

ReSel: N -ary Relation Extraction from Scientific Text and Tables by Learning to Retrieve and Select

Yuchen Zhuang¹, Yinghao Li¹, Jerry Junyang Cheung¹, Yue Yu¹,
Yingjun Mou¹, Xiang Chen², Le Song^{3,4}, Chao Zhang¹

¹Georgia Institute of Technology, Atlanta, USA

²Adobe Research, San Jose, USA ³MBZUAI, Abu Dhabi, UAE ⁴BioMap, Beijing, China

¹{yczhuang, yinghaoli, jzhang3027, yueyu, ymou32, chaozhang}@gatech.edu

²xiangche@adobe.com ³le.song@mbzuai.ac.ae

Abstract

We study the problem of extracting N -ary relation tuples from scientific articles. This task is challenging because the target knowledge tuples can reside in multiple parts and modalities of the document. Our proposed method RESEL decomposes this task into a two-stage procedure that first *retrieves* the most relevant paragraph/table and then *selects* the target entity from the retrieved component. For the high-level retrieval stage, RESEL designs a simple and effective feature set, which captures multi-level lexical and semantic similarities between the query and components. For the low-level selection stage, RESEL designs a cross-modal entity correlation graph along with a multi-view architecture, which models both semantic and document-structural relations between entities. Our experiments on three scientific information extraction datasets show that RESEL outperforms state-of-the-art baselines significantly. ¹

1 Introduction

Scientific information extraction (SciIE) (Augenstein et al., 2017; Luan et al., 2018; Jiang et al., 2019), the task of extracting scientific concepts along with their relations from scientific literature corpora, is important for researchers to keep abreast of latest scientific advances. A key subtask of SciIE is the N -ary relation extraction problem (Jia et al., 2019; Jain et al., 2020), which aims to extract the relations of different entities as N -ary knowledge tuples. This problem is challenging because the entities of the knowledge tuples often reside in multiple sections (*e.g.*, abstracts, experiments) and modalities (*e.g.*, paragraphs, tables, figures) of the document. Effective scientific N -ary relation extraction requires not only understanding the semantics of different modalities, but also performing document-level inference based on interleaving sig-

¹Our code is available on <https://github.com/night-chen/ReSel>.

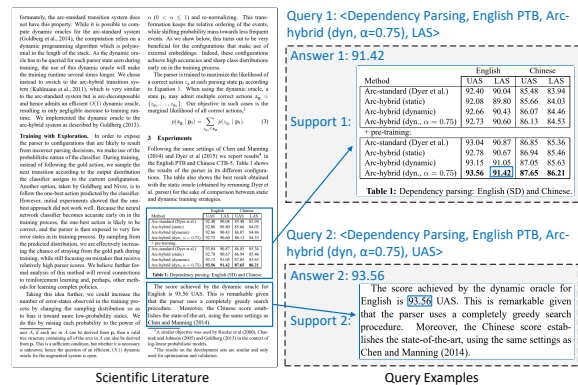


Figure 1: Illustration of the multi-modal scientific N -ary relation extraction problem on the SciREX dataset.

nals such as co-occurrences, co-references, and structural relations, as shown in Figure 1.

Document-level N -ary relation extraction has been studied in literature (Jia et al., 2019; Jain et al., 2020; Viswanathan et al., 2021; Liu et al., 2021). Some works (Zeng et al., 2020; Tu et al., 2019) use graph-based approaches to model long-distance relations in the document with the focus on text only. However, for scientific articles, an equally if not more important data structure is the table, as scientific results are often reported in tables and then referred to and discussed in text. There are also works that pre-train large-scale transformer models on massive table and text pairs (Yin et al., 2020; Herzig et al., 2020). These methods are designed for question answering, which are strong at retrieving answers that semantically match the query but fall short in inferring fine-grained entity-level N -ary relations. Besides, to perform well on SciIE, they usually require large task-specific data to fine-tune the pre-trained model, especially for long documents that contain many candidates. But in practice, such large-scale annotation data can be expensive and labor-intensive to curate. Therefore, extracting N -ary relations jointly from scientific text and tables still remains an important but challenging problem.

We propose RESEL, a hierarchical **Retrieve-and-Selection** model for multi-modal and document-level SciIE. In RESEL, we pose the N -ary relation extraction problem as a question answering task over text and tables (Figure 1). RESEL then decomposes the challenging task into two simpler subtasks: (1) high-level component retrieval, which aims to locate the target paragraph/table where the final target entity resides, and (2) low-level entity extraction, which aims to select the target entity from the chosen component.

For high-level component (*i.e.*, paragraph or table) retrieval, we design a feature set that combines the strengths of two classes of retrieval methods: (1) *sparse retrieval* (Aizawa, 2003; Robertson and Zaragoza, 2009) that represents the query-candidate pairs as high-dimensional sparse vectors to encode lexical features; (2) *dense retrieval* (Karpukhin et al., 2020) that leverages latent semantic embeddings to represent query and candidates. We design sparse and dense retrieval features for query-component pairs by augmenting BERT (Devlin et al., 2019)-based semantic similarities with entity-level semantic and lexical similarities, allowing for training an accurate high-level retriever using only a small amount of labeled data.

The low-level entity extraction stage aims to infer N -ary entity relations from complex and noisy signals across paragraphs and tables. In this stage, we first build a cross-modal entity-correlation graph, which encodes different entity-entity relations such as co-occurrence, co-reference, and table structural relations. While most of the existing methods (Zheng et al., 2020; Zeng et al., 2020) use BERT embeddings as node representations, we find BERT embeddings limited in distinguishing adjacent table cells or similar entities. This issue is even more severe when the BERT embeddings are propagated on the graph. To address this, we design a new *bag-of-neighbors* (BON) representation. It computes the lexical and semantic similarities between each candidate entity and its 1-hop neighbors. We then feed the BON features into a graph attention network (GAT) to capture both neighboring semantics and structural correlations. Such GAT-learned features and BERT-based embeddings are treated as two complementary views, which are co-trained with a consistency loss.

We summarize our key contributions as follows: (1) We propose a hierarchical retrieve-and-select learning method that decomposes N -ary scientific

relation extraction into two simpler subtasks; (2) For high-level component retrieval, we propose a simple but effective feature-based model that combines multi-level semantic and lexical features between queries and components; (3) For low-level entity extraction, we propose a multi-view architecture, which fuses graph-based structural relations with BERT-based semantic information for extraction; (4) Extensive experiments on three datasets show the superiority of both the high-level and low-level modules in RESEL.

2 Related Work

Component Retrieval For component retrieval, traditional sparse retrieval methods such as TF-IDF (Aizawa, 2003) and BM25 (Robertson and Zaragoza, 2009) focus on keyword-level matching but ignore entity semantics. Recently, pre-trained language models have also been used to represent queries and documents in a learned space (Karpukhin et al., 2020) and have been extended to handle tabular context (Herzig et al., 2021; Ma et al., 2022). However, these methods mainly focus on passage-level retrieval, and cannot well capture fine-grained entity-level semantics (Zhang et al., 2020; Su et al., 2021). Such an issue makes them suboptimal for encoding nuanced terms and descriptions in scientific articles. In contrast, RESEL leverages both component- and entity-level semantic and lexical features that help the model better understand the correlations between components and queries.

