Information Extraction in Scientific Articles

Prunelle Daudré–Treuil, Maxime Renard, Pierre Epron,
Inés Hernández, Lisa Del Cucina, Shehenaz Hossain, Mayank Mishra

Abstract

Identifying scientific articles related to a specific feature is crucial for most researchers and students. The field of Natural Language Processing (NLP) has long been interested in this task. Different Information Extraction (IE) pipelines have been developed for this purpose. Most of them adopt a similar approach. They first extract mentions associated with specific entities such as tasks, materials, metrics, and methods. Then, they arrange them into coreference clusters. Lastly, they establish relationships between these clusters. This article aims to show some of the unresolved issues with that approach, by focusing on an already existing pipeline: SciREX. To achieve this, it was first necessary to set up their methodology, to which a few minor modifications were implemented. By analyzing the results, three structural problems were identified: (1) A difficulty in extracting materials correctly. (2) A coreference model biased by overmuch character similarity. (3) A relation extraction model that has difficulties extracting valid relationshipsin

1 Introduction

It is recognized that the number of scientific articles published each year is increasing considerably. A study conducted by the United Nations Educational, Scientific and Cultural Organization (Lewis et al., 2021) reports a 21% increase in the number of publications between 2015 and 2019. This growth is even more significant in the field of artificial intelligence (AI) with an increase of 44%. It has therefore become problematic to follow the evolution of research in a field. Extracting and visualizing relevant information within scientific articles has become a key issue. Some notable applications like Semantic Scholar[1] already provide several features in this direction. These include paper summarization with their TLDRs (Too Long; Didn't Read) models, citation classification which is specifically beneficial for their highly influential citations classes and research feeds recommendations that provide a practical way to solve the issue of publication growth. Other types of applications, such as PubMed[2] or PapersWithCode[3] (PwC) provide specialized research tools in a field. PubMed concentrates on medical and biological topics. It has the advantage of linking the article content with the medical ontology Medical Subject Headings (MeSH)[4]. PwC concentrates on the computer science domain and aims to gather code repositories, articles, datasets and evaluation tables in a unique page. A little-known but no less relevant solution is NLP-progress[5]. It is a leaderboard that aims to follow the state-of-the-art of applications in NLP. It is updated by a community of researchers in the field and has the advantage of sorting its content by language. Recently, the NLP community has started working on extracting and relating different entities in articles. Attention was placed on four types of entities: Task, Dataset, Metric and Method (TDMM). Several projects have attempted to develop a pipeline capable of accomplishing this task. The main objective behind being to use the results of the pipeline to build knowledge graphs (KG).

The primary motivation for this article was to explore the alternative possibilities offered by TDMM pipelines. It was scheduled to attempt to integrate a Language entity using the already existing Named Entity Recognition (NER) model of Schweter and Akbik (2020). By adding a Language - Material relationship, it would have been possible to automate the creation of a leaderboard like that of NLP-progress. Another objective was to seek a so-

[1] Semantic Scholar

[2] PubMED

[3] Papers With Code

[4] MeSH

[5] NLP-Progress

lution to reveal the missing relationships between the entities. For example, highlighting the fact that an adequate Method has never been used on a Task can represent a significant asset for research. To achieve these goals, finding an already existing pipeline was necessary. After some research, it was apparent that the work of Jain et al. (2020) and the SciREX model acceptably corresponded to these tasks. As other works did previously, this pipeline is divided into three distinct components: Mention Extraction (ME), Coreference Resolution (CR) and Relation Extraction (RE). The ME component consists of extracting mentions related to entities by implementing a span classifier. The CR component is used to group mentions belonging to the same entity type using binary classification probability and clustering. Lastly, the RE component classifies the relation between these entities as valid or not. While experimenting with these three components, several structural issues were detected that caused the results to be unsatisfactory. The motivation of this article has therefore become to analyze these issues to apprehend them and propose a solution. The contributions of this article can be summarized as follows:

1. Highlighting issues in mention extraction of material entities, attributed to annotational complexity.

2. Pointing towards the existence of character similarity biases present in pairwise coreference model.

3. Revealing a significant imbalance in the results of the classification model for binary relations.

2 Related work

Many researchers have experimented with automatic information extraction from scientific articles. Early on, works like Tsai et al. (2013) and Gábor et al. (2016) proposed an unsupervised approach to extract concepts and relations from scientific literature. With the SemEval 2017 (Augenstein et al., 2017) and SemEval 2018 (Gábor et al., 2018) datasets, IE on scientific papers has taken a new direction by introducing Task, Material, Process entities and their relations. Several deep learning approaches have experimented with these datasets (Ammar et al., 2017, Luan et al., 2017, Augenstein and Søgaard, 2017).

