Named Entity Recognition for Performance and Synthesis Information of Perovskite Solar Cells Using SpaCy

Mary Zuleika Jiménez-Díaz(✉), Alexander Sepúlveda-Sepúlveda, and Mónica Botero-Londoño

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Universidad Industrial de Santander, Bucaramanga, Santander, Colombia
mary2238319@correo.uis.edu.co, alexander.sepulveda@saber.uis.edu.co, mabotero@uis.edu.co

Abstract. This paper presents a Named Entity Recognition (NER) approach to extract key information related to the structural components, synthesis techniques, and photovoltaic performance metrics of Perovskite solar cells (PSCs). Since NER is a key component of Information Extraction (IE), which identifies and classifies key elements of a text, we propose the use of the *SpaCy* library to build an effective and accessible NER model, suitable for environments with limited computational capacity, unlike other previous works in this field, which make use of large-scale models or high computational resources. Our resulting model was evaluated using *K-fold cross-validation*, obtaining the mean scores of, precision of 89.94%, a recall of 92.47%, and an F1 score of 89.69%. To provide a test of the practical performance of the resulting model, implementing and comparing the obtained results with manual annotations in two Excel reference databases: Odabaşı (2019) [1] and Jacobsson (2022) [2], demonstrating the potential of our work to facilitate and accelerate knowledge extraction, and the possibility of extending this strategy to other scientific fields where automated text extraction is required.

Keywords: Natural language processing · NER · Perovskite Solar Cells · Spacy · Machine Learning · Named Entity Recognition

1 Introduction

In recent years, the development of renewable energy sources with low environmental impact has gained increasing attention, driven by global initiatives to limit global warming and mitigate the costs of traditional energy. According to the REN21 report [3], investment in renewable energy reached a historic high in 2023, reflecting its accelerated growth.

Perovskite solar cells (PSCs), classified as third-generation photovoltaic devices, have emerged as a strong contender to silicon-based technologies, primarily due to their high power conversion efficiency (PCE) and potential cost

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15649, pp. 303–319, 2025.
https://doi.org/10.1007/978-3-031-96997-3_19

advantages. Laboratory studies have reported PCE values exceeding 26% [4], and ongoing research suggests that advances in manufacturing techniques could lead to lower production costs [5–7]. However, their operational stability and material optimization remain key challenges [8,9]. The discovery of new high-performance PSC materials and fabrication techniques still requires extensive experimental validation, which increases both costs and development time despite advances in computational modeling.

PSC data can come from simulations or experimental measurements. While simulations are widely used due to their lower costs [10–13], they do not always account for material defects or real operating conditions. As a result, experimental data remain essential for validating material properties and device performance, but obtaining and analyzing this information manually is time-consuming and costly.

The rise of PSCs and their rapid growth have resulted in a substantial increase in the number of scientific papers, making them one of the fastest growing topics in the area of photovoltaic generation research [14]. This growing amount of information is a valuable resource for materials discovery and analysis. In order to reduce the time and cost associated with prediction, analysis, and the optimization of PSC configurations and synthesis methods, machine learning-based approaches have been proposed [1,15–19]. However, the unstructured information contained in the overwhelming volume of scientific articles makes manual data collection challenging [2], thus highlighting the need for automated tools to enhance the efficiency of relevant information extraction.

The information extraction (IE) can be simplified into two main tasks: Named Entity Recognition (NER) and Relationship Extraction (RE) [20]. Where, the NER task consists of analyzing a text to locate and classify unitary words, named "Named Entities" into categories defined for the target task, as an example, the word "MAPbI3" can be defined as a Named Entity of the "Absorbent layer of the cell structure" category. Then, the RE task is in charge of identifying and classifying the contextual and/or grammatical relationships between the Named Entities recognized in the NER task.

The data extraction process and the NER task start with the conversion of the documents to plain text from documents usually in PDF or HTML. Then the texts are preprocessed to de-encode, standardize and segment the information, and finally, the texts are analyzed using computer tools and natural language processing (NLP) techniques suitable for the NER task to retrieve the relevant data. Various tools exist for scientific information extraction, such as ChemDataExtractor [21], ChemicalTagger [22], and OSCAR4 [23], which focus on extracting scientific entities related to elements and chemical compounds [24]. Transformer-based models (LLMs) have also been developed, with platforms like Hugging Face providing widely used open-source libraries for implementing and fine-tuning these models [25]. However, these tools are not optimized for PSCs and do not offer an efficient, low-computational-cost approach for training custom NER models. Without additional fine-tuning on domain-specific data,

they cannot accurately identify key entities related to synthesis methods, device performance, and structural composition.

Some recent studies have addressed automated information extraction in PSCs. *Beard et al.* [26] employed *ChemDataExtractor* as a baseline model for the automatic generation of PSC databases, demonstrating how automation can accelerate scientific data collection. *Xie et al.* [27] proposed an approach based on large language models (LLMs) and convolutional networks for structuring a dataset and predicting PSC performance. Although these methods have shown significant advances, they often require high computational resources and are not designed for environments with limited hardware or small research teams, nor do they enable the rapid, cost-effective, and semi-automated development of custom NER models.

Given this scenario, it is necessary to explore alternatives that enable the development of optimized NER models for PSCs, with more accessible and resource-efficient implementation.

