CitationIE: Leveraging the Citation Graph for Scientific Information Extraction

Vijay Viswanathan, Graham Neubig, Pengfei Liu
Language Technologies Institute, Carnegie Mellon University
{vijayv, gneubig, pfliu3}@cs.cmu.edu

Abstract

Automatically extracting key information from scientific documents has the potential to help scientists work more efficiently and accelerate the pace of scientific progress. Prior work has considered extracting document-level entity clusters and relations end-to-end from raw scientific text, which can improve literature search and help identify methods and materials for a given problem. Despite the importance of this task, most existing works on scientific information extraction (SciIE) consider extraction solely based on the content of an individual paper, without considering the paper's place in the broader literature. In contrast to prior work, we augment our text representations by leveraging a complementary source of document context: the citation graph of referential links between citing and cited papers. On a test set of English-language scientific documents, we show that simple ways of utilizing the structure and content of the citation graph can each lead to significant gains in different scientific information extraction tasks. When these tasks are combined, we observe a sizable improvement in end-to-end information extraction over the state-of-the-art, suggesting the potential for future work along this direction. We release software tools to facilitate citation-aware SciIE development.[1]

1 Introduction

The rapid expansion in published scientific knowledge has enormous potential for good, if it can only be harnessed correctly. For example, during the first five months of the global COVID-19 pandemic, at least 11000 papers were published online about the novel disease (Hallenbeck, 2020), with each representing a potential faster end to a global pandemic and saved lives. Despite the value of this quantity of focused research, it is infeasible for the scientific community to read this many papers in a time-critical situation, and make accurate judgements to help separate signal from the noise.

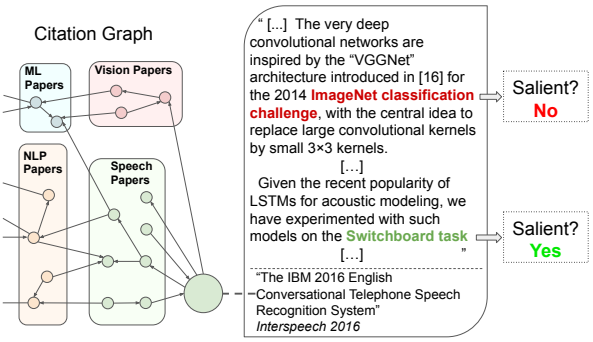

Figure 1: Example of using the citation graph to improve the task of salient entity classification (Jain et al., 2020). In this task, each entity in the document is classified as salient or not, where a *salient* entity is defined as being relevant to its paper's main ideas.

To this end, how can machines help researchers quickly identify relevant papers? One step in this direction is to automatically extract and organize scientific information (e.g. important concepts and their relations) from a collection of research articles, which could help researchers identify new methods or materials for a given task. Scientific information extraction (SciIE) (Gupta and Manning, 2011; Yogatama et al., 2011), which aims to extract structured information from scientific articles, has seen growing interest recently, as reflected in the rapid evolution of systems and datasets (Luan et al., 2018; Gábor et al., 2018; Jain et al., 2020).

Existing works on SciIE revolve around extraction solely based on the content of different parts of an individual paper, such as the abstract or conclusion (Augenstein et al., 2017; Luan et al., 2019). However, scientific papers do not exist in a vacuum — they are part of a larger ecosystem of papers, related to each other through different conceptual relations. In this paper, we claim a better under-

[1] https://github.com/viswavi/ScigraphIE

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 719–731
August 1–6, 2021.

standing of a research article relies not only on its content but also on its relations with associated works, using both the content of related papers and the paper's position in the larger citation network.

We use a concrete example to motivate how information from the citation graph helps with SciIE, considering the task of identifying key entities in a long document (known as "salient entity classification") in Figure 1.

In this example, we see a paper describing a speech recognition system (Saon et al., 2016). Focusing on two specific entities in the paper ("ImageNet classification challenge" and "Switchboard task"), we are tasked with classifying whether each is critical to the paper. This task requires reasoning about each entity in relation to the central topic of the paper, which is a daunting task for NLP considering that this paper contains over 3000 words across 11 sections. An existing state-of-the-art model (Jain et al., 2020) mistakenly predicts the *non-salient* entity "ImageNet classification challenge" as *salient* due to the limited contextual information. However, this problem is more approachable when informed of the structure of the citation graph that conveys how this paper correlates with other research works. Examining this example paper's position in the surrounding citation network suggests it is concerned with speech processing, which makes it unlikely that "ImageNet" is salient.[2]

The clear goal of incorporating inter-article information, however, is hindered by a resource challenge: existing SciIE datasets that annotate papers with rich entity and relation information fail to include their references in a fine-grained, machine-readable way. To overcome this difficulty, we build on top of an existing SciIE dataset and align it with a source of citation graph information, which finally allows us to explore citation-aware SciIE.