As already explained, this type of model is broadly divided into three subtasks: ME, CR and RE. Earlier work (Clark and Manning, 2016, Wiseman et al., 2016, Lee et al., 2017, Adel and Schütze, 2017) has concentrated on attempting to improve CR and has proven that it is a well-understood task. More recent work focuses on ME: Hou et al. (2021) introduces a dataset (TDMSci) annotated with Task, Dataset and Metric entities. This dataset is composed of 2,000 sentences of articles from the Association for Computational Linguistics (ACL). It has been annotated by NLP experts, and they reported an inter-annotator agreement score of 0.842 using Fleiss' k value. Several deep learning models have been experimented on this dataset. They achieved their best results using flair BiLSTM-CRF (Akbik et al., 2018) with an f-score of 0.63. Zaratiana et al. (2022) recent work demonstrates that state-of-art Hierarchical Transformer Model (HNER) outperformed other models on most scientific mention extraction datasets. This includes TDMSci on which they obtained an f-score of 0.678.

Regarding RE, it is a challenging task that many works attempt to solve by doing it jointly with the entity extraction task (Katiyar and Cardie, 2017 Zhang et al., 2017, Zheng et al., 2017, Adel and Schütze, 2017). Jia et al. (2019) proposed an attractive solution to extract relations at document-level by developing a novel technique called Paragraph Embedding. They evaluated the performance of the model on a standard biomedical dataset of drug-genemutation interactions in scientific articles. They demonstrated their models outperformed other work on this task.

Other works have approached the task as a whole. Luan et al. (2018) introduced a brand new dataset: SciERC. It is composed of 500 abstracts from 12 conferences/workshops proceedings on AI. They defined six entity types (Task, Method, Metric, Material, Other-ScientificTerm and Generic) and seven relationship types (Compare, Part-of, Conjunction-Of, Evaluate-for, Feature-of, Used-for, Hyponym-Of). They reported an Inter-Annotator Agreement (IAA) kappa score of 0.769 for entity mentions, 0.638 for coreferences and 0.678 for relations. They also designed a SciIE pipeline by implementing the three components already defined (ME, CR, RE) in a single unit. They achieved this by treating each sub-component as a classification problem to which they provided a representation of the spans computed by a BiLSTM. The SciERC dataset and

the SciIE pipeline are close to what SciREX offers and which was used for this article. The following section will present in detail the contents of the SciREX dataset.

3 SciREX Dataset

The SciREX dataset is made of 438 scientific articles sampled from PwC. It has been annotated with four different properties: Entity mentions, Coreference Relationships, Coreference cluster saliency, and N-ary relations. They report an IAA of 0.95 cohen-k score on a sample of five documents between their annotators. They did not report IAA for each annotation layer.

3.1 Entity mention

They annotated mentions for four types of entities: *Task*, *Material*, *Metrics* and *Methods*. Because the SciREX guidelines[6] do not define properly each entity type, we take the definitions from SciIE which should be similar.

Task: Applications, problems to solve, systems to construct. *E.g. information extraction, machine reading system, image segmentation, etc.*

Material: Data, datasets, resources, Corpus, Knowledge bases. *E.g. image data, speech data, stereo images, bilingual dictionary, paraphrased questions, CoNLL, Panntreebank, WordNet, Wikipedia, etc.*

Metric: Metrics, measures, or entities that can express the quality of a system/method.*E.g. F1, BLEU, Precision, Recall, ROC curve, mean reciprocal rank, mean-squared error, robustness, time complexity, etc.*

Method: Methods, models, systems to use or tools, components of a system, frameworks. *E.g. language model, CORENLP, POS parser, kernel method, etc.*

Entity mentions are stored as a list of spans for each document. A span contains the position of the start and the end token of the mention as well as the entity type of the mention. *E.g. for the text "Lorem ipsum dolor sit amet", a span (1, 3, T) represents the mention "ipsum dolor" of type T*