N -ary Relation Extraction Many existing methods (Jia et al., 2019; Jain et al., 2020; Viswanathan et al., 2021) treat N -ary relation extraction as a binary classification problem and predict whether the composition of N entities in the document are valid or not. However, the candidate space grows exponentially with N , and the performance of the binary classifiers can be largely influenced by the number and quality of negative tuples. Some other methods (Du et al., 2021; Huang et al., 2021) formulate the problem as role-filler entity extraction and propose BERT-based generative models to extract the correct entities for each element of the N -ary relation. None of these methods consider N -ary relation across modalities. Lockard et al. (2020) leverages the layout information for extracting relations from web pages. However, the layout information in science articles are less prominent and harder to be utilized.

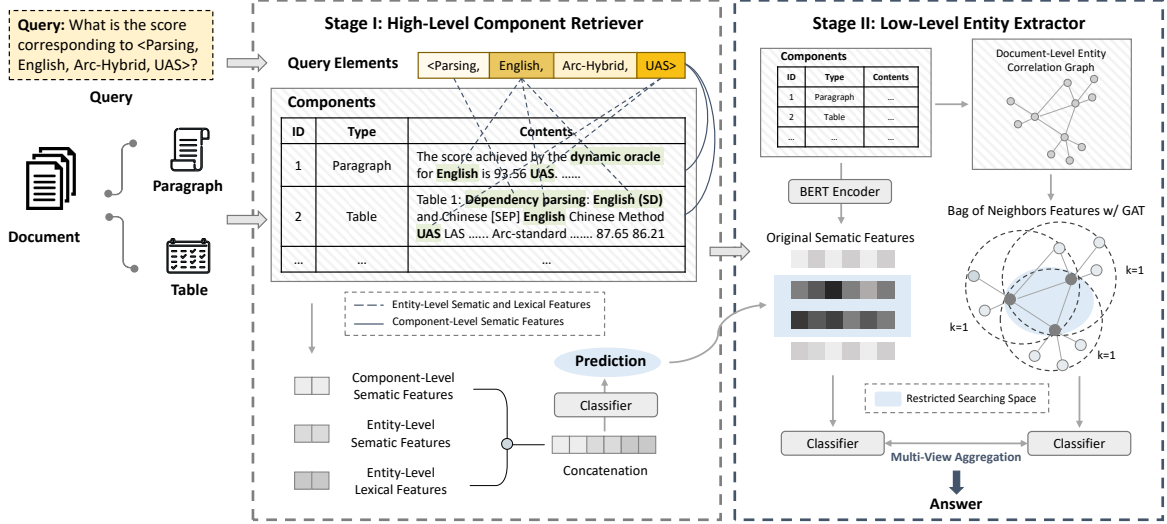


Figure 2: Illustration of RESEL two-stage pipeline: high-level component retrieval and low-level entity extractor.

3 Problem Formulation

In SciIE, we aim to extract information from a corpus of M scientific articles. Each article, denoted as \mathcal{D} , is a sequence of $|\mathcal{D}|$ components, where each component $C_i \in \mathcal{D}$, $i \in \mathbb{N}_{[0, |\mathcal{D}|]}$ can be either a paragraph or a table. A table is flattened and concatenated with its caption as a sequence of words. Given a document \mathcal{D} , we have a set of queries \mathcal{Q} with number $|\mathcal{Q}|$. Each query contains $N - 1$ elements $Q_j = [e_{j,1}, \dots, e_{j,N-1}]$, $j \in \mathbb{N}_{[0, |\mathcal{Q}|]}$, and the task is to extract the correct N -th element from document \mathcal{D} to form a valid N -ary relation. We assume a dataset $\{x_k, y_k\}_{k=1}^M$ that can be used to learn such a N -ary relation extractor, each sample includes a document and a set of queries $x_k = (\mathcal{D}_k, \mathcal{Q}_k)$, and each ground-truth label y_k indicates the target entity in the document \mathcal{D}_k .

4 The RESEL Method

4.1 Component Retriever

In Stage I, we design a high-level model to retrieve the most relevant paragraphs or tables that contain the final answer. We first use BERT to embed the paragraphs/tables into sequences of vectors (details in Appendix A.1). We encode the j -th query $Q_j = [e_{j,1}, \dots, e_{j,N-1}]$, into query embedding $\mathbf{h}(Q_j)$ and get the corresponding element embeddings of the query $\mathbf{h}(e_{j,a})$, $a \in \mathbb{N}_{[0, N]}$. Similar to the query encoder, we encode the i -th component, C_i , as component embedding $\mathbf{h}(C_i)$, and the averaged entity embeddings $\mathbf{h}(m_{i,b})$, where $m_{i,b} \in C_i$, indicates the b -th entity extracted from C_i . With

the encoded sequences of vectors, we compute the different views of features for the component-query pair (C_i, Q_j) as follows to take advantage of both the entity-level matching signals and component-level semantic signals, which are complementary:

Component-Level Semantic Features (CS).

The first view extracts the semantic features for component-query pairs from two different angles: (1) *Embedding-Based Similarity*: the cosine similarities $f_{cs-1}(C_i, Q_j)$ between component and query embeddings. (2) *Entailment-Based Score*: the classification score $f_{cs-2}(C_i, Q_j)$ between Q_j and C_i calculated by feeding them both into a BERT binary sequence classifier as a concatenated sequence (Nogueira and Cho, 2019; Nie et al., 2019). We concat these two scalar features as the first view $\mathbf{f}_{cs}(C_i, Q_j)$.

Entity-Level Semantic Features (ES). The second view computes *entity-level* cosine similarities $f_{es}(m_{i,b}, e_{j,a})$ between the component entity embeddings $\mathbf{h}(m_{i,b})$ and the query elements embeddings $\mathbf{h}(e_{j,a})$. With all these similarity scores, we apply a max-pooling operation over all component entities $m_{i,b}$, and use the obtained maximum $f_{es}(C_i, e_{j,a}) = \max_{m_{i,b} \in C_i} f_{es}(m_{i,b}, e_{j,a})$ to represent the relation between the component C_i and one query element $e_{j,a}$. Then, we gather the relation scores $f_{es}(C_i, e_{j,a})$ as the final entity-level semantic feature vector: $\mathbf{f}_{es}(C_i, Q_j) = [f_{es}(C_i, e_{j,1}), \dots, f_{es}(C_i, e_{j,N-1})]^T$.

Entity-Level Lexical Features (EL). Our third view extracts lexical features between component entities and the query elements. We compute three

text similarities (Appendix A.2): (1) *Levenshtein Distance* (Levenshtein et al., 1966); (2) the length of *Longest Common Substring*; (3) the length of *Longest Common Subsequence*. As the metrics vary in scale according to the length of the strings, we use the normalized metrics $\mathbf{f}_{\text{el}}(m_{i,b}, e_{j,a}) \in [0, 1]^3$ via dividing by involved string lengths. Similar to ES features, we perform max-pooling to obtain the relation scores between the component and a single query element, $\mathbf{f}_{\text{el}}(C_i, e_{j,a}) = \max_{m_{i,b} \in C_i} \mathbf{f}_{\text{el}}(m_{i,b}, e_{j,a})$ and concatenate the results as entity-level lexical features: $\mathbf{f}_{\text{el}}(C_i, Q_j) = [\mathbf{f}_{\text{el}}(C_i, e_{j,1})^\top \oplus \dots \oplus \mathbf{f}_{\text{el}}(C_i, e_{j,N-1})^\top]^\top$.

We aggregate the features to predict which component has the highest probability to contain the final answer. As the features in the three views share the same scale range and similar dimensionality, we just concatenate these features together as $\mathbf{f}^{\text{h}} = [\mathbf{f}_{\text{cs}}^\top \oplus \mathbf{f}_{\text{es}}^\top \oplus \mathbf{f}_{\text{el}}^\top]^\top$, and train one unified classifier over \mathbf{f}^{h} for component retrieval.