Architecturally, we adopt the neural multi-task model introduced by Jain et al. (2020), and establish a proof of concept by comparing simple ways of incorporating the network structure and textual content of the citation graph into this model. Experimentally, we rigorously evaluate our methods, which we call *CitationIE*, on three tasks: mention identification, salient entity classification, and document-level relation extraction. We find that leveraging citation graph information provides significant improvements in the latter two tasks, including a **10 point improvement** on F1 score for relation extraction. This leads to a sizable increase in the performance of the end-to-end CitationIE system relative to the current state-of-the-art, Jain et al. (2020). We offer qualitative analysis of why our methods may work in §5.3.

2 Document-level Scientific IE

2.1 Task Definition

We consider the task of extracting document-level relations from scientific texts.

Most work on scientific information extraction has used annotated datasets of scientific abstracts, such as those provided for SemEval 2017 and SemEval 2018 shared tasks (Augenstein et al., 2017; Gábor et al., 2018), the SciERC dataset (Luan et al., 2018), and the BioCreative V Chemical Disease Relation dataset (Wei et al., 2016).

We focus on the task of open-domain document-level relation extraction from long, full-text documents. This is in contrast to the above methods that only use paper abstracts. Our setting is also different from works that consider a fixed set of candidate relations (Hou et al., 2019; Kardas et al., 2020) or those that only consider IE tasks other than relation extraction, such as entity recognition (Verspoor et al., 2011).

We base our task definition and baseline models on the recently released SciREX dataset (Jain et al., 2020), which contains 438 annotated papers,[3] all related to machine learning research.

Each document consists of sections $D = \{S_1, \ldots, S_N\}$, where each section contains a sequence of words $S_i = \{w_{i,1}, \ldots, w_{i,N_i}\}$. Each document comes with annotations of entities, coreference clusters, cluster-level saliency labels, and 4-ary document-level relations. We break down the end-to-end information extraction process as a sequence of these four related tasks, with each task taking the output of the preceding tasks as input.

Mention Identification For each span of text within a section, this task aims to recognize if the span describes a `Task`, `Dataset`, `Method`, or `Metric` entity, if any.

Coreference This task requires clustering all entity mentions in a document such that, in each cluster, every mention refers to the same entity (Varkel and Globerson, 2020). The SciREX dataset

[2] Our proposed method actually makes correct predictions on both these samples, where the baseline model fails on both.

[3] The dataset contains 306 documents for training, 66 for validation, and 66 for testing.

includes coreference annotations for each `Task`, `Dataset`, `Method`, and `Metric` mention.

Salient Entity Classification Given a cluster of mentions corresponding to the same entity, the model must predict whether the entity is key to the work described in a paper. We follow the definition from the SciREX dataset (Jain et al., 2020), where an entity in a paper is deemed salient if it plays a role in the paper's evaluation.

Relation Extraction The ultimate task in our IE pipeline is relation extraction. We consider relations as 4-ary tuples of typed entities $(E_{\texttt{Task}}, E_{\texttt{Dataset}}, E_{\texttt{Method}}, E_{\texttt{Metric}})$, which are required to be salient entities. Given a set of candidate relations, we must determine which relations are contained in the main result of the paper.

2.2 Baseline Model

We base our work on top of the model of Jain et al. (2020), which was introduced as a strong baseline accompanying the SciREX dataset. We refer the reader to their paper for full architectural details, and briefly summarize their model here.

This multi-task model performs three of our tasks (mention identification, saliency classification, and relation extraction) in a sequence, treating coreference resolution as an external black box. While word and span representations are shared across all tasks and updated to minimize multi-task loss, the model trains each task on gold input. Figure 2 summarizes the baseline model's end-to-end architecture, and highlights the places where we propose improvements for our CitationIE model.

Feature Extraction The model extracts features from raw text in two stages. First, contextualized word embeddings are obtained for each section by running SciBERT (Beltagy et al., 2019) on that section of text (up to 512 tokens). Then, the embeddings from all words over all sections are passed through a bidirectional LSTM (Graves et al., 2005) to contextualize each word's representation with those from other sections.

Mention Identification The baseline model treats this named entity recognition task as an IOBES sequence tagging problem (Reimers and Gurevych, 2017). The tagger takes the SciBERT-BiLSTM (Beltagy et al., 2019; Graves et al., 2005) word embeddings (as shown in the Figure 2), feeds them through two feedforward networks (not shown in Figure 2), and produces tag potentials at each word. These are then passed to a CRF (Lafferty et al., 2001) which predicts discrete tags.

Span Embeddings For a given mention span, its span embedding is produced via additive attention (Bahdanau et al., 2014) over the tokens in the span.

Coreference Using an external model, pairwise coreference predictions are made for all entity mentions, forming coreference clusters.

Salient Entity Classification Saliency is a property of entity clusters, but it is first predicted at the entity mention level. Each entity mention's span embedding is simply passed through two feedforward networks, giving a binary saliency prediction.

To turn these mention-level predictions into cluster-level predictions, the predicted saliency scores are max-pooled over all mentions in a coreference cluster to give cluster-level saliency scores.

