabrication procedures for these devices involve screening or designing the right set of materials and then subjecting them to a sequence of operations under very specific conditions. The performance characteristics of a device critically depend on the materials used in its fabrication, the specific operations carried out, their operating conditions and the specific sequence in which they are carried out. The space of potential materials, operations and operating conditions is vast, and selecting the right combination thereof to achieve the desired characteristics is a knowledge intensive activity. A large amount of such device fabrication knowledge is available in the form of publications, patents, company reports and so on. In this paper, we present a system that systematically extracts this knowledge from materials science literature. The extracted knowledge is represented as knowledge graphs conforming to an ontology that can be queried to make informed decisions in device fabrication procedures. The system first identifies the set of relevant paragraphs that contain fabrication knowledge. It then employs state of the art entity and relation extraction models to identify instances of operations, methods, materials, etc. and relations between them. The system then applies an unsupervised algorithm to identify sequences of operations representing fabrication procedures. We applied our system on solar cell fabrication knowledge extraction and achieved good performance. We believe our results provide much needed impetus for further work in this area.

1 Introduction

Technologies like solar cells, lithium-ion batteries, light emitting diodes etc. have benefited enormously as a result of fundamental research in materials science. Traditional approaches for screening and design of these materials include combination of tedious experimental and theoretical characterization. Lately, in-silico techniques like computational material science and machine learning based property prediction models are being leveraged to help guide the materials design process (Jain et al. 2019b,a). Although these techniques have shown great promise in reducing time and effort for materials design/screening, they still require excessive experimental validation in order to prove their correctness. Moreover, it is not often known a priori that a promising material screened via the above procedure when incorporated with other materials during the device fabrication process, shall result in expected device performance. Devices like solar cells, batteries etc. often comprise of more than one material, are subjected to various operations under very specific conditions, before finally being characterized for desired performance (Chen et al. 2019; Kaur et al. 2019). A large amount of such device fabrication knowledge is available in the form of publications, company reports, patents and so on (Jena, Kulkarni, and Miyasaka 2019). However, as this knowledge is present in textual form, it is usually fragmented and there is no consolidated structured view which can be queried or analysed. A knowledge graph for device fabrication along with suitable query and analysis tools can greatly reduce time and efforts by providing good starting points for fabrication design space exploration and guiding the design process.

In this work, we present a system for automatically extracting device fabrication knowledge from materials science literature. We first present a comprehensive ontology to represent the device fabrication knowledge. It provides constructs to represent the sequence of operations, their parameter set points, materials used, device transformation, and so on. The extracted knowledge is represented using knowledge graphs conforming to this ontology. The system first employs a relevance classifier to identify the set of paragraphs that contain device fabrication knowledge. It then employs state of the art models for entity and relation extraction to identify domain entities i.e. instances of domain concepts such as operation, material, method, etc. and relations between them. These algorithms extensively leverage prior domain knowledge in the form of pre-trained word embedding models and dictionaries. Finally, the system applies an unsupervised algorithm to identify the sequence of operations that make up a device fabrication procedure. The extracted entities, relations and procedures are then combined to create the knowledge graph. We used our system to extract solar cell fabrication knowledge from materials science literature and achieved good performance.

The major contributions of this work are: (1) A comprehensive ontology to represent device fabrication knowledge; (2) A system for automatically extracting knowledge graphs

conforming to the ontology from materials science literature using state of the art entity and relation extraction models; and (3) demonstration of our system on solar cell fabrication knowledge extraction. The rest of the paper is organized as follows. In the next section, we discuss the relevant related work. Section 3 discusses the device fabrication knowledge extraction problem in detail. We then present our knowledge graph extraction system in section 4 and discuss our experiments on solar cell fabrication in section 5. Section 6 then discusses how the extracted knowledge can be used by our framework for device fabrication. We finally conclude our paper and discuss future work directions in section 8.

2 Related Work

With recent advancements in in-silico methods for material design showing promising results (Gómez-Bombarelli et al. 2016), focus has now been shifted to computational synthesis planning. Organic chemistry extensively leverages chemical reactions databases and has been shown to achieve near human accuracies at retrosynthesis routes prediction (Segler, Preuss, and Waller 2017). On the other hand, inorganic material synthesis knowledge is primarily present as natural language text in scientific articles. Recently, Kim et al. have proposed a framework for extracting synthesis parameters of oxide materials from unstructured sources (Kim et al. 2017b,a). Efforts have also been initiated to explore methods for structured representation of synthesis routes of inorganic materials (Mysore et al. 2017; Kuniyoshi et al. 2020). All this infrastructure is eventually being used to develop a digital counterpart for planning inorganic materials synthesis (Kim et al. 2020), which would help accelerate the material discovery-design cycle. In this work, we attempt to address a much broader issue, that of device fabrication knowledge extraction, which entails additional complexity due to involvement of multiple materials and operations and their effect on device performance.