4.2 Entity Extractor

In Stage II, we use the predictions from Stage I to restrict the searching space for low-level entity extraction.

4.2.1 Multi-Modal Entity-Level Graph

To model document-level entity correlations, we construct a multi-modal entity correlation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$ denotes the entity nodes, and $\mathcal{E} = \{E_{(v_i, v_j)} | v_i, v_j \in \mathcal{V}; i, j \in \mathbb{N}_{[1, |\mathcal{V}|]}\}$ denotes the edges between them. Each node $v_i \in \mathcal{V}$ represents a paragraph entity or a table cell. We construct different edge types to model the intra- and inter-modality relations to encode the entity correlation across modalities as in Figure 3: (1) **Co-occurrence Edge** measures whether two entity nodes v_i and v_j occur in the same sentence or adjacent sentences; (2) **Co-reference Edge** extracts the relation information of two entity nodes v_i and v_j referring to the same concept; (3) **Reference Edge** bridges the table and text with reference information (e.g., “in Table 3”); (4) **Table-Structure Edge** extracts the structural information of columns and rows of tables; (5) **Table-Paragraph Connection** enhances the linking between table cells and paragraph entities via text similarities (detailed in Appendix A.3).

With these five edge types from different modalities covering nearly all hidden relations in the document, the multi-modal entity correlation graph can effectively model document-level information. As

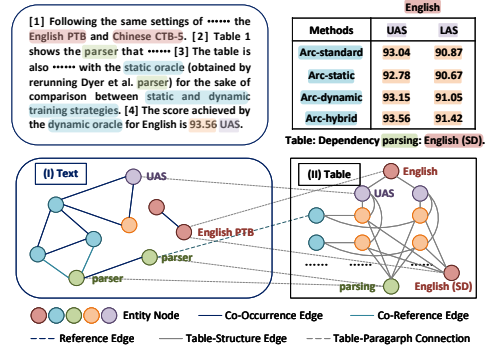


Figure 3: Illustration of constructing the multi-modal entity correlation graph between paragraphs and tables.

all edge types are ranged in $[0, 1]$ and most of them do not overlap, we treat them equally and define the graph as an undirected homogeneous graph.

4.2.2 Bag-of-Neighbors Features

For low-level entity extraction from the retrieved paragraph/table, a key challenge is that the entities (nodes) in the same sentence or adjacent table cells can have very similar BERT embeddings and hard to be discriminated by a BERT-only classifier. Further, such entities often share many common neighbors on the graph, which means their embeddings can be easily further over-smoothed when propagated on the graph. To tackle these challenges, we propose the bag-of-neighbors (BON) features (Figure 4(a)) based on the entity-level semantic and lexical features. Given an entity node v_i and a query Q_j , we define the initial embeddings as:

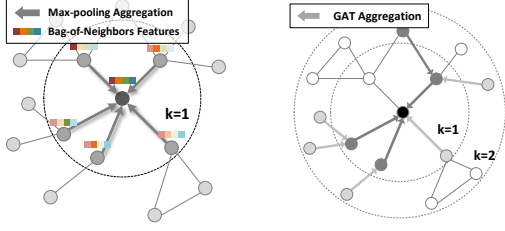
$$\mathbf{g}(v_i) = [f_{\text{es}}(v_i, e_{j,1}) \oplus \dots \oplus f_{\text{es}}(v_i, e_{j,N-1}) \oplus \mathbf{f}_{\text{el}}(v_i, e_{j,1})^\top \oplus \dots \oplus \mathbf{f}_{\text{el}}(v_i, e_{j,N-1})^\top]^\top,$$

where $e_{j,k}$ is the k -th query elements of Q_j . We compute the BON features of node v_i via max-pooling the initial embeddings of the adjacent neighboring nodes $\mathcal{N}(v_i)$:

$$\mathbf{g}_{\text{BON}}(v_i) = \max_{v \in \mathcal{N}(v_i)} \mathbf{g}(v). \quad (1)$$

4.2.3 Graph Attention Network

Using BON features alone may not be expressive enough when there is query information missing in the 1st-order neighborhood. To include multi-hop relations from distant nodes, we apply a graph attention network (Veličković et al., 2018) to aggregate such information (Figure 4(b)). GAT first computes the normalized attention coefficients $\alpha_{i,j}^{(l)}$ between node i in the multi-modal correlation graph



(a) Bag-of-neighbors. (b) Graph attention networks.

Figure 4: Illustration of (a) bag-of-neighbors features and (b) GAT. For both sub-figures, nodes in darker grey contribute more during the aggregation.

and its neighboring node $j \in \mathcal{N}(i)$ in the l -th layer:

$$\alpha_{i,j}^{(l)} = \frac{\exp(\sigma(\mathbf{a}^\top [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}]))}{\sum_{k \in \mathcal{N}(i)} \exp(\sigma(\mathbf{a}^\top [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)} \mathbf{h}_k^{(l)}]))},$$

where $\mathbf{h}_i^{(l)}$ is the l -th layer hidden features of node i , \mathbf{W} is a learnable weight matrix, \mathbf{a} is trainable weight vector parameters, and $\sigma(\cdot)$ is the LeakyReLU(\cdot) activation function. The initial node embeddings are the bag-of-neighbors feature embeddings, *i.e.*, $\mathbf{h}_i^{(0)} = \mathbf{g}_i^{\text{BON}}$. Then, we aggregate the neighbor embeddings as the new $(l+1)$ -th layer node embeddings via computing a weighted sum based on the computed attention coefficients:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{i,j}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l)} \right). \quad (2)$$

For an L -layer GAT, the updated node embedding is denoted as $\mathbf{g}'_{\text{BON}}(i) = \mathbf{h}_i^{(L)}$.

4.2.4 Multi-View Aggregation

Although the GAT-propagated BON representations can enable the model to extract more answers from tables, they can fall short on paragraphs because of the lack of encoding the original semantic information in BERT embeddings. Thus, aside from the graph-based branch introduced in § 4.2.1, § 4.2.2, and § 4.2.3, we add another branch based on the BERT representations of these nodes. However, simply concatenating the BON features and the BERT embeddings might lead to several drawbacks: (1) one of the views dominating the other one during training; (2) different features with different dimensionality, making it difficult to learn a unified classifier on the concatenation. Thus, we design two simple classifiers and make them mutually enhance each other during the entity selection:

(1) one classifier based on the concatenation of the entity nodes' and query elements' BERT embeddings, and (2) the other classifier based on the GAT-updated BON features. Given the node v_i and the query Q_j , and using feedforward neural networks (FFNN) as classifiers, we have:

$$\begin{aligned} \mathbf{h}_s &= [\mathbf{h}(e_{j,1})^\top \oplus \cdots \oplus \mathbf{h}(e_{j,N-1})^\top \oplus \mathbf{h}(v_i)^\top]^\top, \\ \hat{\mathbf{y}}^{(s)} &= \text{FFNN}(\mathbf{h}_s), \hat{\mathbf{y}}^{(n)} = \text{FFNN}(\mathbf{g}'_{\text{BON}}(v_i)). \end{aligned} \quad (3)$$

Then, we average the scores from simple classifiers as the prediction of the final aggregated classifier:

$$\hat{\mathbf{y}}^{\text{low}} = \text{Softmax} \left(\frac{1}{2} (\hat{\mathbf{y}}^{(s)} + \hat{\mathbf{y}}^{(n)}) \right). \quad (4)$$