3.2 Coreference Resolution and Saliency

They annotated the coreference relationships between each mention to produce clusters of mentions belonging to the same entity type. They also annotated the saliency of coreference clusters, which means they decided whether a mention cluster is related to the main topic of the article. In this experiment, saliency classification has been put aside because the primary objective is not to summarize the content of an article. Coreference clusters are stored as a dictionary of key labels associated with a list of span values. Spans contain the position of the start and the end tokens of the mention but not the type of the mention. It can be easily retrieved by using the list of mentions of the same document. *E.g. For a coreference labeled C, the dictionary {C : [(1, 3), ...]} represent the coreference cluster of C*

3.3 N-ary relation

They annotated 4-ary relations between clusters of different entity types. Relations are stored in a list of dictionaries for each document. Each dictionary contains the four types of entities as keys and a coreference cluster as values. When a relation only connects three types of entities, they add a dummy cluster to complete the 4-ary relation. *E.g. For a Task A, a Material B, a Metric C and a Method D, the dictionary {'Task':'A', 'Material':'B', 'Metric':'C', 'Method':'D'} represents the 4-ary relation between them*

This experiment only works on binary relationships. To do this, each 4-ary was "unfolded" into six sub-relations between each entity pair.

3.4 Dataset partition

SciREX is already split into 3 subsets: Train, Test and Dev. Tables 1 and 2 show the distribution of layers of annotation within the different sets. It is clear that the mentions and the co-references are evenly distributed, which is excellent for training the model. However, relations have a more heterogeneous distribution, which was expected as it is unlikely to find a balance for this type of layer by taking a random sample.

4 Method

SciREX additionally provides a three-component pipeline for solving tasks. Because the primary objective of this article was not to propose a new one, the implementation used for the experiment contains few differences. They are all summarized in figure 3 where SciREX represents the baseline pipeline and SciREX+ this paper pipeline. They are also explained in the following sections.

[6] SciREX guidelines on GitHub

		Material		Method		Metric		Task	
Train	Mentions	7,454	6.9%	67,464	62.5%	10,744	9.9%	22,335	20.7%
	Coref	558	32.20 %	465	26.83 %	382	22.04 %	328	18.93 %
Dev	Mentions	1,519	6.5%	14,717	63%	2,294	9.8%	4,835	20.7%
	Coref	145	39.08 %	79	21.29 %	75	20.22 %	72	19.41 %
Test	Mentions	1,642	6.4%	16,277	63.7%	2,294	9%	5,356	20.9%
	Coref	114	31.67 %	89	24.72 %	84	23.33 %	73	20.28 %
Full	Mentions	10,615	6.8%	98,458	62.7%	15,332	9.8%	32,526	20.7%
	Coref	817	33.16 %	633	25.69 %	541	21.96 %	473	19.20 %

Table 1: Size and ratio in percent for every type of entity for the different sets

	Training	Development	Testing
Task/Material	858 16.89 %	185 18.76 %	181 18.34 %
Task/Metric	606 11.93 %	137 13.89 %	130 13.17 %
Task/Method	497 9.78 %	100 10.14 %	91 9.22 %
Material/Metric	1128 22.20 %	231 23.43 %	246 24.92 %
Material/Method	1238 24.37 %	188 19.07 %	189 19.15 %
Metric/Method	754 14.84 %	145 14.71 %	150 15.20 %

Table 2: Size and ratio in percent for every type of relation for the different sets

4.1 Mention Extraction

The SciREX solution to this task is to use a BIOUL based Conditional Random Field (CRF) Sequence tagger fed with a BERT-BiLSTM Embedding. A more advanced solution was provided by the HNER model (Zaratiana et al., 2022) which outperforms other models on two scientific benchmarks. It consists of adding a Word-level layer between the Embedding layer and the CRF layer. A word level layer takes the first subword of each word in the previous layer and encodes their interaction with a single-layer transformer (Vaswani et al., 2017). The addition of this layer provides a better representation of the sequence labeling. The HNER model was used for this experiment.

4.2 Coreference Resolution

This step could be divided into two tasks. The first one consists in calculating for each pair of mentions of the same entity type a pairwise coreference score. The second consists in clustering mentions by using their pairwise coreference scores. SciREX solution computes coreference score by training a binary classifier on each pair of mentions. They build their classification model using a feed forward linear layer fed with a BERT Embedding. They compute the classification probability by using the softmax function. This probability is used as pairwise coreference score for the pair of mentions. To perform clustering, they used an agglomerative hierarchical clustering from Ward Jr (1963) and silhouette score from Rousseeuw (1987) to determine the clusters numbers. According to their code and contrary to what they report in their paper, SciREX seems to use a default BERT embedding for the pairwise classification whereas this experiment used the SciBERT embedding.

