Improving the State of the Art on SciREX

Prabh Simran Singh Baweja
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
prabh@cmu.edu

Jeremy Fisher
Department of Computational Biology
Carnegie Mellon University
Pittsburgh, PA 15213
jeremyf@cmu.edu

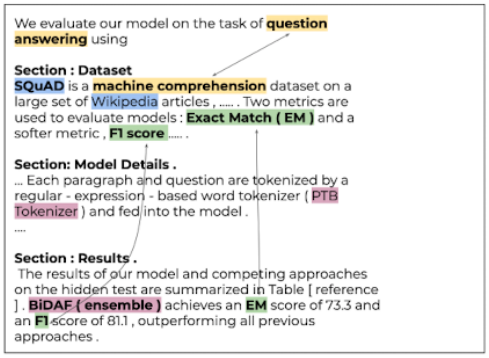
Yurun Tian
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
yurunt@andrew.cmu.edu

Abstract

SciREX is a scientific information extraction dataset. In this paper, we propose improvements to the neural baseline model distributed alongside this dataset, specifically to the modules responsible for Named Entity Recognition and Relation Extraction. We report state-of-the-art results in the Relation Extraction subtask.

1 Introduction

SciREX is a dataset built to spur development in document-level scientific information extraction. This is a challenge comprised of several subtasks: (1) **named entity recognition** (NER), which involves extracting and classifying text spans into a "method," "task," "metric," or "materials"; (2) **co-reference clustering**, which entails classifying which of these entities refer to the same concept; and (3) **4-ary relation extraction**, which refers to the extraction of tuples `(method, task, metric, materials)` that reflect the intent or application of a scientific document. See Figure 1 for an example solution to the first two subtasks, which shall be produced during inference by a model given free-text alone.



We evaluate our model on the task of **question answering** using

Section : Dataset
SQuAD is a **machine comprehension** dataset on a large set of **Wikipedia** articles , ... Two metrics are used to evaluate models : **Exact Match (EM)** and a softer metric , **F1 score** ..... .

Section: Model Details .
... Each paragraph and question are tokenized by a regular - expression - based word tokenizer (**PTB Tokenizer**) and fed into the model .
....

Section : Results .
The results of our model and competing approaches on the hidden test are summarized in Table [reference] . **BiDAF (ensemble)** achieves an **EM** score of 73.3 and an **F1** score of 81.1 , outperforming all previous approaches .

Figure 1: An annotated subsection of a document in the SciRex dataset. Blue highlight indicate a **dataset**, green indicates a **metric**, yellow indicates a **task** and pink indicates a **method**. Arrows between entities imply they reference the same concept, hence forming a co-reference.

Regarding the third subtask, such a tuple could be rendered as: "using method, the authors improved task according to the metric evaluated on a material." For example, one document contains the annotated relationship (Full-Resolution Residual Networks, Semantic Segmentation, mean IoU, Cityscapes) or, semantically: "using Full-Resolution Residual Networks, the authors improved Semantic Segmentation according to the mean IoU evaluated on the Cityscapes dataset." Note that this task can also be formulated as a binary relation extraction task by decomposing each 4-ary relation into six binary relationships; metrics for this formulation of the subtask are reported here, as well.

The authors of SciREX also provide a baseline model to produce these annotations automatically in an end-to-end, neural fashion. In this paper, we expand upon this baseline, which we henceforth refer to simply as "SciREX." In the earlier review of SciREX, we surmised that named entity recognition and relation extraction tasks were the most promising avenues of improvement.

2 Error analysis & Proposals

2.1 Addition of NER features

Named entity recognition has a macro-F1 score of only 0.712. Note that this task requires determining start- and end-positions as well as classification into a "method", "task", "metric" or "material." In an earlier error analysis, we identified that 59% of the false positives have incorrect start positions. Therefore, we sought to engineer features that would enrich the model with task-specific features.