4.3 Training Objective

Given a document D_k and a query Q_j , $\mathbf{y}_{jk}^{\text{high}}$ and $\mathbf{y}_{jk}^{\text{low}}$ indicate the ground-truth label of the correct component and entity for high-level component retrieval and low-level entity extraction, while $\hat{\mathbf{y}}_{jk}^{\text{high}}$ and $\hat{\mathbf{y}}_{jk}^{\text{low}}$ indicate the predictions from the component retriever and entity extractor. We define the following training objectives:

High-Level Component Retrieval We use the traditional classification loss, $\ell_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_i -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$, as the high-level model training objective:

$$\ell_{\text{high}} = \sum_{k=1}^M \sum_{j=1}^{|\mathcal{Q}|} \ell_{\text{CE}}(\mathbf{y}_{jk}^{\text{high}}, \hat{\mathbf{y}}_{jk}^{\text{high}}). \quad (5)$$

Low-Level Entity Selection The training objective for the low-level entity classifiers (§ 4.2.4) is separated into three parts: (1) the aggregated classification loss for the aggregated model:

$$\ell_1 = \sum_{k=1}^M \sum_{j=1}^{|\mathcal{Q}|} \ell_{\text{CE}}(\mathbf{y}_{jk}^{\text{low}}, \hat{\mathbf{y}}_{jk}^{\text{low}}), \quad (6)$$

(2) the classification loss for the two subclassifiers:

$$\ell_2 = \sum_{k=1}^M \sum_{v \in \{s,n\}} \sum_{j=1}^{|\mathcal{Q}|} \ell_{\text{CE}}(\mathbf{y}_{jk}^{\text{low}}, \hat{\mathbf{y}}_{jk}^{(v)}), \quad (7)$$

(3) the consistency loss between the two subclassifiers to encourage them to reach a consensus:

$$\ell_3 = \sum_{k=1}^M \sum_{u,v \in \{s,n\}} \sum_{j=1}^{|\mathcal{Q}|} \|\hat{\mathbf{y}}_{jk}^{(u)} - \hat{\mathbf{y}}_{jk}^{(v)}\|_2^2. \quad (8)$$

The overall objective of the low-level entity extractor is then:

$$\ell_{\text{low}} = \ell_1 + \lambda \ell_2 + \mu \ell_3, \quad (9)$$

where λ and μ are pre-defined balancing hyper-parameters.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our model on three SciIE datasets: (1) **SciREX** (Jain et al., 2020) contains 438 annotated full-length machine learning papers; (2) **PubMed** (Jia et al., 2019) contains 5688 annotated full-length biochemical papers; (3) **NLP-TDMS (Full)** (Hou et al., 2019) contains 332 unannotated full-length natural language processing papers (see Appendix C for more details). We extend the original datasets to include both text and tables from the LaTeX or PDF files. Our experiments show that the domain-specific BERTs work better than the general domain BERT model. For SciREX and TDMS-NLP, we use SciBERT (Beltagy et al., 2019) as the encoder for all methods; for PubMed, we use ClinicalBERT (Alsentzer et al., 2019).

Component Retrieval Baselines. Our baselines contain (1) *sparse retrieval methods*: **TF-IDF** (Aizawa, 2003), **BM25** (Robertson and Zaragoza, 2009); (2) *entity-based methods*: **Entity Cosine Similarities (ECS)**, **Deep Entity Cosine Similarities (DECS)**; (3) *embedding-based methods*: **BERT-Matching (BERT-M)**, **BERT-Entailment (BERT-E)** (Nogueira and Cho, 2019; Nie et al., 2019), **Recurrent Retriever (RR)** (Asai et al., 2019), **Dense Passage Retrieval (DPR)** (Karpukhin et al., 2020).

Entity Selection Baselines. For the low-level model, we restrict the searching space to the ground-truth paragraph/table that contains the answer. The baselines include (1) *BERT-based methods*: **BERT-Base**, **SciREX** (Jain et al., 2020); (2) *graph-based methods*: **Graph Convolutional Network (GCN)** (Kipf and Welling, 2016), **GAT** (Veličković et al., 2018), **Heterogeneous Document-Entity (HDE) graph** (Tu et al., 2019); (3) *pre-trained language models*: **TAPAS** (Herzig et al., 2020), **TDMS-IE** (Hou et al., 2019).

Overall Baselines. (1) **BERT-Base** model searching in the whole document; (2) **GCN** and

(3) **GAT** testing the performance of our proposed graph in the whole document; (4) **BERT-Entailment+Base** combining the best baselines for both high- and low-level stage (see Appendix E for more details about baselines).

Metrics. Following existing works (Karpukhin et al., 2020), we use (1) Accuracy (Acc), (2) Mean Reciprocal Rank (MRR), and (3) Top-k Hit Rate (Hit@K) with $k = 2, 3, 5$ for evaluating both high- and low-level models (see Appendix D).

5.2 Comparison with Baselines

Table 1–3 present the average performances over multiple random trials². RESEL consistently outperforms the strongest baselines by 9.01%, 6.81%, 10.25% in Acc and 4.38%, 12.01%, 8.27% in MRR on the three datasets at all levels. For the other ranking metrics of hit rate, RESEL also show marginal improvements compared with baselines.

As RESEL-H employs both component-level semantic features and entity-level matching features, its high-level performance exceeds that of the sparse and dense retrieval baselines, which captures only single-sided information. For low-level, the embedding-based methods and pretrained-LMs only take advantage of the latent semantic information, while the graph-based methods only focus on the graph topology and can easily suffer from over-smoothing. Compared with these baselines, RESEL-L shows better performances with the multi-view aggregation of GAT-propagated BON features and BERT embeddings.

Comparing the performance gains for the low-level extraction, we find that the components of RESEL-L contribute variously on different datasets. In SciREX, with most of the questioned scores hidden in the tables, the GAT-propagated BON features work better in discriminating the table cells and numeric values. For PubMed, the targets mostly appear in text rather than tables. Thus, the semantic information in BERT embeddings contributes more to the performance increase. NLP-TDMS is a benchmark dataset that includes multiple relevant choices for a given query, the ambiguity of which hurts the performance of all models.

5.3 Ablation Studies

We conduct ablation studies on SciREX and present the results in Table 4.

²The standard deviation is reported in Appendix F.

	SciREX					PubMed					NLP-TDMS				
Methods	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5
TF-IDF	9.31	17.27	12.64	14.48	17.28	30.41	46.60	45.43	54.19	65.71	9.71	18.31	13.31	17.27	25.76
BM25	27.44	42.86	39.90	50.75	63.81	28.29	44.04	42.43	50.56	62.08	13.38	24.19	19.95	24.51	33.14
ECS	16.37	29.11	20.34	25.03	42.00	17.65	32.50	27.91	34.92	47.93	22.84	36.00	31.28	38.88	45.52
DECS	25.82	43.48	37.47	52.68	71.89	37.36	45.57	43.93	52.38	64.14	28.20	46.87	45.67	59.56	68.52
BERT-M	52.38	67.54	53.97	61.11	76.19	45.78	61.13	62.58	70.84	80.78	46.18	62.39	64.83	73.72	83.98
BERT-E	60.59	75.98	82.04	88.92	98.37	47.35	63.28	66.29	73.30	81.64	50.77	66.97	64.62	86.15	86.15
RR	25.42	-	-	-	-	35.29	-	-	-	-	31.87	-	-	-	-
DPR	53.47	50.26	58.42	74.25	88.96	45.31	61.47	64.46	73.22	81.48	57.14	72.98	76.19	85.71	97.62
ReSel-H	71.48	82.48	85.59	93.01	98.61	49.02	63.67	66.21	74.43	83.81	71.62	79.21	83.33	91.87	99.36

Table 1: The performance of different methods for retrieving high-level components. We measure the performance of different methods in retrieving the ground-truth components in terms of accuracy, MRR, and top- k hit ratios.