Relation Extraction The model treats relation extraction as binary classification, taking as input a set of 4 typed salient entity clusters. For each entity cluster in the relation, per-section entity cluster representations are computed by taking the set of that entity's mentions in a given section, and max-pooling over the span embeddings of these mentions. The four entity-section embeddings (one for each entity in the relation) are then concatenated and passed through a feedforward network to produce a relation-section embedding. Then, the relation-section embeddings are averaged over all sections and passed through another feedforward network which returns a binary prediction.

3 Citation-aware SciIE Dataset

Although citation network information has been shown to be effective in other tasks, few works have recently tried using it in SciIE systems. One potential reason is the lack of a suitable dataset.

Thus, as a first contribution of this paper, we address this bottleneck by constructing a SciIE dataset that is annotated with citation graph information.[4] Specifically, we combine the rich annotations of SciREX with a source of citation graph information, S2ORC (Lo et al., 2020). For each paper, S2ORC includes parsed metadata about which other papers cite this paper, which other papers are

[4]We have released code to construct this dataset: https://github.com/viswavi/ScigraphIE

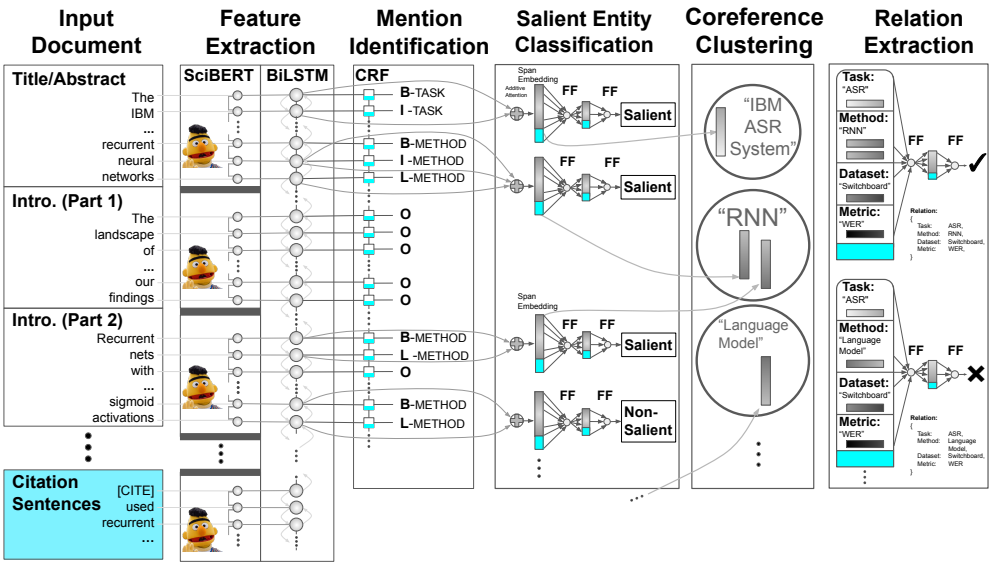

Figure 2: Architecture of the model we use for neural information extraction. Light blue blocks indicate places where we can incorporate information from the citation graph for the citation-aware CitationIE architecture.

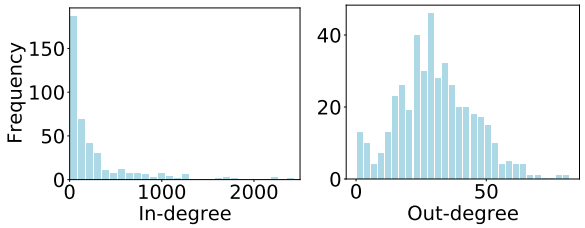

Figure 3: Degree statistics of SciREX documents in the citation graph.

cited by this paper, and locations in the body text where reference markers are embedded.

To merge SciREX with S2ORC, we link records using metadata obtained via the Semantic Scholar API:[5] paper title, DOI string, arXiv ID, and Semantic Scholar Paper ID. For each document in SciREX, we check against all 81M documents in S2ORC for exact matches on any of these identifiers, yielding S2ORC entries for 433 out of 438 documents in SciREX. The final mapping is included in our repository for the community to use. Though our work only used the SciREX dataset, our methods can be readily extended to other SciIE datasets (including those mentioned in §2.1) using our released software.

Statistics Examining the distribution of citations for all documents in the SciREX dataset (in Figure 3), we observe a long-tailed distribution of citations per paper, and a bell-shaped distribution of references per paper.

[5] https://www.semanticscholar.org/

In addition to the 5 documents we could not match to the S2ORC citation graph, 7 were incorrectly recorded as containing no references and 5 others were incorrectly recorded as having no citations. These errors are due to data issues in the S2ORC dataset, which relies on PDF parsers to extract information (Lo et al., 2020).

4 CitationIE

We now describe our citation-aware scientific IE architecture, which incorporates citation information into mention identification, salient entity classification, and relation extraction. For each task, we consider two types of citation graph information, either separately or together: (1) *structural information* from the graph network topology and (2) *textual information* from the content of citing and cited documents.

4.1 Structural Information

The structure of the citation graph can contextualize a document within the greater body of work.