2.1.1 Strategy: Incorporate Global Word Position

SciREX uses SciBERT as a fixed-feature extractor to embed the individual sections of a document [1]. These are concatenated and propagated into a BiLSTM layer; then, finally, a CRF layer decodes the start-/end-positions and classifies an entity-type. BERT-derived models utilize position encoding, but the global position is lost after concatenating the section of the documents [2]. This strategy explicitly engineers the feature of global position of each word in the document along with the SciBERT embeddings.

2.1.2 Strategy: Incorporate Cased Contextualized

SciREX uses the case-insensitive variant of SciBERT. We expected that the presence of a proper noun (like "New York" or "Bidirectional Encoder Representations from Transformers") would strongly suggest the presence of a named entity. In order to include this feature, we substituted the uncased version of the SciBERT tokenizer and weights with the cased version.

2.2 Modification to the Relation Extraction Decoding Algorithm

After predicting clusters of named entities, SciREX assigns a probability to each possible 4-nary relation. Note, however, it does not prescribe how to decode these probabilities during training. Rather, at the conclusion of each training epoch, a threshold (λ) is determined that maximizes the F1-score of the relation extraction task. Only if a candidate relation has a probability greater than this threshold is it reported.

This algorithms allows for violating an obvious assumption: namely, that documents comprise one or more 4-nary relations. That is to say, it is possible that no relations are reported at all. Indeed, this was the case with 35% of documents in the validation dataset.

To address this issue, we propose three strategies that enforce this document-wide constraint.

2.2.1 Strategy: Report At Least One Relation

First, the REPORT_AT_LEAST_ONE strategy is that, when no relations are otherwise reported, report the most likely relation.

2.2.2 Strategy: Report Relations Probabilistically

The `REPORT_PROBABILISTICALLY` strategy follows from the observation that the number of relations in a document can be modeled as a discrete probability distribution, like the geometric distribution. (Indeed, the maximum likelihood fit of the geometric distribution, $p = 0.404$, appears adequate, as in Figure 2.)

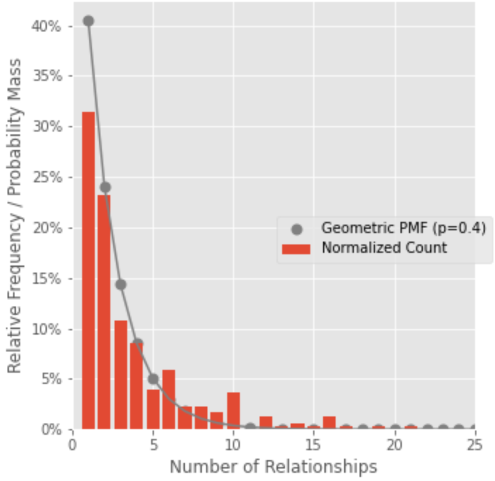


Figure 2: The distribution of the number of relations per document (red) and the geometric distribution fit using the maximum likelihood estimator (gray).

We follow the posterior regularization framework [3] by restricting the space of the model posteriors to follow the above geometric distribution. This strategy allocates probability mass to a relation distribution (that is, $\{r_1, ..., r_n\}$) according to the number of relations (i.e., its cardinality), in concert with the probability assigned to each relation by the model. In all, a given set of relations has a probability as follows:

$$\{r_1, ..., r_n\} \in \mathcal{P}(\text{Tasks} \times \text{Metrics} \times \text{Materials} \times \text{Methods}) \setminus \{\emptyset\}$$

$$P(\{r_1, ..., r_n\}) = Geometric(n \mid p = 0.404) \cdot \prod_{i=1}^{n} \text{SciRex}(r_i)$$

The single highest scoring set of relations under this probability model are each reported.