AI applications have long been using various knowledge representation formalisms such as semantic networks, frames, ontologies etc. for explicit domain modelling. Recently, knowledge graphs have emerged as a standard way to store domain entities and relations as they facilitate advanced analytics and query support. These knowledge graphs are either manually created or automatically populated using various information extraction techniques (Ji et al. 2020). Recurrent neural network based models such as BiLSTM, GRU, etc. in deep learning are extensively used for this purpose. Recently, transformer (Vaswani et al. 2017) based models have emerged as a strong alternative. In this work, we experiment with variants of these models.

3 Device Fabrication Knowledge Extraction

Device fabrication typically involves identifying the right set of materials and then subjecting them to various operations under very specific conditions. The final performance characteristics of the device depend not just on the parameter set points, methods, etc. of the individual operations but also on the sequence in which these operations are performed.

We present a comprehensive ontology for representing device fabrication knowledge. An ontology provides a formal mechanism to model the domain of interest in terms of the concepts occurring in the domain and relations between them (Gruber 2009). Figure 1 shows the concepts and relations present in the device fabrication ontology. The class *Device* represents the products that are being developed. A device has an associated *fabrication process* which captures the procedure involved in its fabrication in the form of a sequence of operations (using *next* relation). The class *Operation* represents an atomic process performed on a material and/or device with a particular set of parameters. A *parameter* is a measurable factor like time, temperature, speed etc., required for the operation. An operation may use one or more *materials* and apply a specific *method* using an *apparatus* to achieve the desired state of the device and/or material.

The class *Material* represents materials used in an operation. A material may be a chemical compound, a chemical element, a solution or a mixture. Different materials may play different roles depending on how they are used in an operation. A material that is used as an input to the operation is referred as an *inMaterial* (e.g. reactants in a synthesis operation). Similarly, a material which is the outcome of an operation is referred as an *outMaterial* (e.g products of synthesis operation). Whereas, a material that is necessary to carry out the operation in an efficient manner but not transformed chemically is referred as a *secondary* material (e.g. materials such as catalyst and promoter).

The class *Characterization* describes the performance of the device, which is a measure of the property the device exhibits when subjected to specified conditions. The conditions here refer to the set of operating parameters used during characterization.

Our goal is to mine knowledge graphs corresponding to this ontology. A knowledge graph consists of domain entities and relations between them. Ontology specifies the types of entities and relations we are interested in. We apply state of the art entity and relation extraction models for knowledge graph extraction. In this work, we demonstrate our system on solar cell fabrication knowledge extraction.

3.1 Perovskite Solar Cells

Perovskite solar cells (PSC) have recently emerged as one of the most promising photovoltaic technology (Park 2020). A typical PSC fabrication involves sequential deposition of 4 to 5 materials on the conductive substrate, where each material serves a specific purpose. In the normal architecture, the conductive substrate is coated with an electron transport layer (ETL) followed by the light absorbing material (perovskite in this case), a hole transport layer (HTL) and eventually an electrode (Kim et al. 2012). The amount of material options available at each of these layers are enormous (barring maybe the electrodes), thus giving rise to several solar cells, with varied performances (Fu et al. 2018). Even with the same set of materials, there could be several fabrication routes varying in operation specific details again giving rise to solar cells with diverse characteristics. Thus, it is often a daunting task for a human to process such scattered knowledge to arrive at crucial decisions while fabricating such devices. Figure 2 shows an example text fragment describ-