Prior works in scientific information extraction have predominantly used the citation graph only to analyze the content of citing papers, such as *CiteTextRank* (Das Gollapalli and Caragea, 2014) and *Citation TF-IDF* (Caragea et al., 2014), which is described in detail in §4.2.2. However, the citation graph can be used to discover relationships between non-adjacent documents in the citation graph; prior works struggle to capture these relationships.

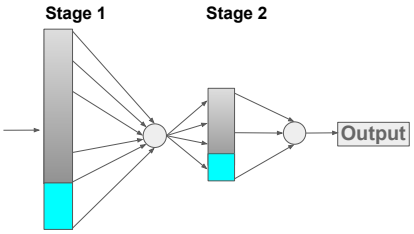

Figure 4: Feedforward architecture in each task (with CitationIE-specific parameters shown in light blue).

Arnold and Cohen (2009) are the only prior work, to our knowledge, to explicitly use the citation graph's structure for scientific IE. They predict key entities related to a paper via random walks on a combined knowledge-and-citation-graph consisting of papers and entities, without considering a document's content. This approach is simple but cannot generalize to new or unseen entities.

A rich direction of recent work has studied learned representations of networks, such as social networks (Perozzi et al., 2014) and citation graphs (Sen et al., 2008; Yang et al., 2015; Bui et al., 2018; Khosla et al., 2021). In this paper, we show citation graph embeddings can improve scientific information extraction.

Construction of Citation Graph To construct our citation graph, we found all nodes in the S2ORC citation graph within 2 undirected edges of any document in the SciREX dataset, including all edges between those documents. This process took 10 hours on one machine due to the massive size of the full S2ORC graph, resulting in a graph with ∼1.1M nodes and ∼5M edges.

Network Representation Learning We learn representations for each node (paper) using DeepWalk[6] (Perozzi et al., 2014) via the GraphVite library (Zhu et al., 2019), resulting in a 128-dimensional "graph embedding" for each document in our dataset. For each task, we incorporate the document-level graph embedding into that task's model component, by simply concatenating the document's graph embedding with the hidden state in that component. We do not update the graph embedding values during training.

Incorporating Graph Embedding Each task in our CitationIE system culminates in a pair of feedforward networks. Figure 4 describes this general architecture, though the input to these networks varies from task to task (SciBERT-BiLSTM embeddings for mention identification, span embeddings for salient entity classification, and per-section relation embeddings for relation extraction).

This architecture gives two options for where to concatenate the graph embedding into the hidden state - Stage 1 or Stage 2 - marked with a light blue block in Figure 4. Intuitively, concatenating the graph embedding in a later stage feeds it more directly into the final prediction. We find Stage 1 is superior for relation extraction, and both perform comparably for salient entity classification and mention identification. We give details on this experiment in Appendix A.3.

4.2 Textual Information

Most prior work using the citation graph for SciIE has focused on using the text of citing papers. We examine how to use two varieties of textual information related to citations.

4.2.1 Citances

Citation sentences, also known as "citances" (Nakov et al., 2004), provide an additional source of textual context about a paper. They have seen use in automatic summarization (Yasunaga et al., 2019), but not in neural information extraction.

In our work, we augment each document in our training set with its citances, treating each citance as a new section in the document. In this way, we incorporate citances into our CitationIE model through the shared text representations used by each task in our system, as shown in Figure 5. If our document has many citations, we randomly sample 25 to use. For each citing document, we select citances centered on the sentence containing the first reference marker pointing to our document of interest, and include the subsequent and consequent sentences if they are both in the same section.

We ensure the mention identification step does not predict entities in citance sections, which would lead to false positive entities in downstream tasks.

4.2.2 Citation TF-IDF

Citation TF-IDF (Caragea et al., 2014), is a feature representing the TF-IDF value (Jones, 1972) of a given token in its document's citances. We consider a variant of this feature: for each token in a document, we compute the TF-IDF of that token in each citance of the document, and average the per-citance TF-IDF values over all citances. We imple-

[6] An empirical comparison by Khosla et al. (2021) found DeepWalk to be quite competitive on two citation graph node classification datasets, despite its speed and simplicity.

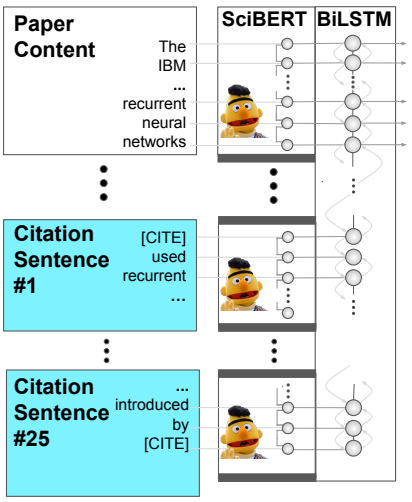

Figure 5: Incorporating citances into the text representation extractor.

mented this feature only for saliency classification, as it explicitly reasons about the significance of a token in citing texts. As a local token-level feature, it also does not apply naturally to relation extraction, which operates on entire clusters of spans.