2.2.3 Strategy: Gradient-based Output Constraints

Finally, the `GRADIENT_BASED_OUTPUT_CONSTRAINTS` (or, simply, GBI) strategy trains the model in a gradient-based manner to output constrained relations directly, inspired by the paper "Gradient-based Inference for Networks with Output Constraints" [4]. This is accomplished as follows: after training SciREX normally, the weights (W_{original}) are stored and the relation extraction loss function ($f(y, \hat{y})$ = binary cross entropy) is modified. For the first modification: the cross-entropy term is multiplied by a "constraint loss", $g(\{r_1, ..., r_n\})$, which reflects the degree of violation of the constraint. Here, this is defined as the difference between the probability of the most probable relation and the threshold, λ if and only if no relation's probability falls above the threshold:

$$r_{\text{most likely}} = \max_{i=1}^{n} SciRex(r_i)$$

$$g(\{r_1, ..., r_n\}) = \begin{cases} \lambda - r_{\text{most likely}} & \text{if } r_{\text{most likely}} \leq \lambda \\ 0 & \text{otherwise} \end{cases}$$

For the second modification, a regularization term is added to minimize the difference between the stored weights and the weights as they continue to change under gradient descent. In all, the objective function for this training regime is:

$$\underset{W}{\operatorname{argmin}}\, f(y, \hat{y}) \cdot g(\hat{y}) + \alpha||W - W_{\text{original}}||_2$$

Where $\hat{y}$ is the probability for each relation under W and α is the degree of regularization.

3 Results & Analysis

3.1 NER

Baseline	Word Position	Casing	F1-score
✓			0.740
✓	✓		0.739
✓		✓	**0.741**
✓	✓	✓	0.739

Table 1: Named Entity Recognition Results by Added Features

Although adding the case feature yielded the highest F1-score for predicted named entities, this was only a marginal improvment. It is unclear as to whether this is a statistical artifact. On further analysis, we suspected that SciREX was overfitting. Visualizing the training curve, however, revealed a different intuition: that performance does not improve on the validation fold over the course of training (see Figure 3).

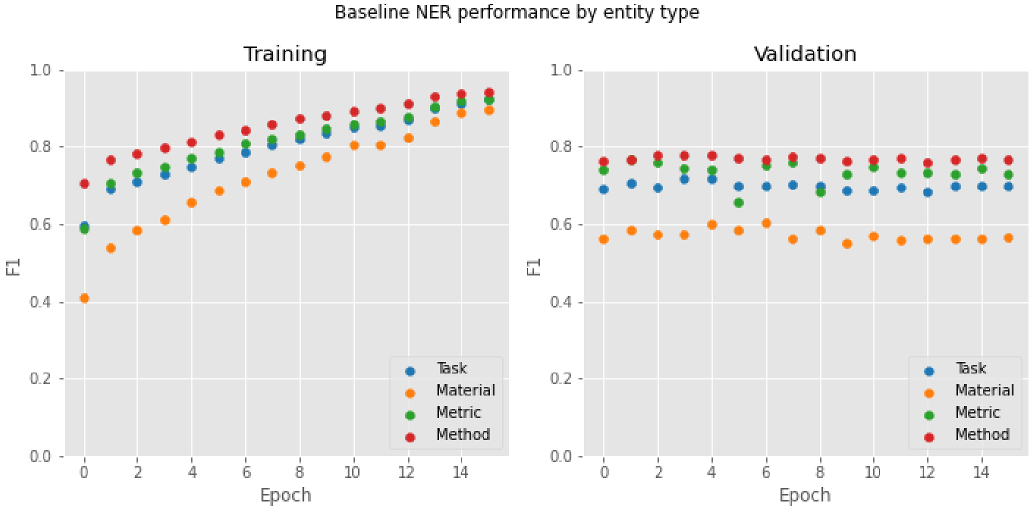


Figure 3: The training and validation F1 over training time, separated by the entity type (Task, Material, Metric or Method)

According to the validation metric, this indicates that the learned features do not generalize, despite that the model was capable of fitting relatively well to the training data. Perhaps this might be because loss is measured against a specific annotated solution, rather than against the set of reasonable solutions. Table 2 shows a manual inspection of examples that are classified as incorrect, but are nonetheless reasonable.