4.3 Binary Relation Classification

This task involves generating a combination of two entities (eg, a *Task* and a *Material*) under the assumption that these are related and then, this hypothesis is tested to determine if the relation is valid or not. The solution implemented in SciREX is based on Jia et al. (2019). It consists of: For a relation $R = (C_1, C_2)$ where $C_i = m_{i1}, ..., m_{ij}$ is a cluster of mentions which belongs to the same

entity. It encodes this relation into a single vector by computing for all paragraphs $P = P_1, ...P_p$ an embedding E^pC_i and aggregates them to construct the document representation of the relation R. Paragraph embedding $E^p_{C_i}$ is done by using a Bi-LSTM layer on each paragraph and concatenating hidden states of mention by using either max-pooling or logsumexp. Relation embedding is computed by using a feed forward linear layer as a concatenation method such that $E^p_R = FFN([E^p_{C1}; E^p_{C2}])$. Finally, the classification is made by feeding the document-level representation E_R to a Feed Forward Linear Layer with a sigmoid function applied to results to compute probability. E_R is the mean of each paragraph-level representation such that $E_R = \frac{1}{|P|}\sum_{p=1}^{|P|} E^p_R$. SciREX default model used maxpooling to concatenate the hidden states of the mentions and a sigmoid function to calculate the last layer probability, but following Jia et al. (2019) recommendations, this experiment uses logsumexp for the concatenation and a softmax function for the probability. Logsumexp can be defined as: $logsumexp(x_1, ..., x_k) = log \sum_{i=1}^{k} exp(x_i)$ This is a smooth version of max-pool that better represents the weaker signals expressed by some mentions.

5 Experiment

All models were trained with a maximum of 10 epochs using the early exit algorithm from Pytorch Lightning[7] with a *patience* of 5. Dataset splits (train, dev, test) are the same as the SciREX original ones and the dev f-score was used to trigger early exit. Adam optimizer was used in order to supervise training with a starting learning rate of 2e-5. For the CR and RE models, only a part of the invalid peers is used during training. This sampling is done by calculating the probability that a pair is valid in the training set and randomly choosing invalid pairs to keep based on this probability. This part is essential to solve the imbalance issue between valid and invalid peers. The ME and CR models were trained 10 times with different random parameters to ensure their consistency. Due to high GPU memory requirements, the RE model was trained only once.

5.1 Metrics

ME: MUC Metrics from the workshop of Chinchor and Sundheim (1993) were used during the validation steps and for the final evaluation. NEREvaluate package[8] was used to compute those metrics.
CR: Common precision, recall and f-score were used for the binary classification part and SciREX custom evaluation metric for clustering.
RE: Common precision, recall, f-score.

6 Results

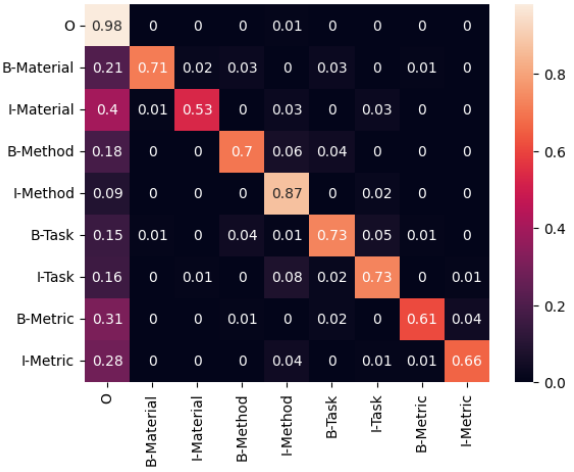

Figure 1: Confusion matrix of mention extraction model. Rows represent the gold labels and columns the predicted labels.