4.3 Graph Structure and Text Content

We lastly consider using graph embeddings and citances together in a single model for each task. We do this naively by including citances with the document's input text when first computing shared text features, and then concatenating graph embeddings into downstream task-specific components.

5 Experiments

5.1 Metrics, Baselines and Training

5.1.1 Metrics

The ultimate product of our work is an end-to-end document-level relation extraction system, but we also measure each component of our system in isolation, giving end-to-end and per-task metrics. All metrics, except where stated otherwise, are the same as described by Jain et al. (2020).

Mention Identification We evaluate mention identification with the average F1 score of classifying entities of each span type.

Salient Entity Classification Similar to Jain et al. (2020) we evaluate this task at the mention level and cluster level. We evaluate both metrics on gold standard entity recognition inputs.

Relation Extraction This is the ultimate task in our pipeline. We use its output and metrics to evaluate the end-to-end system, but also evaluate relation extraction separately from upstream components to isolate its performance. We specifically consider two types of metrics:
(1) *Document-level*: For each document, given a set of ground truth 4-ary relations, we evaluate a set of predicted 4-ary relations as a sequence of binary predictions (where a matching relation is a true positive). We then compute precision, recall, and F1 scores for each document, and average each over all documents. We refer to this metric as the "*document-level*" relation metric. To compare with Jain et al. (2020), this is the primary metric to measure the full system.
(2) *Corpus-level*: When evaluating the relation extraction component in isolation, we are also able to use a more standard "*corpus-level*" binary classification evaluation, where each candidate relation from each document is treated as a separate sample.

We also run both these metrics on a binary relation extraction setup, by flattening each set of 4-ary relations into a set of binary relations and evaluating these predictions as an intermediate metric.

5.1.2 Baselines

For each task, we compare against Jain et al. (2020), whose architecture our system is built on. No other model to our knowledge performs all the tasks we consider on full documents. For the 4-ary relation extraction task, we also compare against the DocTAET model (Hou et al., 2019), which is considered as state-of-the-art for full-text scientific relation extraction (Jain et al., 2020; Hou et al., 2019).

Significance To improve the rigor of our evaluation, we run significance tests for each of our proposed methods against its associated baseline, via paired bootstrap sampling (Koehn, 2004). In experiments where we trained multiple models with different seeds, we perform a hierarchical bootstrap procedure where we first sample a seed for each model and then sample a randomized test set.

5.1.3 Training Details

We build our proposed CitationIE methods on top of the SciREX repository[7] (Jain et al., 2020) in the AllenNLP framework (Gardner et al., 2018).

For each task, we first train that component in isolation from the rest of the system to minimize

[7] https://github.com/allenai/SciREX

Model	F1	P	R
Salient Mention Evaluation			
Baseline (reported)	57.9	57.5	58.4
Baseline (reimpl.)	57.5	50.5	66.8
CitationIE			
w/ Citation-TF-IDF	57.1	50.2	66.1
w/ Citances	58.7†	51.4	**68.5**†
w/ Graph Embeddings	**59.2**†	**53.5**†	66.3
w/ Graph + Citance	58.4†	51.3	67.8†
Salient Entity Cluster Evaluation			
Baseline (reimpl.)	39.1	28.5	**75.8**
CitationIE			
w/ Citation-TF-IDF	38.6	28.4	74.3
w/ Citances	38.7	28.2	74.8
w/ Graph Embeddings	**40.3**	**29.8**	74.5

Table 1: Salient entity classification results. Baseline (Jain et al., 2020) and Graph Embedding model evaluations are each trained with 3 different model seeds, then metrics averaged; rest are from single model due to computational limitations. † indicates significance at 95% confidence. Best model is in bold for each metric.

the task-specific loss. We then take the best performing modifications and use them to train end-to-end IE models to minimize the sum of losses from all tasks. We train each model on a single GPU with batch size 4 for up to 20 epochs. We include detailed training configuration information in Appendix A.1.

For saliency classification and relation extraction, we trained the baseline and the strongest proposed models three times,[8] to improve reliability of our results. For mention identification, we did not retrain models, as the first set of results strongly suggested our proposed methods were not helpful.

5.2 Quantitative Results

Mention Identification For mention identification, we observe no major performance difference from using citation graphs, and include full results in Appendix A.2.

Salient Entity Classification Table 1 shows the results of our CitationIE methods. We observe:
(1) Using citation graph embeddings significantly improves the system with respect to the salient mention metric.
(2) Graph embeddings do not improve cluster evaluation significantly (at 95%) due to the small test size[10] (66 samples) and inter-model variation.
(3) Incorporating graph embeddings and citances simultaneously is no better than using either.
(4) Our reimplemented baseline differs from the results reported by Jain et al. (2020) despite using their published code to train their model. This may be because we use a batch size of 4 (due to compute limits) while they reported a batch size of 50.

Relation Extraction Table 2 shows that using graph embeddings here gives an 11.5 point improvement in document-level F1 over the reported baseline,[11] and statistically significant gains on both corpus-level F1 metrics.

Despite seemingly large gains on the document-level F1 metric, these are not statistically significant due to significant inter-model variability and small test set size, despite the graph embedding model performing best at every seed we tried.