A similar analysis was applied to the other sub-tasks, demonstrating that generalization error does not improve over training time, with the exception of span extraction (see Figure 6).

Predicted	Annotation
Based on this observation, we convert the label of each image into a discrete label distribution , and learn the label distribution by minimizing <metric>Kullback - Leibler divergence</metric>between the predicted and ground - truth label distributions using <method>deep ConvNets</method>.	Based on this observation , we convert the label of each image into a discrete label distribution , and learn the label distribution by minimizing a Kullback - Leibler divergence between the predicted and ground - truth label distributions using <method>deep ConvNets</method>.
<method>ProNet</method> outperforms previous state-of-the-art significantly on <material>PASCAL VOC 2012</material> and <material>MS COCO datasets</material>.	<method>ProNet</method> outperforms previous state-of-the-art significantly on PASCAL <task>VOC</task> 2012 and <material>MS COCO datasets</material>.
English conversational telephone <task>LVCSR</task> System	<method>English conversational telephone LVCSR System</method>

Table 2: Predicted named entities are reasonable, although they are different from the annotation

3.2 Relation Extraction

Task	Model	P	R	$F1$
End-to-end (gold salient clustering)				
Binary Relations	Baseline	0.535	0.513	0.514
	At Least One	0.669	0.624	**0.632**
	Probabilistic	0.670	0.625	**0.633**
	GBI	0.535	0.513	0.513
4-ary Relations	Baseline	0.515	0.498	0.493
	At Least One	0.651	0.593	**0.599**
	Probabilistic	0.652	0.596	**0.601**
	GBI	0.514	0.495	0.492

Table 3: Evaluating different strategies on Relation Extraction end-to-end tasks using gold salient clusters.

Examining the Relation Extraction metrics in Table 3, we report a 12% improvement in F1-score for the binary formulation and a 10% improvement in the 4-ary formulation using the the `REPORT_PROBABILISTICALLY` strategy. Almost identical results were obtained with the `REPORT_AT_LEAST_ONE` strategy. This demonstrates it is possible to meaningfully address the issue of no relations being reported using simple post-processing strategies.

We were hopeful that the `GRADIENT_BASED_OUTPUT_CONSTRAINTS` would yield similar, if not greater, improvements. However, the performance was nearly identical to the baseline performance. We expect that this has to do with over-regularization, which may have overwhelmed the constraint loss term. We expect that tuning the amount of regularization (α) is a promising future direction to establishing a neural, end-to-end solution to constraining Relation Extraction.

3.2.1 Threshold Analysis

As mentioned previously, relation tuples must have a probability higher than a threshold, λ, to be reported as a model output. This threshold is determined by varying the threshold and maximizing the resulting F1 on the training dataset.

However, by varying the threshold manually and comparing the performance on the validation dataset, we find that a lower threshold yields a higher F1. This is non-linear: only thresholds lower than

0.8 yield the highest level of performance, and all other thresholds lead to the same, lower level of performance (see Figure 4, right). Moreover, varying the threshold does not affect the issue of relations being reported for twenty-three out of sixty-six document (see Figure 4, left).

On closer inspection, this is not an issue with decoding but with the model itself. Typically, the model produces a probability of either 0.0 or 0.96, and rarely anything in between. This indicates that the model is highly-confident, despite that it consistently produces non-sensical outputs (i.e., where there are no relations predicted).