In this paper, we present an approach to perform NER tasks on PSC-related data from PDF and HTML files. Where we train a model using the SpaCy NLP library [28], to extract entities related to the performance, synthesis methods, and structural composition of PSCs. The final model obtained was trained, tested, evaluated, and validated using a dataset of 584 PDF articles about conventional PSCs, as reported by Odabaşı (2019) [1], and 4 HTML articles extracted from Jacobsson (2022) [2]. Therefore, we developed a proposal that offers a specialized, accessible, and adaptable alternative to training NER models in environments with limited computational resources.

2 Method

In this work, we use *SpaCy* [28], an open-source Python library. It provides natural language processing tools such as tokenization, part-of-speech (POS) tagging, dependency parsing, named entity recognition (NER), relation extraction, lemmatization, text classification, customizable pipelines, word embeddings, and similarity analysis, among others. These tools enable the training of a customized model tailored to the linguistic and vocabulary needs of a specific task. However, before this, it is necessary to obtain training data, considering the semantic and linguistic specifications relevant to the task at hand.

2.1 Named Entity Recognition (NER)

In our work, we describe the basic structure of a Perovskite solar cell with three main layers: a Perovskite layer, which acts as the active material and absorbs light; an electron transport layer (ETL), which facilitates electron movement; and a hole transport layer (HTL), which enables the flow of positive charge. This arrangement corresponds to the standard structure of conventional Perovskite solar cells, as illustrated in Fig. 1.

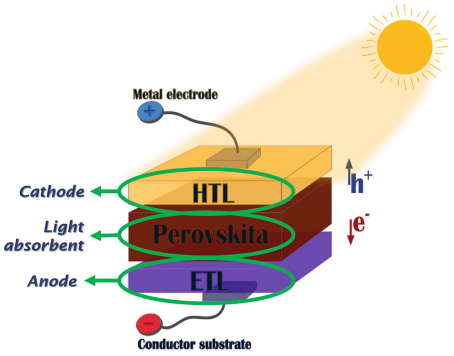


Fig. 1. Illustration of the schematic configuration of Perovskite solar cells.

In this context, we use Named Entity Recognition (NER), a fundamental technique in natural language processing (NLP) that identifies and classifies entities within a text, such as names of people, organizations, locations, or dates. In our case, NER was applied to identify and classify entities specifically related to Perovskite solar cells (PSC) in a set of scientific articles. The key entities used in this study are presented in Table 1.

Table 1. Named entity definitions used in the Model.

Category	Description	Character Type	Entity Label
Performance-related	Power Conversion Efficiency	Alphabetic	PCE
		Numeric	NPCE
	Open Circuit Voltage	Alphabetic	VOC
		Numeric	NVOC
	Short Circuit Current	Alphabetic	JSC
		Numeric	NJSC
	Filling Factor	Alphabetic	FF
		Numeric	NFF
Cell Structure	Hole Transporting Layer	Alphabetic	HTL
	Perovskite Absorber Layer	Alphabetic	PTF
	Electron Transporting Layer	Alphabetic	ETL
Synthesis-related	Synthesis Methods	Alphabetic	MTDS
		Alphabetic	VMTDS
	Method Variables	Numeric	NMTDS
	Environmental Conditions	Alphabetic	ENMTDS

2.2 Data Collection and Preprocessing

The dataset consists of 3,278 sentences derived from 584 articles in PDF format, selected from a database of 800 articles on Perovskite solar cells, published between 2013 and 2018. This database, manually compiled in the study reported by Odabaşı (2019) [1], contains information on a total of 1.921 PSCs. Of these, 584 articles were selected for providing specific information about conventional PSC structures. These structures generally correspond to those employing standard Perovskite materials as the light-absorbing layer.

To construct the training set, the selected 3,278 sentences were manually extracted from the training documents, with 3,088 sentences allocated for training ($\approx$ 94%) and 190 sentences for evaluating the NER model ($\approx$ 6%). These sentences may or may not contain information related to the defined entities, as presented in the *Named Entity Recognition (NER)* section and Fig. 1 and Table 1.

After manually selecting the sentences, a search was performed for the entities in each sentence, identifying their positions and labels using the methods in the *spaCy* library. These methods allow the use of a lightweight pre-trained NER model in the corresponding language, for the identification and initial assignment of named entities to each defined label (Table 1). Using them required the manual creation of lists of terms associated with specific labels, such as HTL, ETL, PTL, or MTDS, along with the use of patterns that use a modified version of regular expressions to obtain other labels, such as NPCE, NVOC, NJSC, or NMTDS, which involve numerical values and units of measurement.