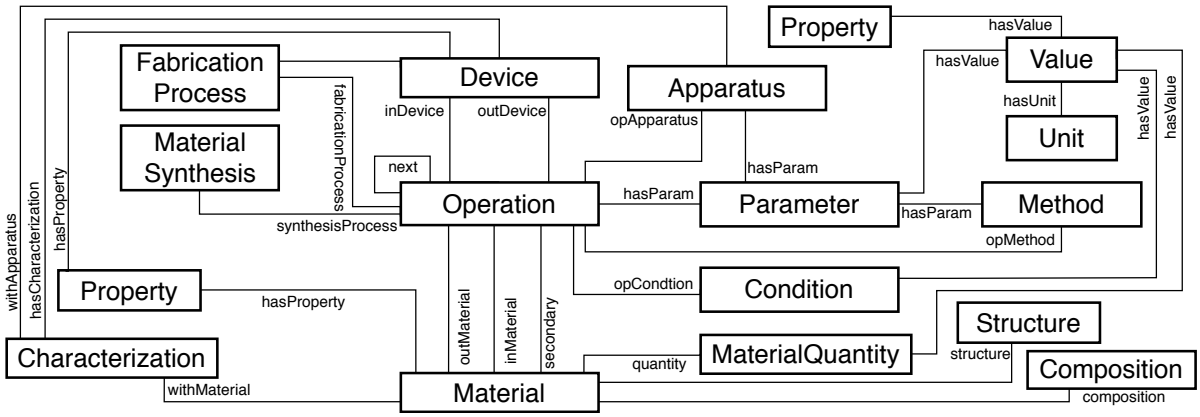

Figure 1: Ontology for representing device fabrication knowledge

Device fabrication and photovoltaic characterization: First, a thin (about 5 nm) PT layer was deposited on ITO-coated glass by electrochemical polymerization. Then the PT film was infiltrated with PbI2 by spin-coating at 8,000 rpm for 60 s with a PbI2 solution in dimethylformamide (DMF) (462 mg·mL–1, at 20 °C) in a N2 glove box. After drying, the film was dipped in a CH3NH3I solution in 2-propanol (10 mg·mL–1) at 70 °C for 90 s in air, then rinsed with 2-propanol. After the CH3NH3I perovskite was annealed at 100 °C for 40 min in air, C60 (30–50 nm)/BCP (5–15 nm) were deposited sequentially under high vacuum. Finally, Ag (100 nm) was thermally evaporated on top of the device to form the cell's back contact.

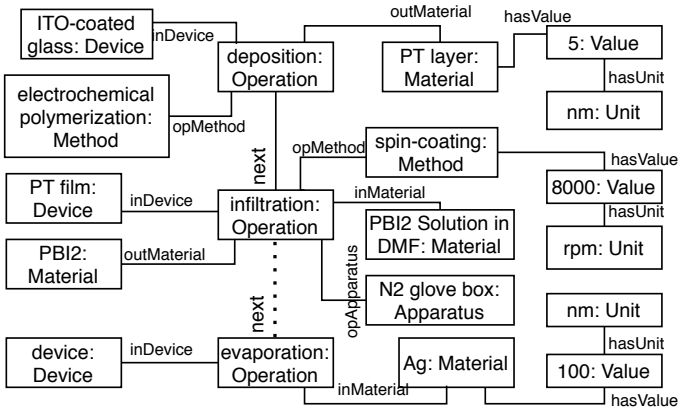

Figure 2: Text fragment from (Yan et al. 2015) describing solar cell fabrication and the corresponding knowledge graph

ing solar cell fabrication knowledge and the corresponding knowledge graph we extract from the fragment.

4 Knowledge Graph Extraction System

Our extraction system takes research articles in text form as input and first identifies the set of text fragments i.e. paragraphs describing device fabrication knowledge. It then processes these paragraphs to identify instances of domain entities and relations. A knowledge graph is then created combining the extracted instances. Our system applies the following three modules in a pipelined fashion.

- **Relevance Classifier:** The device fabrication knowledge is generally present in the article in a few sections or paragraphs such as *Experimental methods*, *Materials and methods*, etc. This module identifies such paragraphs by employing binary classification models. Our system provides support for three models: Logistic Regression (LR), BiLSTM and Hierarchical Attention Network (HAN) (Yang et al. 2016). The LR model builds a dictionary of unique words present in the document corpus and applies Bag of Words (BoW) to represent paragraph text. The BiLSTM model on the other hand takes the sequence of words into account. The individual words in a sequence are represented using their dense embeddings and the last hidden state of BiLSTM represents the paragraph text. The HAN model treats this as a document classification task. It first applies word level attention network to build sentence representations. It then applies sentence level attention to learn the paragraph representation. All the three models compose softmax layer on paragraph representation and apply cross-entropy loss to learn model weights.

- **Entity Extraction:** This module extracts instances of domain entities from the paragraphs identified by the relevance classifier. It first splits the paragraph into a sequence of sentences. It then applies entity extraction at sentence level by treating this as a sequence labelling task. Our system supports two deep learning models: BiLSTM; and BiLSTM-CRF. The BiLSTM model takes the sequence of tokens in a sentence as input and predicts labels for each token. The BiLSTM-CRF model (Huang, Xu, and Yu 2015) extends it by appending a linear-chain conditional random field (CRF) layer to the output labels thereby taking output label dependencies into account. It then uses Viterbi decoding to collectively predict the output label sequence as opposed to predicting tags for each token independently in BiLSTM. Both the models rep-

resent each token in the input sequence using following set of features: word embedding; character embeddings; Parts of Speech (PoS) tag; casing (features such as first character capital, all character capital, etc.); and concept type if present in the domain dictionary.

- **Relation Extraction:** This module identifies relations between extracted entities. This is performed in two stages. The module first identifies relations between entities present within a sentence. We use both deep learning and pattern based models for this task. The deep neural network based model first extracts Natural Language Processing (NLP) based features for the source (e_1) and target (e_2) entities. These features[1] include: next/previous words and their PoS tags; sequence of tokens in e_1 and e_2; dependency path between e_1 and e_2; their common ancestor in the dependency path, and so on. These features are then suitably represented: embeddings for words, one-hot vectors for PoS tags, BiLSTM for encoding token sequences (e.g. dependency path), etc. Finally, these features are passed to a deep feed forward neural network followed by a soft-max layer with cross-entropy loss. The pattern based model on the other hand uses regular expressions over token sequences and dependency relations to extract relation instances. We use TokensRegex (Chang and Manning 2014) and Semgrex (Chambers et al. 2007) from Stanford for this purpose. The regular expressions support various NLP features such as words, lemmas, PoS tags, entity types and dependency relations.