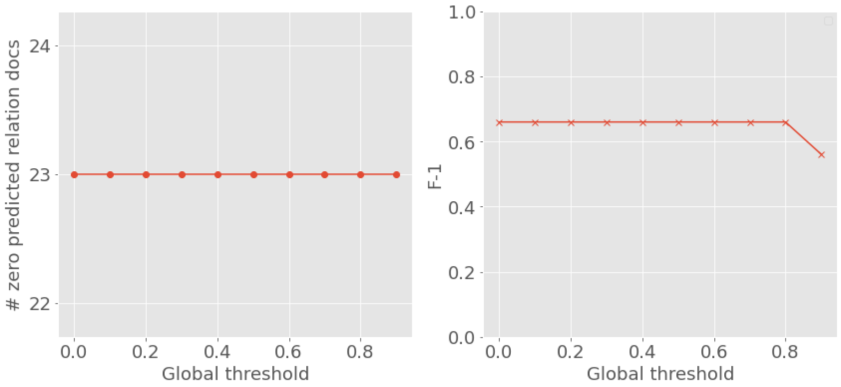


Figure 4: Adjusting global threshold does not change the number of documents with zero relation predicted (left) or affect F1 score of Relation Extraction task on 4-ary relations for threshold lower than 0.8 (right).

3.2.2 Misannotations

Finally, on inspection of individual failure cases, we find that the annotations often do not differ meaningfully from the prediction. This is analogous to our analysis of NER failure cases: although the model produces a reasonable solution, it is not the specific solution provided by the annotators.

Predicted	Annotation
{'Material': 'ImageNet', 'Metric': 'MAP', 'Task': 'Object_Detection', 'Method': 'Inception_V1', 'Confidence': 0.9124}	{'Material': 'ImageNet_Detection', 'Metric': 'MAP', 'Task': 'Object_Detection', 'Method': 'Inception_V1'}

Table 4: Failure case: Predicted relation tuple is more general but reasonable, while the annotation is more specific

4 Future work

SciREX produces reasonable outputs for most subtasks, and often a "failure" arises from the ambiguity of the task rather than the model itself. In fact, as demonstrated in Table 2 and 4, the model prediction is better or more straight-forward than the annotation. This suggests a role in using the model to foreground mistakes in the dataset, which could be rapidly reviewed by human-annotators. This is similar to a human-in-the-loop active learning approach, but applied to improving the existing dataset.

Ideally, a validation metric would consider not only the difference between the prediction and the single, human-provided annotation – but, rather, between the prediction and any valid interpretation of the relations or entities in the document. It may be possible to use SciRex to generate these many possible, albeit reasonable solutions in a scalable manner. Then, by modifying the loss function to address this multiplicity of valid outputs, the model would become even more robust and generalizable.

Another future direction involves mitigating over-confidence in Relation Extraction. There are several potential remedies. Since the most widely used neural network architecture–RELU classification networks, which is adopted by SciREX's Relation Extraction module, has demonstrated the tendency of yielding high confidence scores for data points that are far away from the training data [5]. Approximate Bayesian inference has been shown to reduce the predictive uncertainty to deal with overconfidence issue for RELU based networks[6]. Specifically, the work validates that applying the Bayesian treatment to only a few layers of the neural network provides an significant improvement to the model uncertainty problem. Besides, more general methods targeting at deep learning model uncertainty can be investigated as well. For example, [7] models the distribution of the network outputs using Dirichlet Distribution. [8] quantifies the predictive uncertainty from the perspective of ensembling and frequentist calibration.

5 Code

Our repository can be found on Github and the experiment results can be found here.

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text, 2019.

[2] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[3] Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049, 2010.

[4] Jay Yoon Lee, Michael L. Wick, Jean-Baptiste Tristan, and J. Carbonell. Enforcing constraints on outputs with unconstrained inference. *ArXiv*, abs/1707.08608, 2017.

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017.

[6] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks, 2020.

[7] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks, 2018.

[8] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[9] Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. Scirex: A challenge dataset for document-level information extraction. In *ACL*, 2020.