Although these *spaCy* methods enables named entity identification without training, its accuracy is low and its capabilities are highly limited. For these reasons, these techniques were used only to generate an initial labeled training dataset. This first set, composed of sentences and their respective entities, was stored in JSON format using Python dictionaries. Figure 2 illustrates an example of the structure of the training dataset, showing the extracted sentence, the entities, and their positions within the text.

```
[
    "The stability of spiro-MeOTAD/CH3NH3PbI3/ZnO nanorod devices was
    investigated by storing them in air at room temperature without
    further encapsulation.\n",
    {
        "entities": [
            [41,44,"ETL"],[30,40,"PTF"],[17,29,"HTL"]
        ]
    }
],
[
    "The spin coating program includes two steps, first 1000 rpm for
    10 s with a ramp of 200 rpm s-1, then 6000 rpm for 30 s with a
    ramp of 2000 rpm s-1.\n",
    {
        "entities": [
            [4,16,"MTDS"],[76,80,"VMTDS"],[127,131,"VMTDS"],
            [51,59,"NMTDS"],[64,68,"NMTDS"],[102,110,"NMTDS"],
            [115,119,"NMTDS"],[135,147,"NMTDS"],[84,95,"NMTDS"]
        ]
    }
],
```

Fig. 2. Example of a training sentence, composed of the extracted sentence from the article, the entities, and their positions in the text.

Upon verifying the training dataset, issues such as entity duplication, labeling errors, and failures in locating entities within the text were identified. Additionally, inconsistencies in character encoding were found due to the lack of standardization in the PDF files.

To address these challenges, we developed an internal beta-phase software tool with a graphical interface for the semi-automatic correction of the dataset, providing a customized and more accessible alternative to the commercial annotation tool *Prodigy* [29]. The software, initially designed by us in a previous work [30] and further improved for this study, was developed in *Python* using *PyQt5* and *QtDesigner*. Figure 3 shows the main window of the tool and shows how it facilitates the user to load, edit and delete selected named entities from a sentence, through its different options. To generate the final labeled training dataset, we performed two semi-automated reviews using our tool: the first was carried out individually, and the second in collaboration with one of the co-authors, a materials science professor who observed, resolved ambiguous cases, and corrected the entity labeling in each sentence.

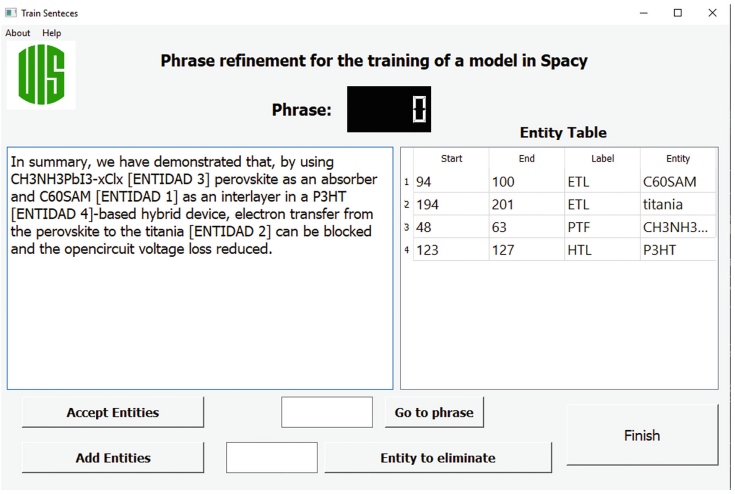


Fig. 3. Main window of the graphical interface tool used for the semi-automatic correction of the data file.

2.3 Model Training

To train the new NER model in *spaCy*, both rule-based entity recognition (*Rule-based NER*) and supervised training using neural networks were employed. A total of 16 new entities labels were defined: “HTL”, “ETL”, “PTF”, “VOC”, “PCE”, “JSC”, “FF”, “NVOC”, “NPCE”, “NJSC”, “NFF”, “MTDS”, “VMTDS”, “ENMTDS”, and “NMTDS”. These correspond to the main layers, performance

parameters, and synthesis variables of PSCs, as described in the previous section, *Named Entity Recognition (NER)*, and illustrated in Table 1.

Using the set of sentences obtained in the *Data Collection and Pre-processing* section, the model was trained. Multiple versions of the model were trained, varying the hyper-parameters: N, the number of iterations; D, the dropout rate; and B, the batch size per iteration. Finally, the model with the best performance was selected, based on the evaluation metrics defined in the following section.

2.4 Model Evaluation

Once the model was trained, its performance was evaluated using 190 new test sentences, distinct from those used during training. For this, we used the *scores* and *evaluate* functions from *spaCy*, which allow us to compute evaluation metrics: **Precision**, which measures the proportion of correctly identified entities relative to the total model predictions; **Recall**, which measures the proportion of true entities correctly identified; and **F1-Score**, which represents the harmonic mean between precision and recall [31,32].

To validate the model, we used two methods. The first was the *k-Fold Cross-Validation* method with the 3,088 training sentences. This method is recommended when working with a limited dataset, as it reduces variance in error estimation without a high computational cost compared to other methods. Although it is not included in spaCy 3.0, this validation can be performed in Python using the *scikit-learn* library [33], employing the *sklearn.model_ selection.KFold* function to split the data into K validation groups.

As a second method, we conducted an additional evaluation of the model using a dataset different from the one used in training. For this, we used the Perovskite solar cell dataset from Jacobsson (2022) [2], which contains more than 400 variables and 42,000 records of Perovskite solar cells collected between 2003 and 2020. We manually downloaded four articles in HTML format. These were then converted into plain text programmatically, sectioned, cleaned, and refined by removing irrelevant symbols that could interfere with model predictions and standardizing part of the chemical nomenclature used in materials science. For validation, we compared our model's extractions with the data available in Jacobsson's (2022) dataset.