  In the next stage, we process the operation entities extracted from across the sentences to identify operation sequences that make up fabrication procedures. We apply a sequential model similar to (Mysore et al. 2017) for this purpose which links operations in the order in which they are mentioned in the paragraph. We also extract device characterization as a set of performance characteristics and their values. We use a dictionary based approach to locate instances of characteristics and then employ pattern based model to identify their values.

Our algorithms use domain specific knowledge in the form of pre-trained embeddings and domain dictionaries. In our deep learning models we use pre-trained word embeddings learned from a large materials dataset and fine tune them in a transfer learning setting. We support three types of embeddings: embeddings learned using word2vec objective: Mat-word2vec (Kim et al. 2017b) and mat2vec (Tshitoyan et al. 2019); embeddings with sub-word information: Mat-fasttext (Kim et al. 2020); and contextualized word embeddings: Mat-ELMo (Kim et al. 2020) and SciBERT (Beltagy, Lo, and Cohan 2019). A partial list of entities for various concepts such as operation, method, etc. are often available for a given application domain. Our system uses these concept dictionaries as features in the extraction models. It also supports patterns to extract task specific features.

[1]refer (Pawar, Bhattacharyya, and Palshikar 2017) for the complete list of features

5 Experimental Results

This section describes our experimental results on solar cell fabrication knowledge extraction.

5.1 Dataset

We downloaded 250 research articles containing solar cell fabrication knowledge from renowned journals. We used keywords such as perovskite, solar cell, fabrication, etc. and combinations there of to identify these articles. We then converted these PDF articles into text using Cermine (Tkaczyk et al. 2015). The xml tags present in Cermine output were used to demarcate paragraphs. These paragraphs were then manually annotated by domain experts. This resulted in a dataset of 412 instances for positive class containing device fabrication knowledge and 3884 instances for negative class. To create a dataset for learning entity and relation extraction models, we selected device fabrication paragraphs from 132 research articles (2916 sentences). These paragraphs were then manually annotated with entities and relations using brat annotation toolkit (Stenetorp et al. 2012). In total, we had 18683 and 15798 annotations for entities and relations respectively. We divided all of our datasets into train (70%), dev (10%) and test (20%) set. The results are reported in terms of precision, recall and F1-score.

5.2 Relevance Classifier

The relevance classifier models are trained using the annotated paragraphs. In addition to the features described in section 4, we also leverage dictionary and pattern based features. We created partial dictionaries for various concepts such as operation, method, apparatus, etc. in the solar cell domain. We also created a dictionary of section header names by applying patterns (e.g.: `^(\d.)?([A-Z](\S+\s*){1,3})[.:\n]`) to each paragraph in the training set. A set of binary features are designed to indicate presence of a token in a given dictionary. These dictionary features are concatenated with token features for model training. Our dataset contains only 9.5% of instances for the positive class that we are primarily interested in. To account for this data skewness, we updated our loss function such that the cost for misclassifying positive class instance is considerably higher than the negative class. Table 1 reports our results on identifying fabrication paragraphs. The baseline logistic regression model achieves F1-score of 0.79. The recurrent neural network models in BiLSTM and HAN improve precision by about 5% resulting in overall improvement in F1-score. Contrary to what is reported in literature, the hierarchical model in HAN did not improve beyond standard BiLSTM in our case.

Model	Precision	Recall	F1
LogisticRegression	0.8819	**0.7152**	0.7879
BiLSTM	0.9286	0.7143	**0.8075**
HAN	**0.9403**	0.6923	0.7975

Table 1: Results on relevance classifier models

PreTrained Embeddings	Strict Matching						Relaxed Matching					
	BiLSTM			BiLSTM-CRF			BiLSTM			BiLSTM-CRF		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Mat-word2vec	0.650	0.688	0.668	0.687	0.711	0.699	0.672	0.715	0.693	0.719	0.753	0.736
mat2vec	0.650	0.688	0.668	0.655	0.689	0.672	0.672	0.714	0.693	0.681	0.723	0.701
Mat-fasttext	0.672	0.694	0.683	0.676	0.699	0.687	0.689	0.719	0.704	0.694	0.725	0.709
Mat-ELMo	0.681	0.696	0.688	**0.733**	0.703	**0.718**	0.708	0.732	0.719	**0.765**	0.744	**0.754**
SciBERT	0.681	0.723	0.702	0.668	**0.725**	0.695	0.709	0.760	0.733	0.696	**0.762**	0.727

Table 2: Entity extraction results on BiLSTM and BiLSTM-CRF models

Type	Prec	Rec	F1
operation	0.828	0.753	0.789
method	0.634	0.662	0.648
material	0.649	0.628	0.638
device	0.588	0.617	0.603
apparatus	0.802	0.786	0.794
parameter	0.488	0.385	0.430
paramVal	0.923	0.938	0.930
property	0.609	0.483	0.538
propVal	0.685	0.768	0.724
unit	0.873	0.943	0.907
condition	0.662	0.613	0.636
conditionVal	0.333	0.154	0.211
matQuantity	0.400	0.105	0.167
matQuantityVal	0.804	0.860	0.831