End-to-End Model From Table 3, we observe:
(1) Using graph embeddings appears to have a positive effect on the main task of 4-ary relation extraction. However, these gains are not statistically significant ($p = 0.235$) despite our proposed method outperforming the baseline at every seed, for the same reasons as mentioned above.
(2) On binary relation evaluation, we observe smaller improvements which had a lower p-value ($p = 0.099$) due to lower inter-model variation.
(3) Using citances instead of graph embeddings still appears to outperform the baseline (though by a smaller margin than the graph embeddings).

5.3 Analysis

We analyzed our experimental results, guided by the following four questions:

Do papers with few citations benefit from citation graph information? Our test set only contains two documents with zero citations, so we cannot characterize performance on such documents. However, Figure 6 shows that the gains provided by the proposed CitationIE model with graph embeddings counterintuitively shrink as the number of citations of a paper increases. We also observe

[8] See Appendix A.1 for exact seeds used

[9] Reported as "Component-wise Binary and 4-ary Relations" in Jain et al. (2020)

[10] The limited size of this test set is an area of concern when using the SciREX dataset, and improving statistical power in SciIE evaluation is a crucial area for future work.

[11] The large gap between reimplemented and reported baselines is likely due to our reproduced results averaging over 3 random seeds. When using the same seed used by Jain et al. (2020), the baseline's document-level test F1 score is almost 20 points better than with two other random seeds.

Model	F1	P	R	F1	P	R
	4-ary Relation Extraction					
	Document-Level Metric			Corpus-Level Metric		
Baseline (reported)[9]	57.0	82.0	44.0	N/A	N/A	N/A
Baseline (reimpl.)	49.8	50.1	50.1	48.0	48.1	48.2
DocTAET	65.5	62.4	**85.1**	39.9	55.7	56.8
CitationIE						
w/ Citances	**69.2**	**70.0**	76.6	39.4	39.9	41.9
w/ Graph Embeddings	68.5	67.5	76.2	**58.7**†	**61.0**†	**59.6**
w/ Graph + Citance	67.5	66.8	75.0	51.9	54.6	54.5
	Binary Relation Extraction					
	Document-Level Metric			Corpus-Level Metric		
Baseline (reported)	61.1	53.1	71.8	N/A	N/A	N/A
Baseline (reimpl.)	50.8	51.1	51.1	41.2	48.4	44.6
CitationIE						
w/ Citances	69.2	69.2	**71.3**	43.3	46.7	44.0
w/ Graph Embeddings	**72.9**	**70.4**	56.1	**51.0**†	**54.1**†	**57.1**
w/ Graph + Citance	66.2	65.9	68.1	48.0†	51.4	52.7

Table 2: Comparing methods on relation extraction. Baseline, Graph Embedding, and Graph + Citance models were evaluated over 3 model seeds, and the remainder with a single seed. We use Macro-F1 for corpus-level evaluation. † indicates significance at 95% confidence, and best implemented model in each metric is bolded. Graph embeddings significantly improve over baseline on 4-ary and binary corpus-level F1 ($p < 0.05$), but are less significant on document-level F1 metrics ($p \approx 0.11$).

Model	F1	P	R
	4-ary Relation Extraction		
Baseline (reported)	0.8	0.7	17.3
Baseline (reimpl.)	0.44	0.23	**22.66**
CitationIE			
w/ Graph Embeddings	**1.48**	1.31	20.04
w/ Citances	0.75	**7.03**	13.36
	Binary Relation Extraction		
Baseline (reported)	9.6	6.5	41.1
Baseline (reimpl.)	6.48	4.09	43.83
CitationIE			
w/ Graph Embeddings	**7.70**	**5.42**	37.17
w/ Citances	7.61	4.97	**43.57**

Table 3: End-to-end model evaluation. Each model was evaluated over 3 model seeds.

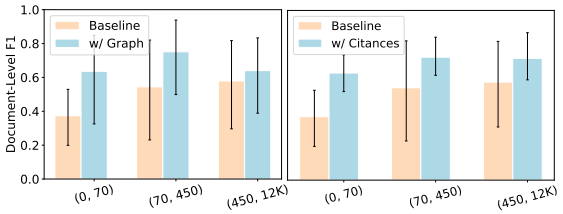

Figure 6: Document-level relation extraction F1 score of CitationIE models with graph embeddings (left) and citances (right), compared with the baseline (red) on documents grouped by number of citations.

this with citances, to a lesser extent. This suggests more work needs to be done to represent citation graph nodes with many edges.

How does citation graph information help relation extraction? With relation extraction, we found citation graph information provides strongest gains when classifying relations between distant entities in a document, seen in Figure 7. For each relation in the test set, we computed the average distance between pairs of entity mentions in that relation, normalized by total document length. We find models with graph embeddings or citances perform markedly better when these relations span large swaths of text. This is particularly useful since neural models still struggle to model long-range dependencies effectively (Brown et al., 2020).