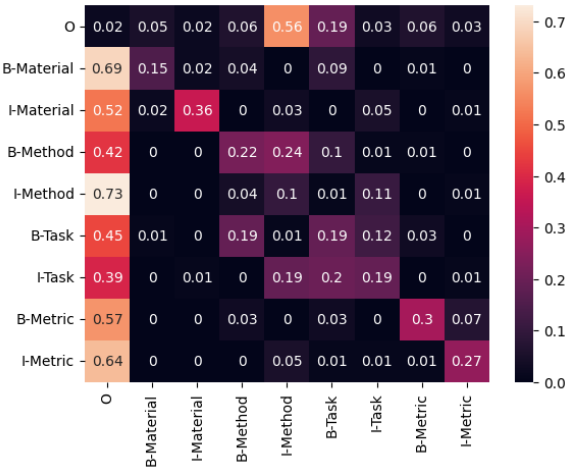

Figure 2: Confusion matrix of mention extraction model by using the second most likely label. Rows represent the gold labels and columns the predicted labels.

6.1 Mention Extraction

As Table 4 shows, SciREX+ slightly outperforms SciREX. It can be explained by the use of the

[7] Pytorch lightning

[8] nervaluate on GitHUb

ME	SciREX	SciBERT	BiLSTM	CRF tagger	
	SciREX+	SciBERT	BiLSTM	WordLvl layer	CRF Tagger
CR	SciREX	BERT	Binary Classifier	Agglomerative Hierarchical Clustering	
	SciREX+	SciBERT	Binary Classifier	Agglomerative Hierarchical Clustering	
RE	SciREX	SciBERT	Paragraph Embbeding	MaxPooling	Sigmoid
	SciREX+	SciBERT	Paragraph Embbeding	LogSumExp	Softmax

Table 3: Differences between the SciREX and the SciREX+ models

Model	**Precision**			**Recall**			**F-Score**		
	min	mean	max	min	mean	max	min	mean	max
Mention Extraction									
SciREX	-	0.707	-	-	0.717	-	-	0.712	-
Paperjam	0.705	0.717	0.728	0.726	0.736	0.745	0.715	0.726	0.738
Coreference Resolution (Pairwise)									
SciREX	-	0.861	-	-	0.852	-	-	0.856	-
SciREX+	0.942	0.946	0.947	0.943	0.947	0.951	0.942	0.946	0.950
Coreference Resolution (Clusters)									
SciREX	-	1.000	-	-	0.984	-	-	0.987	-
SciREX+	0.942	0.988	1.000	0.536	0.952	1.000	0.696	0.967	0.999
Binary Relation Extraction									
SciREX	-	0.820	-	-	0.440	-	-	0.570	-
SciREX+	-	0.657	-	-	0.698	-	-	0.670	-

Table 4: Precision, Recall, F-Score statistics are evaluated on the test set of the 10 random training runs of the Paperjam (PJ) models compared to the result of SciREX models. PJ Binary Relation Classification mean value corresponds to the only completed run. The mean value of the SciREX models corresponds to the reported results in their paper

HNER architecture. Precise results by entity can be observed in Table 5, which shows that the Method entity generally outperforms other entities. Material also has a less convincing f-score when compared to the other entity types. Because SciREX experiments do not report results by entity, it is difficult to say if the issue comes from the SciREX+ implementation. The TDMsci experiment raises the same issue with several distinct models on their Dataset entity type. Looking a little closer into SciREX annotations revealed some mistakes and inconsistencies in Material mentions as in Figure 6. There is also sometimes ambiguity in determining whether an entity is a material or a task. This frequently happens with workshops where their name can be employed to refer either to a material or a task. An example of this type of ambiguity is the word SciREX itself which can refer to the dataset or the pipeline used on the dataset. Figures 1 and 2 deliver a partial explanation of the issue. Figure 1 represents the confusion matrix of the model. It is explicit that the confusion lies between the intermediate tokens of Material (I-Material). The value of the [I-Material, O] cell is higher than the values of the other [x, O] cells, and the values of the other [I-Material, x] cells are all extremely low. Figure 2 shows the same confusion matrix but using the second most likely label according to the model. By adding the values of [I-Material, I-Material] of the two matrices, we obtain a score higher than 0.80 which indicates the model is not completely wrong.