Table 3: Entity type level results using BiLSTM-CRF with Mat-ELMo embeddings

5.3 Entity Extraction

We evaluate our entity extraction models with two settings: Strict matching; and Relaxed matching. In strict matching, an extracted entity spanning multiple tokens is considered true positive only when both the predicted label as well as the entity span match with the annotated data. In the relaxed matching setting, we moderate the definition of true positive to practically evaluate our models. Consider a tagged entity: *pre-cleaned FTO glass substrate*. A trained model partially identifying this entity as *FTO glass* or *FTO glass substrate* is still useful from practical usage standpoint. We devised a set of partial matching rules by analysing the training set. An extracted entity is then considered true positive when the predicted label matches with the annotated data and the entity span satisfies the partial matching rules. We use BIO encoding for representing entities spanning multiple tokens.

Table 2 reports our results on entity extraction models. The BiLSTM-CRF model using contextualized embeddings in Mat-ELMo achieves highest performance with F1-score of 0.72 for strict matching and 0.75 for relaxed matching. These scores are better than the corresponding human F1 score[2] of 0.68 and 0.72 for strict and relaxed matching respectively. Overall, we observe that: (1) the output label dependencies captured by the CRF layer help improve the model performance in BiLSTM-CRF; (2) The contextualized embeddings perform better than the embeddings learned using word2vec objective. Table 3 shows detailed results for all entity types. The model extracts operation, method and apparatus entities with good accuracy. The device and material entities look quite similar and appear in similar contexts. Consider a text fragment: *The glass substrate was heated at 70 C to form the perovskite film with 300m thickness.* Though *perovskite* is generally considered a material, the fragment here discusses *perovskite film* as a device. This results in some amount of loss in extraction accuracy for material and device. The entities for types such as parameter, property etc. are not specified explicitly in text. Consider a phrase: *spin-coating the solution at 2000 rpm for 60s.* Here, both the parameters *speed* and *time* are not explicitly mentioned. This results in insufficient annotations for these entity types resulting in low extraction accuracy. However, parameter values are extracted with a high degree of accuracy as values have well defined patterns.

5.4 Relation Extraction

The relation extraction models take source and target entities along with the sentence as input and predict a relation label between them (if it exists). We evaluate our models with three settings: Gold-ER; Strict-ER; and Relaxed-ER. Being a pipelined system, errors from entity extraction models directly affect relation extraction performance. Gold-ER setting avoids this by considering gold (manual) entity annotations as input. This helps us in evaluating our relation extraction models as a separate system. The other two settings are similar to the ones in entity extraction and evaluate relation extraction models in an end-to-end scenario. In the Strict-ER setting, a relation classification instance is considered as true positive only when 1) the labels of the source and target entities are correct and their spans completely match with the annotated data; and 2) the relation label is correct. The Relaxed-ER setting on the other hand relaxes the first condition by applying partial matching rules (refer section 5.3) in place of exact span matching for the source and target entities.

We applied deep neural network based model for relations involving material and device (total 6 relations in fig.

[2]Human F1 is a practical way of quantifying the difficulty of a given classification task (similar to inter-annotator agreement). F1 score for a pair of annotators A1 and A2 is computed by considering the annotations from A1 as the output of a classification algorithm and the annotations from A2 as the gold annotations. Human F1 is then defined as the mean of the F1 scores from all pairs of annotators.

opApparatus

1. {ner:/operation/}=operation </.*/ ({word:/carried/} >/nmod:in/ {ner:/apparatus/}=apparatus)
2. {ner:/operation|method/}=operation >/nmod:through/ {ner:/apparatus/}=apparatus)

hasValue

1. ([{tag:CD}]) [word:/:/] ([{tag:CD}]) [{ner:/unit/}] [word:/solution|suspension/] [word:/of/] ([{ner:/solvent |material/}]) [word:/in|:/] ([{ner:/solvent|material/}])
2. [{tag:CD}|ner:/conditionVal|paramVal|propVal|matQuantityVal/=value] </.*/ ({ner:/unit/} >/nmod:of/ {ner:/material|method|operation|parameter|property|solute|solvent/}=entity)

hasUnit

1. ([{tag:CD}|{ner:/conditionVal|paramVal|propVal|matQuantityVal/}]) ([{ner:"unit"}])

hasProperty

1. {ner:/property/}=prop >/nmod:of|amod/ {ner:/device|material|solute|solvent/}=device
2. {ner:/device|material|solute|solvent/}=device [>/nmod:(with|as|of)/ {ner:/property/}=prop | </nsubj.*/ {ner:/property/}=prop]

opMethod

1. {ner:/operation/}=operation </amod/ ({tag:/NN.*/} </nsubj.*/ {ner:/method/}=method)
2. {ner:/operation/}=operation >/nmod:(agent|by)/ {ner:/method/}=method

opCondition

1. {ner:/method|operation/}=method >/nmod:at/ ({ner:/unit/} >/nmod:for/ ({ner:/unit/} >/nmod:(under|in)|amod/ {ner:/condition/}=condition))
2. {ner:/method|operation/}=method </.*/ ({tag:/VB.*|NN.*/} >/nmod:(in|at|under)/ {ner:/condition/}=condition)