Furthermore, we evaluated and measured inference times across different file formats (PDF, HTML, plain text, and sentence-level processing) in the following scenarios: a single sentence, a PDF file, an HTML file, a set of 584 PDF files, and a set of four HTML files. Testing was performed on a system with a 3rd-generation Intel Core i7 processor, 16 GB of RAM, and Windows 10.

3 Application Example

In this section, as an application example, we perform automatic information extraction from conventional PSC articles using the Odabaşı (2019) [1] database. To do this, we use the *PyMuPDF* [34], Data Collection library to extract plain

text from PDFs. Data is then preprocessed by text cleaning and standardization to improve the detection of our entities in the NER task. Finally, we compare the entities associated with the cell structure, extracted by the model with those obtained from the manually extracted Odabaşı (2019) [1] database.

3.1 Data Processing

Scientific articles in PDF format use an extended ASCII encoding, which varies depending on the software used to generate them. This makes text extraction difficult, especially due to special symbols. To solve this problem, we used the *PyMuPDF* library, which allows articles to be converted to plain text. We then preprocessed the text of the articles, segmenting them into the sections we considered relevant (Abstract, Results, Discussions, etc.), cleaning and standardizing the textual entities to remove irrelevant text and symbols. We retained only the general names of each compound, replacing any variations found that could affect the model's effectiveness.

Figure 4 compares the evolution of two example sentences during the cleaning process: (A) shows the extracted text in its raw form, containing decoding errors and extended ASCII symbols; (B) presents the initial uncleaned JSON format, where issues persist; and (C) illustrates the sentences after preprocessing, with standardized nomenclature and without irrelevant symbols. This process ensures that the data is structured and ready for use in the model.

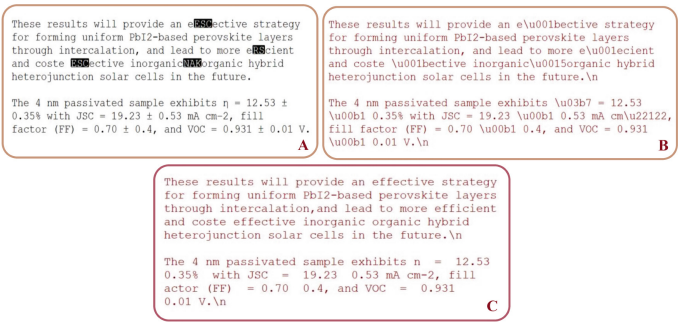


Fig. 4. Capture a photograph of the evolution of two example sentences in the cleaning process, where (A) represents the sentences in plain text (.txt), (B) shows the sentences converted to JSON format without cleaning, and (C) displays the sentences in JSON format after the cleaning process.

3.2 Extraction Using the Named Entity Recognition Model

With the data cleaned and standardized, we performed entity extraction using our model. With these results, we performed post-processing on the "Performance-Related" category entities (See Table 1), using basic Python logic (for, if) and pattern matching (re) to relate and consolidate value-type entities

with their corresponding label-type entities. For example, when a value such as "0.1 V" (entity type: *NVOC*) was found near a label of type *VOC*, both were combined into a single entity with type *VOC* and value "0.1 V". This step allowed us to transform and group the initial set of 16 entity labels into a refined set of 11 final entity labels, which were stored in the database. Figure 5 illustrates the JSON structure used to store the extracted metadata from the 584 articles processed with our SpaCy-trained model. In this format, the data is organized as a hierarchical key-value structure, where each key represents a processed PDF, each sub-key corresponds to an entity category (e.g., Synthesis Method, Performance Metric), and the corresponding values contain the extracted information from the text.


```
{
    "a0001.pdf": {
        "CAPAS": {
            "ETL": ["C60","titania","SAM","alumina","C60SAM","TiO2"],
            "HTL": ["Spiro- OMeTAD","PCPDTBT","Ma,","spiro-OMeTAD","P3HT","Li-TFSI"],
            "PTF": ["perovskite","C.-R.","CH3NH3PbI3-xClx.1","CH3NH3PbI3","CH3NH3PbI3-xClx"]
        },
        "SINTESIS": {
            "METODOS": ["thermally","spin-coated","sintered","immersing","evaporated","heated","coated",
                        "screen-printed"],
            "VARIABLES": [],
            "VALORES": ["24 h.","500 C","45 min","30 min","100 C"],
            "CONDICIONES": ["vacuum","high vacuum"]
        },
        "PARAMETROS": {
            "VOC": ["0.7 V","0.82-0.84 V","0.8 V"],
            "PCE": ["4%","3.8%","11.7%","6.7%"],
            "JSC": ["14.9 mA"],
            "FF": ["0.65-0.72"]
        },
        "ORACIONES": [
            "The architecture was also used with spiro-OMeTAD as the hole transporter to produce hybrid solar
            cells with power conversion efficiencies of 11.7% .",
            "By contrast, a titania sample coated with a layer of benzoic acid mimicking the attachment group
            on the fullerene exhibits significant electron density on the titania17 and no peak at 1030 nm.",
            "Finally, the hole transporting polymer P3HT was spin-coated on top and silver electrodes (150 nm)
            thermally evaporated under high vacuum to complete the devices.",
```


Fig. 5. Example of Metadata extracted by the SpaCy-trained model.