6.2 Coreference Resolution

Differences between the performance of the two models for *Pairwise Coreference* shown in Table 4 can be explained by the different base embedding used. It should also be noticed that *False pairs* outperform a little *True pairs* with a mean f-score of 0.968 versus a mean f-score of 0.848. Because many coreference pairs are identical or practically identical words: we used the Levenshtein ratio, which measures the character similarity between two strings, to provide a more in-depth assessment of the model performance. For a given threshold (T) we kept only pairs (a, b) of the test set that

Model	Precision			Recall			F-Score		
	min	mean	max	min	mean	max	min	mean	max
Task	0.662	0.682	0.704	0.667	0.677	0.695	0.665	0.679	0.693
Material	0.478	0.484	0.492	0.552	0.561	0.567	0.511	0.520	0.526
Metric	0.721	0.733	0.748	0.653	0.660	0.678	0.680	0.694	0.705
Method	0.716	0.753	0.790	0.774	0.783	0.796	0.762	0.768	0.771

Table 5: Precision, Recall and F-Score statistics of the 10 random runs of the Paperjam model for each entity type.

had a Levenshtein ratio ($\mathcal{L}$) greater or equal to the threshold. We progressively increased the threshold (by 0.05) and plot the result as Figure 3 shows. Thus we see how the model performs when i varies over $\mathbb{N}$ for $A_i = \{(a, b) \mid \mathcal{L}[(a, b)] \geq T + 0.05i\}$. Similarly, for pairs where the Levenshtein ratio is less than the threshold value for each successive iteration, we have Figure 4. Several observations can be made: For Figure 3 even if the starting value is more than expected, True pairs f-score increase when the threshold increases. The F-score of False pairs significantly decreases when the threshold reaches 0.95. It means that the model has difficulties identifying False pairs that have a really close Levenshtein ratio. For Figure 4, we can clearly see that the model is biased by the proximity between the two words of the pair. Performance is under 0.5 until the Levenshtein ratio becomes greater than 0.95. The good results of the global model evaluation are explained by the fact that most of the true pairs have a really close Levenshtein distance (Figure 5 show that 90% of pairs have a Levenshtein distance greater than 0.95).

Coreference Clustering results (Table 4) show a significant inconsistency related to the random parameter of the model. The minimum Recall of the Paperjam model is 0.536 versus a maximum of 1.000 and a mean of 0.952. Further investigation shows that only one of the runs reaches such a low performance. Because SciREX did not process several runs (or did not report such results) it is difficult to deduce if the issue comes from SciREX+ implementation or if it is more general.

6.3 Binary Relation Classification

Major differences in results (Table 4) could be explained by the use *logsummax* or by the use of softmax to compute probability. The weighted performance does not reveal the performance imbalance between valid and invalid pairs with an F-Score of 0.8 for valid and 0.3 for invalid. A solution from Jia et al. (2019) which has not been tested on this

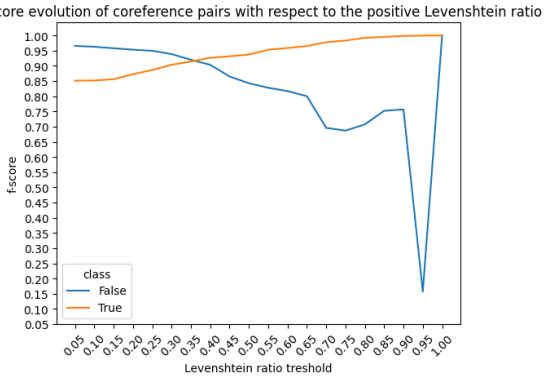

Figure 3: F-score evolution of coreference pairs with respect to the positive Levenshtein ratio. $L(a, b) > T$. Each line represents a class (True or False).

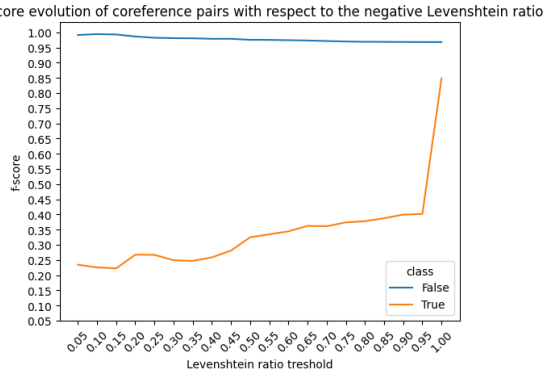

Figure 4: F-score evolution of coreference pairs with respect to the positive Levenshtein ratio. $L(a, b) < T$. Each line represents a class (True or False).

experiment should be to gather the result of different models trained on different levels of embedding such as document, paragraph or sentence.