Table 4: Patterns used in pattern based relation extraction model

1). It uses the set of NLP features described in section 4 to represent inputs. Other relations were extracted using pattern based model as they have well defined patterns. These patterns are specified using regular expressions over token stream and dependency paths. A few example patterns and the corresponding extracted relation instances include,

- **opMethod:** {ner:/operation/}=operation >/nmod:(agent|by)/ {ner:/method/}=method – **example:** Perovskite films were **deposited** by **spin-coating** the $CH_3NH_3PbI_3$ precursor at 5000 rpm for 30s on the ITO glass substrate.
- **opApparatus:** {ner:/operation/}=operation >/nmod:(in|with)/ {ner:/apparatus/}=apparatus – **example:** The resulting layers were subsequently **annealed** at 150C for 10 min in a **glove box**.

The complete list of patterns are reported in table 4.

Table 5 reports our results[3] for both deep learning as well as pattern based models. Overall, the pattern based model achieves F1-score of about 0.8 for all relations (except *hasValue*) in Gold-ER setting. On the other hand, the deep learning model for relations involving materials (*inMaterial*, *secondary*, *outMaterial*) achieves F1-score of about 0.6. The operations in solar cell fabrication utilize various kinds of materials including precursors, secondary materials, etc. Due to this, the problem of identifying relationship type for a given operation-material pair is difficult resulting in lower F1-score. In the end-to-end evaluation scenario, the performance of our algorithms decreases significantly compared to the Gold-ER setting. This is primarily due to the fact that misclassification of either the source or the target entity, both affect the precision and recall of relation extraction. For relations such as *hasProperty* and *hasParam*, the number of instances are quite low due to implicit mentions of participating entities i.e. property and parameter, affecting the quality of our models. The performance on these relations degrades further in the end-to-end setting.

The sequential unsupervised model described in section 4 works well for the solar cell fabrication domain resulting in F1-score of 0.86 for procedure extraction (i.e. operation sequence extraction). As with sentence level relations, the performance decreases in the end-to-end setting due to entity extraction errors. We also extracted performance characterization for identified fabrication procedures. We used dictionary based matching to extract performance metrics such as open circuit voltage (V_{oc}), power conversion efficiency (PCE), and so on. We then applied pattern based value relation extraction using patterns similar to the ones discussed in (Shah et al. 2018). We achieved a precision of 0.91 and a recall of 0.623. The recall can further be improved by adding additional patterns for characterization value extraction.

6 Knowledge Guided Decision Support

The knowledge graphs extracted by our system can be queried to make informed decisions during device fabrication. Consider an example, suppose a designer wants to fabricate a solar cell with V_{oc} of 1.15 V among other standard characteristics and decides to use $MaPbBr_3$ perovskite. The knowledge graphs can be queried to identify solar cells with similar characteristics. Analysing these devices helps the designer narrow down the design space of materials, opera-

[3]Table 5 does not include a few relations from Figure 1. We are not mining Structure and Composition since these can largely be obtained from catalogues once the material is known. Hence the corresponding relations are also not mined. MaterialSynthesis and FabricationProcess stand for root entities that represent sequence of operations, hence created implicitly.

Relation Type	Gold-ER			Strict-ER			Relaxed-ER		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Supervised relation extraction using deep learning with NLP features									
secondary	0.5913	0.5074	0.5461	0.3873	0.3208	0.3510	0.3873	0.3208	0.3510
inMaterial	0.5646	0.6803	0.6171	0.2782	0.2759	0.2770	0.3140	0.3114	0.3127
outMaterial	0.5809	0.5865	0.5837	0.2448	0.2307	0.2376	0.2551	0.2403	0.2475
inDevice	0.7681	0.8502	0.8071	0.3677	0.4385	0.4	0.4215	0.5026	0.4585
outDevice	0.7088	0.6292	0.6666	0.2833	0.1910	0.2281	0.3	0.2022	0.2416
hasParam	0.7333	0.44	0.5499	0.25	0.0980	0.1408	0.25	0.0980	0.1408
Pattern based relation extraction									
opApparatus	0.8901	0.7788	0.8308	0.6190	0.5	0.5531	0.6309	0.5096	0.5638
hasValue	0.8577	0.5540	0.6732	0.6003	0.3985	0.4790	0.6153	0.4084	0.4910
hasUnit	0.9953	0.8740	0.9307	0.8	0.7534	0.7760	0.8294	0.7811	0.8045
hasProperty	0.88	0.7586	0.8148	0.1875	0.1034	0.1333	0.25	0.1379	0.1777
opMethod	0.9109	0.9293	0.9199	0.6310	0.6565	0.6435	0.6504	0.6767	0.6633
opCondition	0.8667	0.7027	0.7761	0.5178	0.3918	0.4461	0.5357	0.4054	0.4615
Unsupervised fabrication procedure extraction									
next	0.8493	0.8664	0.8578	0.5159	0.4699	0.4918	0.5159	0.4699	0.4918

Table 5: Relation extraction results for deep neural network and pattern based models

tions and operating conditions. Further to this, suppose the designer wishes to carry out the fabrication in air as opposed to the standard practice of using a glove box. This constraint can be utilized to narrow down the design space further. The knowledge graphs help in providing design decision support of the kind described above using suitable query and analysis tools.