Does citation graph information help contextualize important terms? Going back to our motivating example of a speech paper referring to ImageNet in passing §1, we hypothesized that adding context from citations helps deal with terms that are important in general, but not for a given document.

To measure this, we grouped all entities in our test dataset by their "global saliency rate" measured on the test set: given a span, what is the probability that this span is salient in any given occurrence?

In Figure 8, we observe that most of the improvement from graph embeddings and citances comes at terms which are labeled as salient in at least 20%

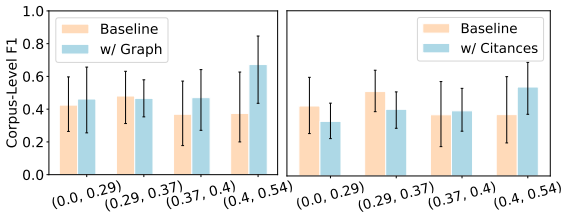

Figure 7: Corpus-Level F1 of relation extraction models, bucketed by the average distance between entity mentions in each relation.

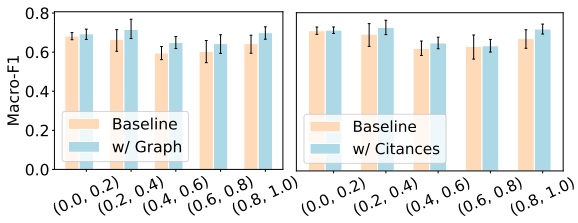

Figure 8: Macro F1 of salient mention classification models, evaluated on test-set spans, each bucketed by their training-set global saliency rate.

of their training-set mentions. This suggests that citation graph information yields improvements with reasoning about important terms, without negatively interfering with less-important terms.

6 Implications and Future Directions

We explore the use of citation graph information in neural scientific information extraction with *CitationIE*, a model that can leverage either the structure of the citation graph or the content of citing or cited documents. We find that this information, combined with document text, leads to particularly strong improvements for salient entity classification and relation extraction, and provides an increase in end-to-end IE system performance over a strong baseline.

Our proposed methods reflect some of the simplest ways of incorporating citation graph information into a neural SciIE system. As such, these results can be considered a proof of concept. In the future we will explore ways to extract richer information from the graph using more sophisticated techniques, hopefully better capturing the interplay between citation graph structure and content. Finally, we evaluated our proof of concept here on a single dataset in the machine learning domain. While our methods are not domain-specific, verifying that these methods generalize to other scientific domains is important future work.

Acknowledgments

The authors thank Sarthak Jain for assisting with reproducing baseline results, Bharadwaj Ramachandran for giving advice on figures, and Siddhant Arora and Rishabh Joshi for providing suggestions on the paper. The authors also thank the anonymous reviewers for their helpful comments. This work was supported by the Air Force Research Laboratory under agreement number FA8750-19-2-0200. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

References

Andrew O. Arnold and William W. Cohen. 2009. Information extraction as link prediction: Using curated citation networks to improve gene detection. *International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, 5682:541–550.

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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ICLR*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP/IJCNLP*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Thang D. Bui, Sujith Ravi, and Vivek Ramavajjala. 2018. Neural graph learning: Training neural networks using graphs. *Proceedings of the Eleventh*

ACM International Conference on Web Search and Data Mining.

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1435–1446, Doha, Qatar. Association for Computational Linguistics.

Sujatha Das Gollapalli and Cornelia Caragea. 2014. Extracting keyphrases from research papers using citation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).

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

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ICANN'05, page 799–804, Berlin, Heidelberg. Springer-Verlag.

Sonal Gupta and Christopher Manning. 2011. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1–9, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Ken Hallenbeck. 2020. The covid-19 deluge: Is it time for a new model of data disclosure? *ASBMB Today: The Member Magazine of the American Society for Biochemistry and Molecular Biology*.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2019. Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5203–5213, Florence, Italy. 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.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*.

Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. 2020. AxCell: Automatic extraction of results from machine learning papers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594, Online. Association for Computational Linguistics.

Megha Khosla, Vinay Setty, and Avishek Anand. 2021. A comparative study for unsupervised network representation learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(5):1807–1818.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

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, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. 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 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Preslav I. Nakov, Ariel S. Schwartz, and Marti A. Hearst. 2004. Citances: Citation sentences for se-

mantic analysis of bioscience text. In *In Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics*.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *ArXiv*, abs/1707.06799.

George Saon, Tom Sercu, Steven Rennie, and Hong-Kwang J. Kuo. 2016. The IBM 2016 english conversational telephone speech recognition system. In *INTERSPEECH*.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine*, 29:93–106.

Yuval Varkel and Amir Globerson. 2020. Pre-training mention representations in coreference models. In *EMNLP*.

Karin M. Verspoor, K. Cohen, Arrick Lanfranchi, C. Warner, Helen L. Johnson, Christophe Roeder, Jinho D. Choi, Christopher S. Funk, Yuriy Malenkiy, Miriam Eckert, Nianwen Xue, W. Baumgartner, M. Bada, Martha Palmer, and L. Hunter. 2011. A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC Bioinformatics*, 13:207 – 207.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Marringly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database : the journal of biological databases and curation*.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network representation learning with rich text information. In *IJCAI*.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7386–7393.