7 Further work

Improving Dataset: This experience shows there are still problems to solve on already existing datasets like SciREX. In particular, problems extracting mentions from Material replaced one of the primary motivations for this article. Adding the Language entity and trying to link it to the Material entity could have represented a significant contribution to the field of NLP. The creation of a new

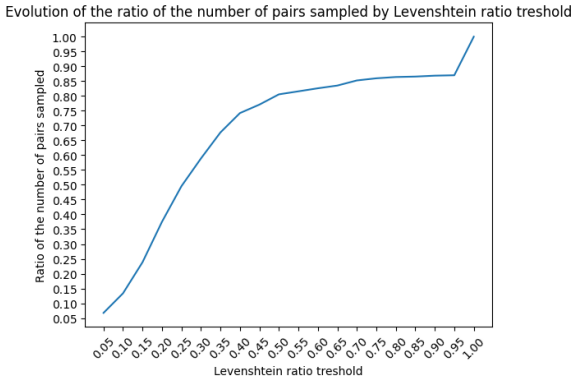

Figure 5: Ratio of the number of pair sampled for $L(a, b) > T$

dataset centered on NLP papers employing these models for a first automatic labeling is one of the solutions that we have explored. We started doing this work on a subsample of SciREX including 68 articles also present in the ACL anthology. To extract the languages, we used the NER trained on OntoNotes (Schweter and Akbik, 2020). Annotation is currently midway through, and we are still struggling to stay consistent on the material.

Binary Relation Classification: This is a challenging task and this article has failed to solve it properly. There are still improvement possibilities as showcased in the results section. There is also another type of more complex model: Neural Transition-based Model. It consists in predicting the transition sequence from an initial configuration to a terminal configuration. Relations can be derived from these transitions. Initially, this type of model was primarily implemented in POS tagging (Chen and Manning, 2014), but in recent years its scope has widened, as for example the extraction of relations between arguments (Bao et al., 2021).

Annotation Tool: To visualize and correct different results of the models we use the open source annotation tool INCEpTION[9] . As seen in Figure 7, given that there are several chains and relations and that they both go across the document, the interface is cluttered and hardly readable, making correction a difficult process. The lack of adapted tools to perform such correction slows down considerably work on complex relation tasks. Developing a tool specialized in the annotation of this kind of document could be an important contribution to the NLP community.

[9]Project INCEpTION on Github

8 Conclusion

In this paper, we presented the results and issues encountered after training several models on the SciREX dataset to perform three tasks: Mention Extraction, Coreference Resolution and Binary Relation Classification. The result analysis lead us to two observations: (1) Material entity is hard to define and after taking a closer look at the dataset we could find some annotation inconsistencies. (2) There is a character similarity bias in coreference resolution models. (3) Binary relations classification using paragraph embedding and document-level representation produces unbalanced results on valid and invalid relationships. This leads us to conclude that finding a large and robust dataset to perform a specific task is extremely complicated. Furthermore, using a non-symbolic model requires sustained involvement in the evaluation to understand the main issues of the model and solve them. In addition, we also provided an analysis of the ecological impact of the training of our models.

8.1 Ecological Impact

All training runs were performed on Grid5000 clusters which provide a tool to monitor power consumption. We recovered the energy consumption of 10 runs of the ME model, 10 runs of the CR model and 1 run of the RE model. Table 6 shows the consumption of these runs. By using this Wikipedia article[10], we calculated that the power consumption of these runs is equivalent to 84 inhabitants of France or 182 inhabitants of the world during its total training time, which is 5 days and 15 hours. Note that these runs are not the only ones we carried out. For example, after discovering a problem during the validation of the ME model, we restarted about ten executions without monitoring the energy consumption. In total, monitored runs account for two-thirds of our total Grid5000 usage. Note also that even if the RE model had been trained only one time, the results provided in the table would be multiplied by 10 to make comparison easier. This is interesting because even if this model requires a large amount of GPU memory ($>$ 32 giga) it consumes significantly less energy compared to the other models. This is easily explained by the duration of the training which is less than 2 hours, due to the smaller size of the training data. On the other hand, the CR model is trained with a large amount of data and the training duration is more 12h in average which also explains the high energy consumption.

Model	kWh
ME	13.967
CR	50.157
RE	0.010
All	64.134

Table 6: Power consumption for this experiments in kWh compared to power consumption of inhabitant in some country/region

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1723–1729, Copenhagen, Denmark. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Waleed Ammar, Matthew E. Peters, Chandra Bhagavatula, and Russell Power. 2017. The AI2 system at SemEval-2017 task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 592–596, Vancouver, Canada. Association for Computational Linguistics.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Isabelle Augenstein and Anders Søgaard. 2017. Multi-task learning of keyphrase boundary classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 341–346, Vancouver, Canada. Association for Computational Linguistics.