	SciREX					PubMed					NLP-TDMS				
Methods	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5
Base	14.72	25.81	20.96	28.61	40.60	72.50	77.36	82.01	89.44	95.87	9.37	21.42	15.62	21.88	33.54
SciREX	14.21	23.25	20.42	26.56	37.23	52.44	63.42	70.97	80.36	91.45	11.35	23.75	17.86	26.33	41.09
GCN	10.74	21.60	16.72	22.87	31.00	57.36	72.37	77.09	87.02	94.61	12.79	21.60	17.55	20.31	37.24
GAT	12.09	21.72	17.39	22.28	34.23	57.44	71.35	77.13	86.93	95.95	14.69	25.12	17.62	33.61	42.58
HDE	14.78	24.79	21.12	27.07	33.70	60.73	72.90	81.32	87.77	93.93	15.38	28.74	19.23	32.47	46.75
TAPAS	25.45	-	-	-	-	8.63	-	-	-	-	23.79	-	-	-	-
TDMS-IE	18.41	-	-	-	-	6.42	-	-	-	-	13.44	-	-	-	-
ReSel-L	41.68	51.45	49.30	55.70	65.50	74.71	79.57	84.22	91.06	97.57	25.77	36.91	25.91	40.64	49.72

Table 2: The performance of different low-level methods in extracting the target entities.

Multi-View Features. **CS features** enable RESEL-H to measure the matching scores between the main topics of the components and the query. Removing this feature will cause larger performance degradation than removing the other two features, indicating that for high-level retrieval, component-level information guides primary retrieval, while entity-level information refines it. At the entity level, **ES features** and **EL features** allow the model to capture semantic and lexical relevance between paragraph/table and query elements. By removing these entity-level features, the model will rely solely on BERT embeddings, which are less expressive for lengthy paragraphs and tables.

Graph-based Branch. When we replace the **BON features** with BERT embeddings, RESEL-L’s performance drops a lot. This demonstrates that BERT embeddings cannot discriminate table entities well, especially for numeric values and adjacent table cells. In contrast, the **BON features**, encoding neighboring information and graph topology, can distinguish such entities. On the other hand, **GAT** also improves the performance. As **BON features** are based on adjacent nodes, the

pooling aggregation is only over 1st-order neighbors. Thus, **GAT** can complement **BON features** by propagating distant neighborhood information.

Original Semantic Branch. For low-level model, including BERT-based **Original Semantic (OS) Features** improves the performance. Although the improvement is marginal on SciREX, it is notable on other datasets (e.g., PubMed) where many answers reside in paragraphs. Removing **Multi-View Aggregation (MVA)** will make RESEL-L’s performance decrease significantly. This is because when simply concatenating the BERT embeddings and the **GAT**-propagated **BON features**, the BERT embeddings (which has much higher dimensionality) can dominate the learning process.

5.4 Parameter Studies

Figure 5 shows our parameter study results.

λ and μ . The loss ℓ_1 in the aggregated classifier in Eq. (4) plays the leading role in training objective. When λ is too small or μ is too large, the regularization of consistency between two classifiers will contribute more than their respective classifica-

Methods	SciREX					PubMed					NLP-TDMS				
	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5
Base	6.53	11.35	9.42	10.42	15.15	26.63	30.16	26.06	33.56	43.21	3.13	5.62	4.14	4.80	6.66
GCN	4.03	5.87	5.54	6.11	10.11	16.63	25.95	21.42	28.47	40.41	7.61	16.73	9.18	16.83	20.92
GAT	8.44	11.93	9.73	10.19	13.47	16.80	26.21	22.79	29.60	38.59	9.82	16.24	13.13	14.42	15.79
BERT-E+B	16.27	22.85	18.02	23.98	31.09	29.30	37.59	41.17	43.36	53.50	7.58	11.86	7.18	11.48	13.34
ReSel	38.69	43.66	42.90	46.35	48.51	33.73	40.77	42.38	44.99	46.19	13.71	17.55	15.06	18.80	22.87

Table 3: The overall document-level extraction performance of different methods.

High-Level Models	Acc	MRR	Hit@2	Hit@3	Hit@5
ReSel-H w/o CS	34.12	55.02	57.93	73.01	81.74
ReSel-H w/o ES	58.73	72.03	72.22	79.36	92.06
ReSel-H w/o EL	52.38	67.88	66.66	76.98	96.03
ReSel-H	64.28	74.90	74.60	80.15	92.06
Low-Level Models	Acc	MRR	Hit@2	Hit@3	Hit@5
ReSel-L w/o BON	15.86	26.15	21.37	25.51	37.93
ReSel-L w/o GAT	41.37	45.50	41.37	41.37	57.24
ReSel-L w/o OS	48.28	53.08	51.03	55.86	60.69
ReSel-L w/o MVA	19.31	27.65	24.13	27.58	33.10
ReSel-L	51.72	57.60	56.55	60.00	64.83

Table 4: Performance comparison of ablation study.

tion loss, making them more intended to generate incorrect-but-same predictions; Conversely, when λ is too large or μ is too small, the classifiers begin to generate biased predictions, making the aggregation deteriorate to a mere average.

L and H . The number of GAT layers L determines the depth of neighboring information on the graph, also known as the orders of neighbors in aggregation. When L is increasing, we will aggregate more common neighbors for adjacent nodes, making it easier for GAT to fall into over-smoothing. The width of neighboring information on the graph is dictated by the amount of relationships we encode from neighbors. When we increase the number of attention heads H , GAT will learn and combine several sets of attention scores on the neighboring nodes, which can also include more irrelevant or misleading information from them. Besides, whichever H or L is increasing, the model needs to train more parameters, taking more time and data.

5.5 Case Study

Figure 6 shows a representative example to illustrate the efficacy of RESEL. It shows the predictions from GCN baseline and RESEL for two queries on the same document. The darker the color is on the table cell, the higher prediction score we obtain for it. We can clearly see that BERT embeddings alone cannot distinguish which numerical value is the final answer. The graph-propagated

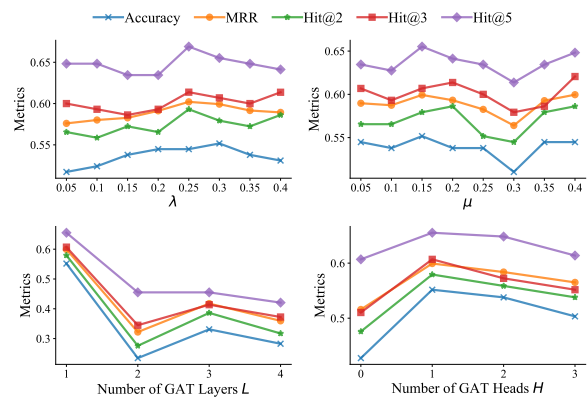


Figure 5: Parameter study of RESEL on SciREX.

embeddings are brought even closer in semantic space by linking with related items in column and row headers. However, with the BON features and graph topology, RESEL can distinguish different values in the table and make correct selections.

Table 2: BLEU scores performance on EMNLP2017 WMT.

Method	SeqGAN	RankGAN	LeakGAN	p -value
BLEU-2	0.8590	0.778	0.956	$< 10^{-6}$
BLEU-3	0.6015	0.478	0.819	$< 10^{-6}$
BLEU-4	0.4541	0.411	0.627	$< 10^{-6}$
BLEU-5	0.4498	0.463	0.498	$< 10^{-6}$

Table 2: BLEU scores performance on EMNLP2017 WMT.