The JSON follows a nested tree structure: at the top level, a dictionary groups the processed PDFs. At the next level, each PDF contains extracted entities organized into predefined categories. Each category acts as a key and maps to a list of dictionaries where individual instances of the extracted entities are stored. This structure ensures that our entities associated with the synthesis methods, material properties and device performance metrics are organized correctly and facilitate their future observation and/or analysis.

3.3 Comparison

Once our database is generated in JSON format through our NER model trained with *SpaCy*, we can make comparisons with the manual database of Odabaşı (2019) [1]. We first make the automated quantitative comparison, creating a Python script that uses a loop using nested "for" statements, to compare our Entities associated with the cell structure, named, *HTL* (Hole transport layer), *ETL* (Electron transport layer) and *PTF* (Perovskite absorber layer).

For the quantitative comparison, the general names of each compound were standardized, replacing any variations, and then the semantic similarity between

the extracted entities was measured using the *SequenceMatcher().ratio()* function (which provides the similarity ratio using LCS, Longest Contiguous Matching Subsequence) from the *difflib* library, with a threshold of 0.65. Thus, only words with a similarity score equal to or greater than 0.65 were considered equivalent and, therefore, considered "successful extraction" for the comparison.

Subsequently, bar charts were generated for a qualitative comparison, to visually analyze the frequencies of each entity category and observe the similarity of the automatic extraction with the manually obtained data. Only entities considered "successful extraction" were included in the bar charts. This was done to prevent the model from appearing to extract more entities than were manually recorded, thus ensuring a balanced comparison of the frequencies of each category.

4 Results

To illustrate the entity recognition process, Fig. 6 provides a visual representation of how identified entities appear in a sample sentence extracted from an article.

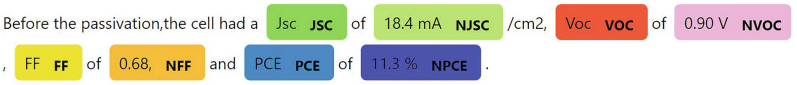


Fig. 6. Visualization of the entities identified in a sample sentence.

4.1 Model

To obtain the final model, 18 training sessions were conducted while varying the training hyper-parameters. The best-performing model for our application was trained with $N = 400$ iterations, $D = 0.5$ as the dropout rate, and $B = 100$ as the batch size.

Model Evaluation. The model was evaluated with the 190 test sentences from the Odabaşı (2019) dataset [1]. The evaluation results indicate that the model achieved **precision, recall, and F1-score** values of 93.07%, 87.38%, and 88.57%, respectively, as shown in Table 2.

Table 2. Model Evaluation

	Entities Precision		Entities Recall		Entities F1-score	
	test	evaluation	test	evaluation	test	evaluation
Count	2,597	186	2,597	186	2,597	186
mean	1	**0.9307**	1	**0.8738**	1	**0.8857**
std	0	0.1424	0	0.1942	0	0.1540

Model Validation. For the first validation method, we applied *k-Fold Cross-Validation* with $K = 5$ using the 3,088 training sentences. We computed the **average precision, recall, and F1-score** obtained during cross-validation, achieving values of 89.94%, 92.47%, and 89.69%, respectively, as shown in Table 3.

Table 3. Model k-Fold Cross-Validation

	Entities Precision	Entities Recall	Entities F1-score
Count	519.4	519.4	519.4
mean	**0.8994**	**0.9247**	**0.8969**
std	0.2080	0.1773	0.1835

For the second validation method, we compared the model's extractions with data from four articles in the Jacobsson (2022) [2] dataset, using their HTML versions. To illustrate this *qualitative comparison*, Table 4 presents the extracted entities from Savva's (2019) [35] article. Only entities that matched the database are included in the table, omitting irrelevant or incorrect terms.

Table 4. Comparison between the SpaCy-trained model and the Jacobsson (2022) [2] database, using the article *"Inverted perovskite photovoltaics using flame spray pyrolysis solution based CuAlO2/Cu-O hole-selective contact"* by Savva et al. (2019) [35]

Entities	Model Extraction	Manually Curated Database
ETL	AZO | PC70BM	PCBM-60 | AZO-np
HTL	Cu(OH)2	CuAlO2 | CuO
PTF	CH3NH3PbI3	MAPbI3
MTDS	spin coating | evaporation	Spin-coating | Evaporation
NPCE	16%	16.0%
NVOC	1.07 V	1.07 V
NJSC	19.1 mA	19.1 mA cm-2
NFF	79.6 %	79.6 %

For the *quantitative comparison*, we focused on the most relevant sections of the five articles, such as Abstract, Results and Discussion, Methods, and Experimental Section. The comparison with the database considered:

– **Exact Matches:** The extracted entity is identical to the one in the database.

- **Approximate Matches:** Similarity metrics were used to recognize variations of the same entity with minor differences. We used the *SequenceMatcher* class from the *difflib* library with a similarity threshold > 0.65.
- **Normalization:** Since material names can vary in the literature, we applied a normalization process before comparison, which included:
  - Standardizing chemical notations (e.g., MAPbI3 and CH3NH3PbI3).
  - Correcting spaces and hyphens (e.g., spiro-OMeTAD = spiro-MeOTAD).