Dani Yogatama, Michael Heilman, Brendan O'Connor, Chris Dyer, Bryan R. Routledge, and Noah A. Smith. 2011. Predicting a scientific community's response to an article. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 594–604, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Zhaocheng Zhu, Shizhen Xu, Meng Qu, and Jian Tang. 2019. Graphvite: A high-performance cpu-gpu hybrid system for node embedding. In *The World Wide Web Conference*, pages 2494–2504. ACM.

A Appendices

A.1 Training Configurations

We train each model on a single 11GB NVIDIA GeForce RTX 2080 Ti GPU with a batch size of 4. We train for up to 20 epochs, and set the `patience` parameter in AllenNLP to 10; if the validation metric does not improve for 10 consecutive epochs, we stop training early. For each task-specific model, we use a product of validation loss and corpus-level binary F1 score on the validation set as the validation metric. For salient entity classification and relation extraction, we choose the best threshold on the validation set using F1 score.

In total, training with these configurations takes roughly 2 hours for salient entity classification, 8 hours for mention identification, 18-24 hours for relation extraction, and 24-30 hours for the end-to-end system. Our CitationIE models took roughly as long to train as the baseline SciREX models did.

For models that we trained three different times, we use different seeds for each software library:

- For PyTorch, we use seeds 133,[12] 11, and 22
- For Numpy, we use seeds 1337, 111, and 222
- For Python's `random` library, we use seeds 11370, 1111, and 2222

A.2 Mention Identification Results

Model	F1	P	R
	Mention Identification		
Baseline (reported)[13]	70.7	71.7	71.2
Baseline (reimpl.)	**74.6**†	73.7	**75.6**†
w/ Citances	74.0	73.0	75.0
w/ Graph Embeddings	74.4	**74.4**†	74.3
w/ Graph + Citance	73.6	73.0	74.3

Table 4: Mention Identification Results. † indicates significance at 95% confidence. Best model is in bold for each metric.

We include results from using citation graph information for the mention identification task in Table 4. We observe no major improvements in this task. Intuitively, recognizing a named entity in a document may not require global context about the document (e.g. "LSTM" almost always refers to a `Method`, regardless of the paper where it is used), so the lack of gains in this task is unsurprising.

[12] 133/1337/13370 is the default seed setting in AllenNLP.

A.3 Combining Graph Embeddings with Word Embeddings

Each of our task-specific components in the CitationIE model contains two feedforward networks where we may concatenate graph embedding information. We refer to these two options for where to fuse graph embedding information as "early fusion" and "late fusion", illustrated in Figure 4.

Here we show a detailed comparison of early fusion vs late fusion models on Mention Identification (Table 5), Salient Entity Classification (Table 6), and Relation Extraction (Table 7). Based on these results, we used early fusion in our final CitationIE models for mention identification and relation extraction. For saliency classification, the relative performance of early fusion and late fusion differed across our two metrics, making this inconclusive. We used early fusion for saliency classification in the end-to-end model due to strong empirical performance there.

Model	F1	P	R
	Mention Identification		
Graph Embed. (early fusion)	**74.4**†	**74.4**†	74.3
Graph Embed. (late fusion)	74.1	73.1	**75.1**†

Table 5: Comparing CitationIE models for mention identification with early graph embedding fusion vs late fusion. Results are shown from single-model evaluation. † indicates significance at 95% confidence. Best model is in bold for each metric.

Model	F1	P	R
Salient Mention Evaluation			
Graph Embed. (early fusion)	57.1	**54.4**†	60.1
Graph Embed. (late fusion)	**59.2**†	53.5	**66.3**†
Salient Entity Cluster Evaluation			
Graph Embed. (early fusion)	**43.3**†	**33.8**†	72.0
Graph Embed. (late fusion)	40.3	29.8	**74.5**†

Table 6: Comparing CitationIE models for salient entity classification with early graph embedding fusion vs late fusion. The early fusion model was trained once, while late fusion numbers are reported over an average of 3 runs. † indicates significance at 95% confidence. Best model is in bold for each metric.

Model	F1	P	R	F1	P	R
	4-ary Relation Extraction					
	Document-Level Metrics			Corpus-Level Metrics		
Graph Embeddings (early fusion)	**68.5**	**67.5**	**76.2**	58.7	61.0	59.6
Graph Embeddings (late fusion)	63.3	61.8	67.3	**75.8**†	**76.0**†	**76.1**†
	Binary Relation Extraction					
	Document-Level Metrics			Corpus-Level Metrics		
Graph Embeddings (early Fusion)	**72.9**	**70.4**	56.1	51.0	54.1	57.1
Graph Embeddings (late fusion)	58.3	58.0	**59.0**	**53.6**	**58.1**†	**66.4**

Table 7: Comparing CitationIE models for relation extraction with early graph embedding fusion vs late fusion. Early fusion models were trained 3 times, late fusion was trained once. † indicates significance at 95% confidence, and the best model in each metric is bolded.