Method	SeqGAN	RankGAN	LeakGAN	p -value
BLEU-2	0.8590	0.778	0.956	$< 10^{-6}$
BLEU-3	0.6015	0.478	0.819	$< 10^{-6}$
BLEU-4	0.4541	0.411	0.627	$< 10^{-6}$
BLEU-5	0.4498	0.463	0.498	$< 10^{-6}$

Q1: ['emnlp2017_wmt', 'leakgan', 'bleu-3', 'text_generation']; A1: 0.819

Q2: ['emnlp2017_wmt', 'rankgan', 'bleu-4', 'text_generation']; A2: 0.411

Figure 6: The case study of predictions made by the baseline GCN model and RESEL model.

6 Conclusion

We proposed RESEL, a two-stage method for N -ary relation extraction jointly from scientific text and tables. RESEL consists of two key components: a high-level component retriever and a low-level entity extractor. The multiple features defined in the high-level retriever enables our model to leverage semantic and lexical information from

both paragraphs/tables and entities. For low-level entity extractor, the multi-view aggregation effectively encodes both the topology information from the graph and the semantic information from pre-trained BERT embeddings. Extensive experiments on three datasets show that RESEL consistently outperforms all baseline models significantly.

Limitations

While RESEL has demonstrated superior performance compared with the state-of-the-art baselines, it has several limitations that can be addressed in the future. First, although RESEL extends the capability of previous N -ary relation extraction to both text and tables, it cannot extract from images—another important data modality in scientific articles. This necessitates augment RESEL with optical character recognition (OCR) techniques to parse images and jointly extract from the text, table, and image modalities. Second, we found the datasets for SciIE are limited and expensive to curate, especially as we aim to expand to include images. Accurate annotation for multi-modal SciIE is time-consuming and needs more future collaborative efforts from related communities. Third, currently RESEL has not modeled the layout information (*e.g.*, font style, font size, *etc.*), which may also contain some clues for intra- and inter-modality relations. Some existing studies (Xu et al., 2020, 2021a,b) have worked on pre-training models that encode the layout information, which can be interesting to be combined with RESEL.

Acknowledgements

This work was supported in part by NSF (IIS-2008334, IIS-2106961, CAREER IIS-2144338), ONR (MURI N00014-17-1-2656), and Kolon Industries.

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf-idf measures. *Information Processing & Management*, 39(1).

Emily Alsentzer, John Murphy, William Boag, Weihung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. [Publicly available clinical BERT embeddings](#). In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *ICLR*.

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.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021. [GRIT: Generative role-filler transformers for document-level event entity extraction](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, Online. Association for Computational Linguistics.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. [Open domain question answering over tables via dense retrieval](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, Online. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. [TaPas: Weakly supervised table parsing via pre-training](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

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.

- Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. [Document-level entity-based extraction as template generation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic. 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 multi-scale 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.
- Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. 2019. The role of "condition" a novel scientific knowledge graph representation and construction model. In *KDD*, pages 1634–1642.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Thomas N Kipf and Max Welling. 2016. [Semi-supervised classification with graph convolutional networks](#). *arXiv preprint arXiv:1609.02907*.
- Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10.
- Yu Liu, Quanming Yao, and Yong Li. 2021. [Role-aware modeling for n-ary relational knowledge bases](#). In *Proceedings of the Web Conference 2021*, pages 2660–2671.
- Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. 2020. [ZeroShotCeres: Zero-shot relation extraction from semi-structured web-pages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8105–8117, 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.
- Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022. [Open domain question answering with a unified knowledge interface](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1620, Dublin, Ireland. Association for Computational Linguistics.
- Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. [Revealing the importance of semantic retrieval for machine reading at scale](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2553–2566, Hong Kong, China. Association for Computational Linguistics.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. [Passage re-ranking with bert](#). *arXiv preprint arXiv:1901.04085*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Stephen Robertson and Hugo Zaragoza. 2009. *The Probabilistic Relevance Framework: BM25 and Beyond*. Now Publishers Inc.
- Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. [Whitening sentence representations for better semantics and faster retrieval](#). *arXiv preprint arXiv:2103.15316*.
- Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. [Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713, Florence, Italy. Association for Computational Linguistics.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. [Graph attention networks](#). In *ICLR*.
- Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. [CitationIE: Leveraging the citation graph for scientific information extraction](#). 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 719–731, Online. Association for Computational Linguistics.

- Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021a. [LayoutLMv2: Multi-modal pre-training for visually-rich document understanding](#). 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 2579–2591, Online. Association for Computational Linguistics.
- Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. [Layoutlm: Pre-training of text and layout for document image understanding](#). In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.
- Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021b. [Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding](#). *arXiv preprint arXiv:2104.08836*.
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. [TaBERT: Pretraining for joint understanding of textual and tabular data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.
- Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. [Double graph based reasoning for document-level relation extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.
- Zhong Zhang, Chongming Gao, Cong Xu, Rui Miao, Qinli Yang, and Junming Shao. 2020. [Revisiting representation degeneration problem in language modeling](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 518–527, Online. Association for Computational Linguistics.
- Bo Zheng, Haoyang Wen, Yaobo Liang, Nan Duan, Wanxiang Che, Daxin Jiang, Ming Zhou, and Ting Liu. 2020. [Document modeling with graph attention networks for multi-grained machine reading comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6708–6718, Online. Association for Computational Linguistics.

A Methodology Computational Details

A.1 Query and Component Encoder

• **Query Encoder** Each query Q_j contains $(N - 1)$ elements $[e_{j,1}, \dots, e_{j,N-1}]$. To generate a more understandable natural language sequence for the BERT encoder, we re-formulate the query into a question $[q_{j,1}, \dots, q_{j,M_j}]$, where M_j is the number of words in the generated question. In this way, we are able to use the [CLS] token embedding as the query embedding:

$$\mathbf{h}(Q_j) = \text{BERT}_{[\text{CLS}]}([q_{j,1}, \dots, q_{j,M_j}]). \quad (10)$$

$$\{\mathbf{h}(q_{j,1}), \dots, \mathbf{h}(q_{j,M_j})\} = \text{BERT}([q_{j,1}, \dots, q_{j,M_j}]). \quad (11)$$

By averaging the embeddings of words that are related to the query elements, we obtain the a -th query element embeddings $\mathbf{h}(e_{j,a}), e_{j,a} \in Q_j$:

$$\mathbf{h}(e_{j,a}) = \frac{\sum_{k=1}^{M_j} \mathbf{h}(q_{j,k}) \cdot \mathbb{I}(q_{j,k} \in e_{j,a})}{\sum_{k=1}^{M_j} \mathbb{I}(q_{j,k} \in e_{j,a})}. \quad (12)$$

• **Component Encoder** As is mentioned in § 3, each component in a document can be denoted as a sequence of words $[w_{i,1}, \dots, w_{i,N_i}]$. Then, we directly encode the paragraph embedding $\mathbf{h}(C_i)$, the included word embeddings $\{\mathbf{h}(w_{i,1}), \dots, \mathbf{h}(w_{i,N_i})\}$, and the averaged entity embeddings $\mathbf{h}(m_{i,b})$, where $m_{i,b} \in C_i$ indicates the b -th entity extracted from the component C_i :

$$\mathbf{h}(C_i) = \text{BERT}_{[\text{CLS}]}([w_{i,1}, \dots, w_{i,N_i}]), \quad (13)$$

$$\{\mathbf{h}(w_{i,1}), \dots, \mathbf{h}(w_{i,N_i})\} = \text{BERT}([w_{i,1}, \dots, w_{i,N_i}]), \quad (14)$$

$$\mathbf{h}(m_{i,b}) = \frac{\sum_{k=1}^{N_i} \mathbf{h}(w_{i,k}) \cdot \mathbb{I}(w_{i,k} \in m_{i,b})}{\sum_{k=1}^{N_i} \mathbb{I}(w_{i,k} \in m_{i,b})}. \quad (15)$$

A.2 Text Similarities

• **Levenshtein Similarity:** The string similarity based on Levenshtein Distance (Levenshtein et al., 1966):

$$\text{Leven_Sim}(m_{i,b}, e_{j,a}) = 1 - \frac{\text{Leven_Dist}(m_{i,b}, e_{j,a})}{\max(|m_{i,b}|, |e_{j,a}|)}, \quad (16)$$

where $\text{Leven_Dist}(\cdot, \cdot)$ refers to the Levenshtein Distance, which measures how different two strings

are by counting the number of deletions, insertion, or substitutions required to transform one string to another.