7 Path to Deployment

Our company is developing a knowledge-guided, simulation aided materials engineering platform to shorten lab to market of new devices and materials. The platform contains a pipeline of four modules: materials design and selection; device characterization; knowledge engine; and simulation framework. The materials design and selection module contains models (physics/data based) for in-silico design of new materials (Jain et al. 2019a; Ravikumar, Mynam, and Rai 2018). The device characterization module on the other hand is used to define the desired device behaviour (e.g. V_{oc}, I_{sc}, etc. in case of solar cells) for a given application (batteries, solar cell, fuel cell, etc.). For the selected material, the knowledge engine module then helps in identifying potential device fabrication procedures that can achieve the desired device characterization. The decision support system of the kind described earlier in section 6 makes up the primary component of the knowledge engine. The set of fabrication procedures shortlisted using the knowledge engine are then put through the simulation framework to further narrow down the candidates for lab scale experimentation. Of the four modules described above, two are already in place. The extracted knowledge graphs form the core of the knowledge engine module over which we are currently building a decision support layer. Another team is parallelly working on the simulation framework which is also nearing completion. Once these two modules are complete, we are ready to deploy.

8 Summary and Future Work

We present a system to automatically extract device fabrication knowledge from materials science literature. It applies entity and relation extraction models to construct knowledge graphs conforming to an ontology that can be queried to make informed decisions during device fabrication. We applied our system on solar cell fabrication knowledge extraction and achieved good initial results.

We are currently working on the decision support layer that can provide querying and analytics over extracted knowledge graphs. We then want to validate the end to end savings in time and efforts our materials engineering platform can provide in experimentally designing devices, solar cells in our case. We also plan to work further on strengthening our extraction models. For instance some of the inaccuracies reported in section 5.3 are due to missing details in sentences. A domain expert can easily fill in these details from prior knowledge. We want to explore how such domain knowledge can be integrated into the models to improve the accuracy.

Acknowledgements

We thank Venkatamuralidhar K, Jayita Dutta, Varnita Bajpai, Shally Gupta, Santosh Daware, Rinu Chacko, Pallavi Bandi, Dharmendr Kumar, Abhishek Agarwal, Shashank Mishra and Shankar Kausley, our colleagues at Tata Consulatncy Services Ltd., for helping us with manual data annotation.

References

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. Hong Kong, China: Association for Computational Linguistics.

Chambers, N.; Cer, D.; Grenager, T.; Hall, D.; Kiddon, C.; MacCartney, B.; de Marneffe, M.-C.; Ramage, D.; Yeh, E.; and Man-

ning, C. D. 2007. Learning Alignments and Leveraging Natural Logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE 07, 165170. USA: Association for Computational Linguistics.

Chang, A. X.; and Manning, C. D. 2014. TokensRegex: Defining cascaded regular expressions over tokens. Technical Report CSTR 2014-02, Department of Computer Science, Stanford University.

Chen, X.; Huang, H.; Pan, L.; Liu, T.; and Niederberger, M. 2019. Fully integrated design of a stretchable solid-state lithium-ion full battery. *Advanced Materials* 31(43): 1904648.

Fu, Q.; Tang, X.; Huang, B.; Hu, T.; Tan, L.; Chen, L.; and Chen, Y. 2018. Recent progress on the long-term stability of perovskite solar cells. *Advanced Science* 5(5): 1700387.

Gómez-Bombarelli, R.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Duvenaud, D.; Maclaurin, D.; Blood-Forsythe, M. A.; Chae, H. S.; Einzinger, M.; Ha, D.-G.; Wu, T.; et al. 2016. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nature materials* 15(10): 1120–1127.

Gruber, T. 2009. *Ontology*, 1963–1965. Boston, MA: Springer US. ISBN 978-0-387-39940-9. doi:10.1007/978-0-387-39940-9_1318.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *ArXiv* abs/1508.01991.

Jain, D.; Chaube, S.; Khullar, P.; Srinivasan, S. G.; and Rai, B. 2019a. Bulk and surface DFT investigations of inorganic halide perovskites screened using machine learning and materials property databases. *Physical Chemistry Chemical Physics* 21(35): 19423–19436.