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A neural transition-based model for argumentation mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6354–6364, Online. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Nancy Chinchor and Beth Sundheim. 1993. MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 task 7: Semantic relation extraction and classification in

[10] List of countries by electricity consumption

scientific papers. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

Kata Gábor, Haïfa Zargayouna, Isabelle Tellier, D. Buscaldi, and Thierry Charnois. 2016. Unsupervised relation extraction in specialized corpora using sequence mining. In *IDA*.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2021. TDMSci: A specialized corpus for scientific literature entity tagging of tasks datasets and metrics. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 707–714, Online. Association for Computational Linguistics.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multiscale representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3693–3704, Minneapolis, Minnesota. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Jake Lewis, Susan Schneegans, Tiffany Straza, et al. 2021. *UNESCO Science Report: The race against time for smarter development*, volume 2021. UNESCO Publishing.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2641–2651, Copenhagen, Denmark. Association for Computational Linguistics.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition.

Chen-Tse Tsai, Gourab Kundu, and Dan Roth. 2013. Concept-based analysis of scientific literature. *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Urchade Zaratiana, Pierre Holat, Nadi Tomeh, and Thierry Charnois. 2022. Hierarchical transformer model for scientific named entity recognition. abs/2203.14710.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

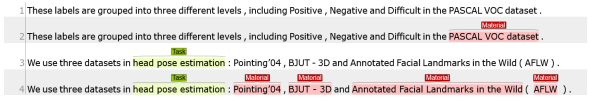

Figure 6: Mistakes in the gold annotation

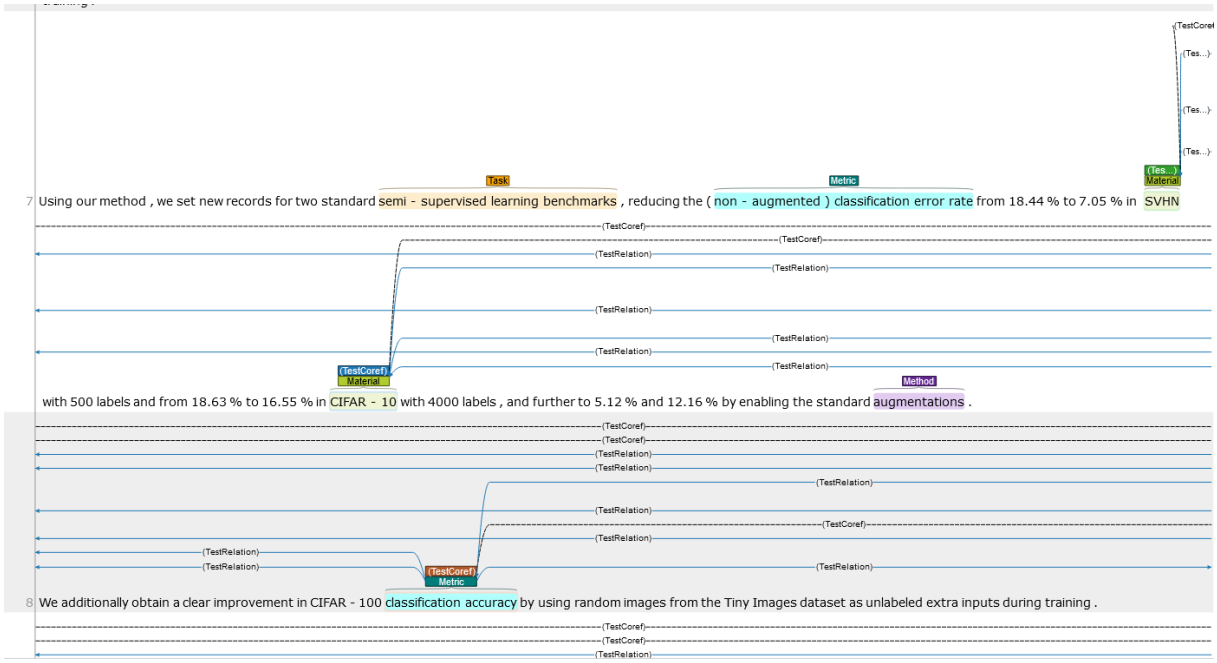

Figure 7: INCEpTION interface screenshot.