After applying normalization and comparison, a total of 18 entities were identified in the four articles based on the database. Of these, the model correctly extracted 14, with 9 approximate matches. The final results showed a **22.22% error rate** and a **77.78% accuracy rate**.

Model Inference. The inference times obtained for each scenario were:

- A single sentence: ≈ 0.6 seconds.
- A PDF file: ≈ 3 seconds.
- An HTML file: ≈ 30 seconds.
- A set of 584 PDF files: ≈ 2 h, 30 min, 52 s.
- A set of 4 HTML files: ≈ 2 min, 5 s.

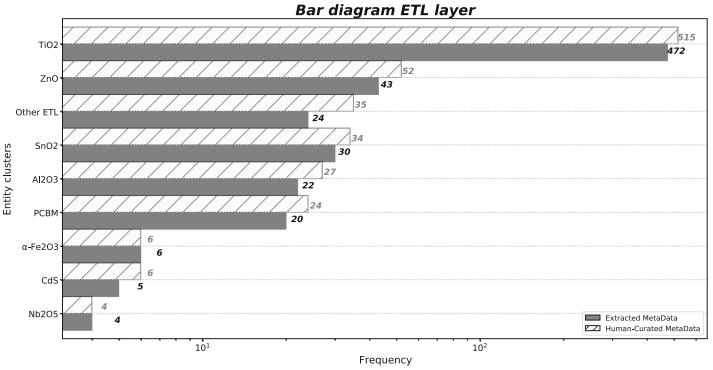
4.2 Information Extraction from PSC Articles

After completing the model evaluation and validation with satisfactory results, we proceeded to extract entities from the full set of 584 conventional perovskite solar cell articles in the Odabaşı (2019) [1] dataset, comparing the results. This analysis focuses on a quantitative and graphical comparison of the most commonly used materials for HTL, ETL, and Perovskite layers, offering a better understanding of material trends in PSC research. After applying normalization, we obtained:

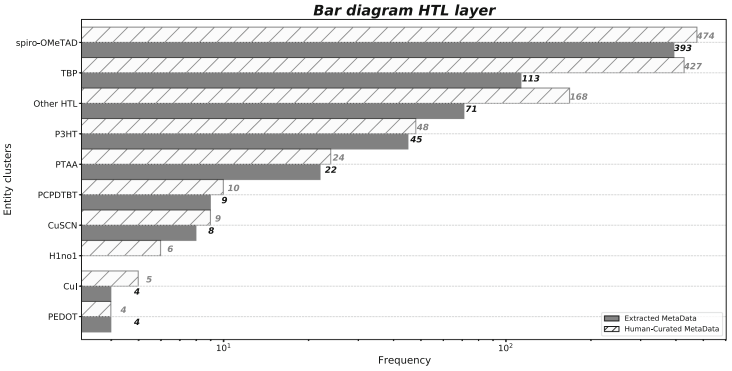
- Total entities in the database: 2,453.
- Correctly identified entities: 1,868.
- Approximate matches: 163.
- **Error rate: 23.85%.**
- **Accuracy rate: 76.15%.**

To better visualize the comparison between our model and the full Odabaşı (2019) [1] database, bar charts were generated, as shown in Fig. 7. These bar charts only consider entities that matched between both databases. Since each layer contains numerous entities, even after normalization, we aimed to reduce noise from the model's extractions by grouping similar entities into clusters. These clusters were defined based on entity frequency in the manual database and additional manual review, ensuring they accurately represented material categories for each perovskite solar cell (PSC) layer.

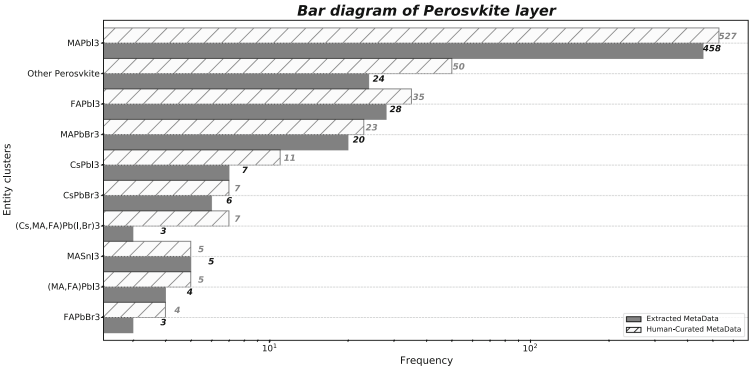
This comparison highlights the consistency of our extraction model with the global trends observed in the manual database. Figure 7 illustrate the alignment between automatic and manual extractions, showcasing trends in each layer across the analyzed scientific articles.



(a) **ETL layer**



(b) **HTL layer**



(c) **Perovskite layer**

Fig. 7. Bar charts comparing manual and automatic entity extraction for the layers from the full set of 584 papers in Odabaşı (2019) [1].

5 Discussions

Throughout the model's development, several challenges were identified that may affect its applicability and should be addressed in future work. One key challenge is the difficulty in obtaining data, as some articles that were previously open access now require a subscription or payment, limiting access to valuable scientific information. In this regard, future improvements could integrate tools that facilitate open-access verification or explore the use of alternative databases to ensure broader availability of training and evaluation data.