Jain, D.; Dwadasi, B. S.; Kumar, D.; Mishra, S.; Ravikumar, B.; Gupta, R.; Srinivasan, S. G.; Jain, V.; Mynam, M.; Maiti, S.; et al. 2019b. Materials Design in Digital Era: Challenges and Opportunities. *Transactions of the Indian Institute of Metals* 72(8): 2199–2208.

Jena, A. K.; Kulkarni, A.; and Miyasaka, T. 2019. Halide perovskite photovoltaics: background, status, and future prospects. *Chemical reviews* 119(5): 3036–3103.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *CoRR* abs/2002.00388. URL https://arxiv.org/abs/2002.00388.

Kaur, N.; Mahajan, A.; Bhullar, V.; Singh, D. P.; Saxena, V.; Debnath, A.; Aswal, D.; Devi, D.; Singh, F.; and Chopra, S. 2019. Fabrication of plasmonic dye-sensitized solar cells using ion-implanted photoanodes. *RSC advances* 9(35): 20375–20384.

Kim, E.; Huang, K.; Saunders, A.; McCallum, A.; Ceder, G.; and Olivetti, E. 2017a. Materials synthesis insights from scientific literature via text extraction and machine learning. *Chemistry of Materials* 29(21): 9436–9444.

Kim, E.; Huang, K.; Tomala, A.; Matthews, S.; Strubell, E.; Saunders, A.; McCallum, A.; and Olivetti, E. 2017b. Machine-learned and codified synthesis parameters of oxide materials. *Scientific data* 4: 170127.

Kim, E.; Jensen, Z.; van Grootel, A.; Huang, K.; Staib, M.; Mysore, S.; Chang, H.-S.; Strubell, E.; McCallum, A.; Jegelka, S.; and Olivetti, E. 2020. Inorganic Materials Synthesis Planning with Literature-Trained Neural Networks. *Journal of chemical information and modeling* .

Kim, H.-S.; Lee, C.-R.; Im, J.-H.; Lee, K.-B.; Moehl, T.; Marchioro, A.; Moon, S.-J.; Humphry-Baker, R.; Yum, J.-H.; Moser, J. E.; et al. 2012. Lead iodide perovskite sensitized all-solid-state submicron thin film mesoscopic solar cell with efficiency exceeding 9%. *Scientific reports* 2(1): 1–7.

Kuniyoshi, F.; Makino, K.; Ozawa, J.; and Miwa, M. 2020. Annotating and Extracting Synthesis Process of All-Solid-State Batteries from Scientific Literature. *arXiv preprint arXiv:2002.07339* .

Mysore, S.; Kim, E.; Strubell, E.; Liu, A.; Chang, H.-S.; Kompella, S.; Huang, K.; McCallum, A.; and Olivetti, E. 2017. Automatically extracting action graphs from materials science synthesis procedures. *arXiv preprint arXiv:1711.06872* .

Park, N.-G. 2020. Research direction toward scalable, stable, and high efficiency perovskite solar cells. *Advanced Energy Materials* 10(13): 1903106.

Pawar, S.; Bhattacharyya, P.; and Palshikar, G. 2017. End-to-end Relation Extraction using Neural Networks and Markov Logic Networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 818–827. Valencia, Spain: Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1077.

Ravikumar, B.; Mynam, M.; and Rai, B. 2018. Effect of salt concentration on properties of lithium ion battery electrolytes: a molecular dynamics study. *The Journal of Physical Chemistry C* 122(15): 8173–8181.

Segler, M. H.; Preuss, M.; and Waller, M. P. 2017. Learning to plan chemical syntheses. *arXiv preprint arXiv:1708.04202* .

Shah, S.; Vora, D.; Gautham, B. P.; and Reddy, S. 2018. A Relation Aware Search Engine for Materials Science. *Integrating Materials and Manufacturing Innovation* 7(1): 1–11.

Stenetorp, P.; Pyysalo, S.; Topić, G.; Ohta, T.; Ananiadou, S.; and Tsujii, J. 2012. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 102–107. Avignon, France: Association for Computational Linguistics.

Tkaczyk, D.; Szostek, P.; Fedoryszak, M.; Dendek, P. J.; and Bolikowski, L. 2015. CERMINE: Automatic Extraction of Structured Metadata from Scientific Literature. *Int. J. Doc. Anal. Recognit.* 18: 317335. ISSN 1433-2833. doi:10.1007/s10032-015-0249-8.

Tshitoyan, V.; Dagdelen, J.; Weston, L.; Dunn, A.; Rong, Z.; Kononova, O.; Persson, K.; Ceder, G.; and Jain, A. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* 571: 95–98. doi:10.1038/s41586-019-1335-8.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 5998–6008.

Yan, W.; Li, Y.; Li, Y.; Ye, S.; Liu, Z.; Wang, S.; Bian, Z.; and Huang, C. 2015. Stable high-performance hybrid perovskite solar cells with ultrathin polythiophene as hole-transporting layer (Supplementary information). *Nano Research* 8(8): 2474–2480.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489. San Diego, California: Association for Computational Linguistics. doi:10.18653/v1/N16-1174. URL https://www.aclweb.org/anthology/N16-1174.