• **Longest Common Substring:** The ratio between the length of longest common substring and the minimum length of the two strings:

$$\text{LCStr_Sim}(m_{i,b}, e_{j,a}) = \frac{|\text{LCStr}(m_{i,b}, e_{j,a})|}{\min(|m_{i,b}|, |e_{j,a}|)}, \quad (17)$$

where $\text{LCStr}(\cdot, \cdot)$ indicates the longest common substring between two given strings.

• **Longest Common Subsequence:** The longest common subsequence (LCS) is the longest subsequence that is common to all given strings. Different from the longest common substring, the elements of the subsequence are not needed to occupy consecutive locations within the original sequences. The ratio between the length of longest common subsequence and the minimum length of the two strings:

$$\text{LCSeq_Sim}(m_{i,b}, e_{j,a}) = \frac{|\text{LCSeq}(m_{i,b}, e_{j,a})|}{\min(|m_{i,b}|, |e_{j,a}|)}, \quad (18)$$

where $\text{LCSeq}(\cdot, \cdot)$ indicates the longest common subsequence between two given strings.

A.3 Cross-Modal Graph Construction

• **Co-occurrence Edge:** When two entity nodes v_i and v_j occur in the same sentence, we connect them with a co-occurrence edge $E_{(v_i, v_j)}$ with weight w_s . If v_i and v_j do not co-occur in the same sentence but in two adjacent sentences, we still connect them but assign a smaller weight w_t ($w_t < w_s$) to edge $E_{(v_i, v_j)}$. In practice, we set $w_t = w_s/2$.

• **Co-Reference Edge** extracts the intra-paragraph relation. When two entity nodes v_i and v_j are referring to the same concept, we connect them with co-reference edge $E_{(v_i, v_j)}$, e.g., abbreviations and full names, common names and scientific names, etc.

• **Reference Edge** extracts the inter-modality relationship between paragraphs and tables. When an entity node v_i occurs in a sentence with a reference mark, (e.g., “Table. 3”, etc.), we link it to any node v_j in the referenced table with a reference edge $E_{(v_i, v_j)}$.

• **Table-Structure Edge** extracts the intra-table relation. We connect a table-structure edge $E_{(v_i, v_j)}$ between a table cell node v_i and another node v_j appearing in the corresponding column header, row header, or the table caption.

Dataset	SciREX	PubMed	NLP-TDSM
Query Elements	<Task, Method, Dataset, Metric>	<Drug, Gene>	<Task, Dataset, Metric>
Answer Type	Score	Mutation	Score
Training/Val/Test	263/88/87	2366/592/799	200/66/66

Table 5: Datasets details. The numbers in training/val/test set split are the numbers of full-length scientific paper documents.

- **Table-Paragraph Connection** bridges the paragraph-table relation. Given an entity node v_i in a paragraph and a cell node v_j in a table, we place a table-paragraph connection edge $E_{(v_i, v_j)}$ between them. The weight of $E_{(v_i, v_j)} \in [0, 1]$ is computed based on text similarities between the surface strings of two nodes.

B Implementation Details

B.1 Hyper-parameters Settings

For the high-level component retriever training, the learning rate is set as $1e - 4$ and the maximum number of epochs is 50. For low-level entity selector training, we use a 1-layer single head GAT (Veličković et al., 2018) based on the bag-of-neighbors features computed on 1-st order neighbors to aggregate graph topology information. We select $\lambda = 0.3$ and $\mu = 0.15$ as the proportion weights in the multi-view aggregation. For the feature-based FFNN classifiers in both high-level and low-level models, we set the dimensions of the hidden layers to 32. The corresponding learning rate and maximum number of epochs to the low-level entity extractor are $1e - 3$ and 50. During training, we use the Adam (Kingma and Ba, 2014) optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ in our experiments for all the models. We select the best set of hyper-parameters of the models based on the accuracy on the corresponding dev sets.

B.2 Implementation Settings

We train and test our code on the System Ubuntu 18.04.4 LTS with CPU: Intel(R) Xeon(R) Silver 4214 CPU@ 2.20GHz and GPU: NVIDIA GeForce RTX 2080. We implement our method using Python 3.8 and PyTorch 1.6 (Paszke et al., 2019).

C Dataset Description

We evaluate our work on three different datasets (see Table 5): (1) **SciREX** (Jain et al., 2020) contains 438 annotated full-length papers, related to machine learning research. We extend the original SciREX dataset by extracting the tabular data

of each paper from the corresponding raw LaTeX or PDF files. For this dataset, the queries are in the format of <Task, Method, Dataset, Metric> and the final answer we aim to find from the documents is the corresponding score; (2) **NLP-TDMS (Full)** (Hou et al., 2019) contains 332 unannotated full-length papers, including both the text data and the tabular data, related to the natural language processing research. For this dataset, the queries are in the format of <Task, Dataset, Metric> and we are looking for the corresponding scores; (3) **PubMed** (Jia et al., 2019) is created by automatically labeling biomedical literature with Gene Drug Knowledge Database. The dataset contains 5688 annotated full-length papers, related to biochemical research. The queries designed for this dataset are in the format of <Gene, Drug>, and the task is to extract the most influenced Mutations.

D Detailed Evaluation Protocol

Given n samples in the test set, assume that $\{\hat{y}_1, \dots, \hat{y}_n\}$ and $\{y_1, \dots, y_n\}$ are the model predictions and ground-truth labels, respectively. Besides, $\{\hat{y}_1^k, \dots, \hat{y}_n^k\}$ indicates the top- k selections made by the models for each test example. The high-level and low-level models use the same set of evaluation metrics, with the only difference that the high-level models use component-level labels, while the low-level models use the entity-level labels. We use the following metrics for all the methods: (1) **Accuracy (Acc)** measures the exact match for queries in the test set. It only counts the cases when the prediction equals to the ground truth:

$$\text{Acc} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(y_i = \hat{y}_i);$$

(2) **Mean Reciprocal Rank (MRR)** is the average reciprocal ranks of a query’s ground-truth answer among all the candidates:

$$\text{MRR} = \frac{1}{n} \sum_{i=1}^n \frac{1}{\text{rank}(y_i)};$$

SciREX						PubMed					NLP-TDMS				
Methods	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5
TF-IDF	4.28	5.08	5.60	6.08	6.48	-	-	-	-	-	3.65	1.85	3.15	1.03	5.47
BM25	10.01	9.30	11.23	10.14	11.39	-	-	-	-	-	2.40	1.45	1.76	3.46	3.88
ECS	2.67	6.05	8.28	6.13	13.56	-	-	-	-	-	0.33	0.31	2.90	0.59	1.29
DECS	7.25	9.23	18.84	13.05	6.08	2.39	7.67	9.73	7.70	6.11	7.25	9.87	14.55	16.78	14.22
BERT-M	11.33	6.72	7.89	6.87	3.47	1.77	1.77	2.12	1.77	2.74	3.26	3.12	5.79	3.91	5.74
BERT-E	10.43	5.88	8.42	7.96	4.23	1.67	1.56	2.41	2.41	0.99	5.74	5.21	6.43	7.92	8.59
RR	6.72	-	-	-	-	8.30	-	-	-	-	7.44	-	-	-	-
DPR	5.05	3.14	2.89	1.94	0.26	1.02	1.72	1.43	1.92	2.25	2.09	0.86	0.66	9.26	6.88
Ours-H	3.80	2.36	2.42	1.77	0.36	0.81	0.41	0.50	0.62	0.51	0.81	0.41	0.50	0.62	0.51

Table 6: The standard deviation of different methods for retrieving high-level components in terms of Acc, MRR, and top- k hit ratios.