Another important aspect to consider is the potential bias that could have been introduced during the generation of the model's labeled training data. The model was trained with the guidance of a materials science professor (co-author); however, since it was developed by an electronic engineering researcher (co-author), some annotation decisions could reflect unintentional bias. To mitigate this, future improvements should include an increase in the labeled training dataset and additional reviews by more experts in the field to improve the model's effectiveness and understanding of scientific terminology.

Unlike other recent studies working with HTML/XML files, which seek to generate large databases and make use of large computational resources, such as *Beard et al.* [26] that uses the specialized tool for chemistry *ChemDataExtractor*, and *Xie et al.* [27] that uses large language models to infer complex structures and predict solar cell performance, our work demonstrates the feasibility of using limited computational resources to automatically extract scientific data and perform the NER task from PDF and HTML articles using the SpaCy library. Opening the possibility that our proposed method and the resulting code in this article allow its adaptation and application in different scientific domains, and the creation of automatic databases, through the generation or use of domain-specific labeled training datasets and minimal modification of certain variables of the codes used.

Future work could explore the semi-automation of document retrieval using web scraping tools, allowing for a more streamlined data acquisition process. Additionally, integrating relation extraction (RE) techniques into the pipeline would enable a more contextualized understanding of extracted information.

Another promising direction is the development of a web tool, user-friendly executable or intuitive interface, which could make this approach more accessible and suitable for use by researchers from other scientific fields and/or with limited programming knowledge. Perhaps this tool could also be incorporated into existing semi-automated annotation software, currently in internal beta, to further improve its usability and applicability.

6 Conclusions

Our proposed approach, which performs NER-focused IE and seeks to minimize the use of computational resources, enables the efficient and accurate extraction of scientific information from complex document formats, such as PDF and

HTML, commonly used in academic research. We hope that this method will contribute to improving the organization and accessibility of large volumes of research publications, helping researchers manage large sets of scientific articles more efficiently. It also reduces the issues associated with data dispersion, missing information, high costs, and long timelines associated with the manual generation of structured datasets, which are increasingly used in scientific research processes, reducing the manual workload and minimizing errors associated with manual annotation.

Acknowledgments. We sincerely thank Universidad Industrial de Santander (UIS) (uis.edu.co) (Webpage:https://uis.edu.co/es/) for the forgivable loan that funded Mary's master's studies. We also appreciate the support of the GISEL-CEMOS research groups at UIS in the development of this study, which made this article possible.

Data Availability Statement. The DOIs and/or references of the research articles used in this work are available in an Excel format at GitHub Repository (Available: https://github.com/maryz1998/NER-for-Performance-and-Synthesis-of-Perovskite-Cells-using-SpaCy.) However, due to copyright restrictions, the manually curated dataset containing entity annotations from 584 articles cannot be shared publicly. This dataset was obtained from the supplementary materials section provided in Odabaşı (2019) [1].

Additionally, a subset of four open-access articles was selected from the Jacobsson (2022) [2] database, available at Perovskitedatabase.com (Available: https://www.perovskitedatabase.com.) Their DOIs can also be accessed at the same repository: GitHub Repository

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Odabaşı, Ç., Yıldırım, R.: Performance analysis of perovskite solar cells in 2013–2018 using machine-learning tools. Nano Energy **56**, 770–791 (2019). https://doi.org/10.1016/j.nanoen.2018.11.069
2. Jacobsson, T.J., et al.: An open-access database and analysis tool for perovskite solar cells based on the FAIR data principles. Nature Energy **7**(1), 107–115 (2022). https://doi.org/10.1038/s41560-021-00941-3
3. Why is renewable energy important?. https://www.ren21.net/why-is-renewable-energy-important/. Accessed 20 Jan 2025
4. Best Research-Cell Efficiency Chart. https://www.nrel.gov/pv/cell-efficiency.html. Accessed 07 Sep 2024
5. Roose, B., et al.: Local manufacturing of perovskite solar cells, a game-changer for low-and lower-middle income countries? Energy Environ. Sci. **15**(9), 3571–3582 (2022). https://doi.org/10.1039/d2ee01343f
6. Shah, A., Meyer, E.L.: Perovskite-based solar cells in photovoltaics for commercial scalability: current progress, challenges, mitigations and future prospectus. Sol. Energy **286**, 113172 (2025). https://doi.org/10.1016/j.solener.2024.113172