6 Supplement

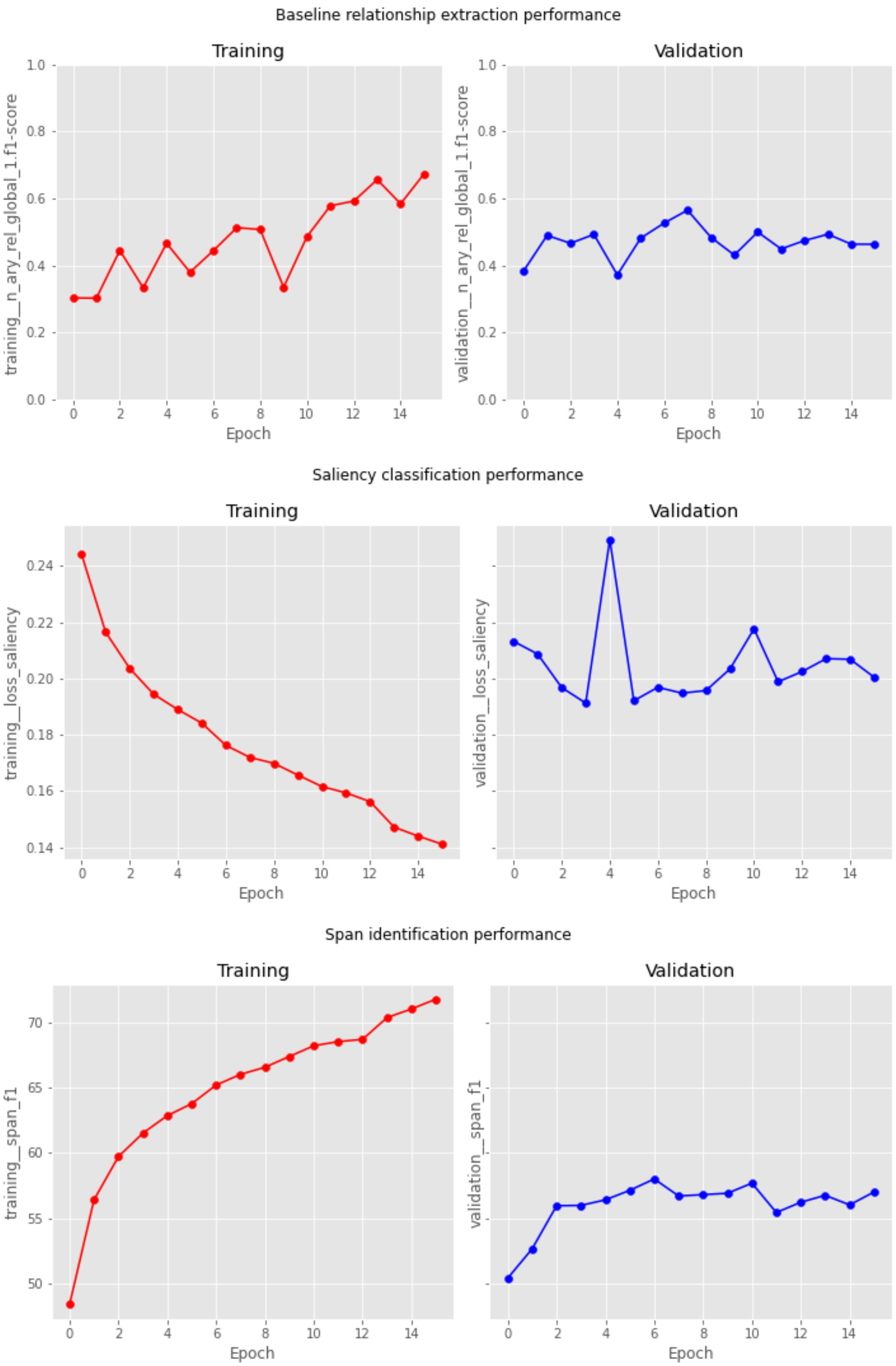


Figure 5: Training and validation performance for relation extraction, saliency classification and span identification. Notice that span identification is the only curve that demonstrates clearly learning generalizable features.)