SciREX						PubMed					NLP-TDMS				
Methods	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5
Base	7.35	7.39	7.12	7.66	11.50	0.57	1.02	0.82	0.84	0.45	2.53	3.26	2.44	2.72	3.13
SciREX	13.87	9.63	13.28	11.45	12.09	1.63	1.29	0.77	2.31	1.87	3.95	3.60	3.35	4.92	4.83
GCN	2.77	4.82	3.31	5.65	8.60	1.59	1.02	2.06	0.75	0.55	3.64	3.72	2.16	3.88	3.81
GAT	3.32	4.89	4.17	6.17	9.47	0.88	0.56	1.63	0.76	0.08	2.38	2.99	2.97	3.11	3.19
HDE	3.86	5.98	6.22	10.08	13.07	1.88	1.04	2.43	2.87	2.55	4.30	3.46	2.45	3.21	3.38
TAPAS	2.76	-	-	-	-	1.07	-	-	-	-	3.66	-	-	-	-
TDMS-IE	4.39	-	-	-	-	1.52	-	-	-	-	4.87	-	-	-	-
Ours-L	4.28	5.25	6.28	4.27	0.70	1.12	1.91	0.21	0.80	0.95	3.26	2.21	3.33	3.87	3.36

Table 7: The standard deviation of different low-level methods in extracting the target entities.

(3) **Top-k Hit Rate (Hit@K)** measures whether the ground-truth answer is included in the top- k selection made by the models:

$$\text{Hit@k} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(y_i \in \hat{y}_i^k).$$

For Top- k Hit Rate, we report Hit@2, Hit@3, and Hit@5. For the high-level scenario, we only evaluate whether the model select the correct component or not; For the low-level scenario, we evaluate the performances restricting the searching space into the ground-truth paragraph/table for entity selection; For the overall scenario, we remove the restriction and test the model performance of entity selection from the whole document.

E Experimental Baselines

We use different baselines for the high-level component retrieval, low-level entity selection, and the overall framework.

High-Level Baselines. For the high-level model, we compare with the following baselines:

• **Sparse Retrieval Methods:** 1) **TF-IDF** (Aizawa, 2003) and 2) **BM25** (Robertson and Zaragoza,

2009) are two sparse-retrieval methods which ranks query-section pairs via computing the relevant score based on key words;

• **Entity-Based Methods:** 1) **Entity Cosine Similarities (ECS)** calculates the cosine similarities between the embeddings of query and section entities, and sums them up as the final prediction score; 2) **Deep Entity Cosine Similarities (DECS)** improves cosine similarities by substituting the sum-up function into a feedforward neural network.

• **Embedding-Based Methods:** 1) **BERT-Matching** is a matching method based on pre-trained BERT embeddings, using the dot product between query and component representations; 2) **BERT-Entailment** is a textual inference method (Nogueira and Cho, 2019; Nie et al., 2019) for calculating the relevance score. 3) **Recurrent Retriever** (Asai et al., 2019) is a graph-based recurrent retrieval method. It selects one paragraph p_i in each step until it selects an end-of-evidence mark ([EOE]); 4) **Dense Passage Retrieval (DPR)** (Karpukhin et al., 2020) is a state-of-the-art model that use BERT as the encoder for passage retrieval in open-domain QA.

Methods	SciREX					PubMed					NLP-TDMS				
	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5	Acc	MRR	Hit@2	Hit@3	Hit@5
Base	3.40	2.74	2.08	2.24	3.65	1.35	2.14	3.72	1.79	2.59	0.73	0.16	0.07	0.43	1.15
GCN	3.19	3.76	3.08	3.08	2.70	3.16	1.94	2.41	3.30	3.29	0.80	0.56	0.74	1.41	1.18
GAT	2.12	3.27	2.62	8.19	1.80	5.85	2.05	5.73	3.83	6.52	1.49	1.23	0.57	1.63	2.09
BERT-E+B	8.58	6.36	3.11	6.04	2.73	3.41	6.73	5.82	5.01	6.16	1.28	2.22	1.44	1.25	1.38
Ours	3.21	3.45	2.29	4.85	2.46	2.86	3.77	1.45	3.54	3.66	1.43	1.10	1.69	0.94	1.94

Table 8: The standard deviation of overall document-level extraction performance of different methods.

Low-Level Baselines. For the low-level model, we restrict the searching space to the ground-truth paragraph/table that contains the final answer and compare with the following baselines:

- **Embedding-Based Methods:** 1) **BERT-Base** is a simple classifier trained directly on the concatenation of query and candidate embeddings; 2) **SciREX** (Jain et al., 2020) composes salient entity embeddings for each paragraph and learns a binary classifier to decide whether the N -ary relation exists or not.

- **Graph-Based Methods:** 1) **Graph Convolutional Network (GCN)** (Kipf and Welling, 2016) and 2) **Graph Attention Network (GAT)** (Veličković et al., 2018) are two classic graph neural network structures, we report performances by applying them on our proposed graph structure; 3) **Heterogeneous Document-Entity (HDE) graph** (Tu et al., 2019) is a heterogeneous graph model which conducts multi-hop reading comprehension by leveraging the relation between document, entity, and candidate nodes;

- **Pre-trained LMs:** 1) **TAPAS** (Herzig et al., 2020) is the start-of-the-art pre-trained model on text and tables. We fine-tune the pre-trained model on our own datasets; 2) **TDMS-IE** (Hou et al., 2019) is an entailment model based on the score context and hypothesis of dataset and metric to judge whether these elements are related to each other.

Overall Baselines. For the overall performance our two-stage model, we compare with the following baselines: 1) The **BERT-Base** model searching in the whole document; 2) **GCN** and 3) **GAT** testing the performance of our proposed graph in the whole document; 4) **BERT-Entailment+Base** is a two-stage model combining the best baselines for both high-level and low-level stage.

F Standard Deviation of Main Results

Table 6–Table 8 list the standard deviations we obtain for the main results from multiple trials. The results indicate that RESEL show competitive stability compared with all the baselines on three different datasets under different settings. The evaluation computation is based on the number of queries, but we split the training, validation, and test set based on the number of documents to prevent data leakage. Due to the fact that various documents include varying numbers of queries, the exact number of queries in train/val/test set is not fixed, causing the performances to vary in different trials and the standard deviations to increase. For PubMed dataset, the test set is fixed and the random seeds can only influence the split between training and validation set, thus the standard deviations on this dataset is relatively smaller than on the other two datasets, SciREX and NLP-TDMS.