7. Afre, R.A., Pugliese, D.: Perovskite solar cells: a review of the latest advances in materials, fabrication techniques, and stability enhancement strategies. Micromachines **15**, 192 (2024). https://doi.org/10.3390/mi15020192
8. Rong, Y., et al.: Challenges for commercializing perovskite solar cells. Science **361**(6408), eaat8235 (2018). https://doi.org/10.1126/science.aat8235
9. Jung, H.S., Park, N.G.: Perovskite solar cells: from materials to devices. small **11**(1), 10–25 (2015). https://doi.org/10.1002/smll.201402767
10. Takahashi, K., Takahashi, L., Miyazato, I., Tanaka, Y.: Searching for hidden perovskite materials for photovoltaic systems by combining data science and first principle calculations. ACS Photonics **5**(3), 771–775 (2018). https://doi.org/10.1021/acsphotonics.7b01479
11. Chaube, S., Khullar, P., Goverapet Srinivasan, S., Rai, B.: A statistical learning framework for accelerated bandgap prediction of inorganic compounds. J. Electron. Mater. **49**(1), 752–762 (2019). https://doi.org/10.1007/s11664-019-07779-2
12. Pilania, G., Balachandran, P.V., Kim, C., Lookman, T.: Finding new perovskite halides via machine learning. Front. Mater. **3**, 19 (2016). https://doi.org/10.3389/fmats.2016.00019
13. Yılmaz, B., Yıldırım, R.: Critical review of machine learning applications in perovskite solar research. Nano Energy **80**, 105546 (2021). https://doi.org/10.1016/j.nanoen.2020.105546
14. Yang, C., et al.: Achievements, challenges, and future prospects for industrialization of perovskite solar cells. Light Sci. Appl. **13**(1), 227 (2024). https://doi.org/10.1038/s41377-024-01461-x
15. Jiménez, M., Botero L., M., Sepulveda, A: Optimization of thin-film solar cells by using adaptive learning rate gradient descend algorithms. MRS Advances (2025). https://doi.org/10.1557/s43580-024-01109-3
16. Odabaşı, Ç., Yıldırım, R.: Assessment of reproducibility, hysteresis, and stability relations in perovskite solar cells using machine learning. Energ. Technol. **8**(12), 1901449 (2020). https://doi.org/10.1002/ente.201901449
17. Parikh, N., et al.: Is machine learning redefining the perovskite solar cells?. J. Energy Chem. **66**, 74–90 (2022). https://doi.org/10.1016/j.jechem.2021.07.020
18. Li, J., Pradhan, B., Gaur, S., Thomas, J.: Predictions and strategies learned from machine learning to develop high-performing perovskite solar cells. Adv. Energy Mater. **9**(46), 1901891 (2019). https://doi.org/10.1002/aenm.201901891
19. Vélez, J., Botero L., M.A., Sepulveda, A.: Measurement of information content of Perovskite solar cell's synthesis descriptors related to performance parameters. Emergent Mater **7**, 1961—1968 (2024). https://doi.org/10.1007/s42247-024-00667-4
20. Jurafsky, D., James H. M.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. 3rd edn.(2025). Online manuscript released January 12, 2025, stanford, USA https://web.stanford.edu/~jurafsky/slp3
21. Swain, M.C., Cole, J.M.: ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. J. Chem. Inf. Model. **56**(10), 1894–1904 (2016). https://doi.org/10.1021/acs.jcim.6b00207
22. Hawizy, L., Jessop, D.M., Adams, N., Murray-Rust, P.: ChemicalTagger: a tool for semantic text-mining in chemistry. J. Cheminformatics **3**, 1–13 (2011). https://doi.org/10.1186/1758-2946-3-17
23. Jessop, D.M., Adams, S.E., Willighagen, E.L., Hawizy, L., Murray-Rust, P.: OSCAR4: a flexible architecture for chemical text-mining. J. Cheminformatics **3**(1), 41 (2011). https://doi.org/10.1186/1758-2946-3-41

24. Olivetti, E.A., et al.: Data-driven materials research enabled by natural language processing and information extraction. Appl. Phys. Rev. **7**(4) (2020). https://doi.org/10.1063/5.0021106
25. Transformers Documentation - Hugging Face. https://huggingface.co/docs/transformers/index. Accessed 12 Nov 2024
26. Beard, E.J., Cole, J.M.: Perovskite-and dye-sensitized solar-cell device databases auto-generated using chemdataextractor. Sci. Data **9**(1), 329 (2022). https://doi.org/10.1038/s41597-022-01355-w
27. Xie, T., et al.: Creation of a structured solar cell material dataset and performance prediction using large language models. Patterns **5**(5) (2024). https://doi.org/10.1016/j.patter.2024.100955
28. Spacy. https://spacy.io. Accessed 05 Aug 2024
29. Prodigy: An annotation tool for AI, Machine Learning & NLP. https://prodi.gy/. Accessed 11 June 2024
30. Jiménez-Díaz, M. Z.: Herramienta de software para la extracción automática de parámetros de desempeño a partir de publicaciones científicas de celdas solares de perovskita. Undergraduate thesis, Universidad Industrial de Santander (2022)
31. Murphy, K.P.: Machine Learning: A Probabilistic Perspective, 1st edn. MIT press, Cambridge, London, England (2012)
32. Eisenstein, J.: Introduction to Natural Language Processing, 2nd edn. MIT Press, Cambridge, London, England (2019)
33. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011). http://jmlr.org/papers/v12/pedregosa11a.html
34. Prodigy: An annotation tool for AI, Machine Learning & NLP. https://pymupdf.readthedocs.io/en/latest/. Accessed 11 June 2024
35. Savva, A., et al.: Inverted perovskite photovoltaics using flame spray pyrolysis solution based CuAlO2/Cu–O hole-selective contact. ACS Appl. Energy Mater. **2**(3), 2276–2287 (2019). https://doi.org/10.1021/acsaem.9b00070