International Journal on Digital Libraries
https://doi.org/10.1007/s00799-023-00393-y

A BERT-based sequential deep neural architecture to identify contribution statements and extract phrases for triplets from scientific publications

Komal Gupta[1] · **Ammaar Ahmad**[1] · **Tirthankar Ghosal**[2] · **Asif Ekbal**[1]

Received: 1 September 2022 / Revised: 21 December 2023 / Accepted: 22 December 2023

Abstract
Research in Natural Language Processing (NLP) is increasing rapidly; as a result, a large number of research papers are being published. It is challenging to find the contributions of the research paper in any specific domain from the huge amount of unstructured data. There is a need for structuring the relevant contributions in Knowledge Graph (KG). In this paper, we describe our work to accomplish four tasks toward building the Scientific Knowledge Graph (SKG). We propose a pipelined system that performs contribution sentence identification, phrase extraction from contribution sentences, Information Units (IUs) classification, and organize phrases into triplets (*subject, predicate, object*) from the NLP scholarly publications. We develop a multitasking system (ContriSci) for contribution sentence identification with two supporting tasks, *viz. Section Identification* and *Citance Classification*. We use the Bidirectional Encoder Representations from Transformers (BERT)—Conditional Random Field (CRF) model for the phrase extraction and train with two additional datasets: *SciERC* and *SciClaim*. To classify the contribution sentences into IUs, we use a BERT-based model. For the triplet extraction, we categorize the triplets into five categories and classify the triplets with the BERT-based classifier. Our proposed approach yields the F1 score values of 64.21%, 77.47%, 84.52%, and 62.71% for the contribution sentence identification, phrase extraction, IUs classification, and triplet extraction, respectively, for non-end-to-end setting. The relative improvement for contribution sentence identification, IUs classification, and triplet extraction is 8.08, 2.46, and 2.31 in terms of F1 score for the *NLPContributionGraph* (NCG) dataset. Our system achieves the best performance (57.54% F1 score) in the end-to-end pipeline with all four sub-tasks combined. We make our codes available at: https://github.com/92Komal/pipeline_triplet_extraction.

Keywords Scholarly article · Information extraction · Knowledge graph · Machine learning · Multitask learning

1 Introduction

In recent years, there has been a substantial increase in the availability of scientific articles online, with a significant growth observed over the past decade [34]. Due to this, extracting new scholarly information has become a major challenge for researchers. Given the large volume of publications available, science researchers and others interested in the field often struggle to efficiently navigate through the vast amount of information available to them [11]. In each article, new systems and tasks are introduced as the scientific organizations expand and evolve, and various methodologies are compared. Manual search and analysis of scientific literature can be time-consuming and error-prone. Despite the advancements in search engines, detecting new technologies and their relation to previous ones is still hard [1]. Search engines can return overwhelming results that may not be

✉ Komal Gupta
komal_2021cs16@iitp.ac.in

Ammaar Ahmad
1801cs08@iitp.ac.in

Tirthankar Ghosal
ghosalt@ornl.gov

Asif Ekbal
asif@iitp.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology Patna, Bihta, Patna, Bihar 80116, India

[2] National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA

Published online: 23 January 2024

relevant or accurate [8]. However, scientific paper recommendation systems often require access to a researcher's personal data and browsing history, raising privacy concerns for some users [63]. Traditional databases alone are insufficient for such recommendation systems as they struggle to handle unstructured data and lack the capability to effectively combine information from external sources [49]. Moreover, a knowledge graph provides a structured and comprehensive repository of scientific information [19]. This information can be easily accessed and analyzed by intelligent algorithms. Therefore, intelligent algorithms are needed to extract and organize scientific information from the vast knowledge graph, facilitating quick identification of new technologies and tasks by researchers. Information extraction (IE), such as identifying scientific entities and their relations, is important to organize the data into knowledge bases, including KG. KG provides a way to represent knowledge as a graph of entities and their relation enabling researchers to quickly identify new technologies and tasks by analyzing the connections between different entities within the graph. It is crucial to extract contributions from the research articles for building the KG. Additionally, KG can help understand novelty and concepts by providing a structured representation of existing knowledge and identifying gaps or missing links in the knowledge graph. This can help researchers identify new and unique connections between concepts, leading to the discovery of novel ideas and approaches. Novelty refers to new, original, or previously unknown ideas, while concepts that have not been captured or disregarded may still be old or existing ideas that are overlooked or not given enough attention. So there is an increasing demand for systems that help to extract and organize scientific information from scientific articles and automatically build the KG.

However, the process of building a high-quality SKG highlights several challenges and limitations. First, scientific data are highly heterogeneous, distributed, and often incomplete, making it difficult to integrate and represent in a unified graph structure. Second, extracting knowledge from unstructured text data, such as scientific publications, requires advanced NLP techniques and domain-specific ontologies. Representing and linking entities and relations in the KG requires careful design and curation to ensure accuracy and consistency. Despite the challenges, there have been significant efforts in recent years to build SKG, such as the Semantic Scholar [71], Microsoft Academic Graph [73], and CORD-19 [78] SKG. The limitation of these KGs is the absence of a contribution graph, which would enable the identification of the specific contributions made by research articles. As the number of research publications increases, it will become crucial to extract contributing sentences from scholarly articles and design KG to efficiently represent the knowledge. One such work is *NLPContributionGraph* (NCG) [21], an annotation system for describing academic contributions in NLP articles. The NCG corpus is annotated using this annotation scheme. Its objective is to automate scientific papers annotation to create scholarly contribution graphs across NLP domains. The NCG dataset is annotated for four different challenges: (1) extracting contribution sentences that show significant contributions in the research article. (2). Extract phrases from the contribution sentences. (3). Classification of contribution sentences into IU and (4). Triplet extraction. The available NCG dataset is annotated in the same sequence way, providing a useful resource for these tasks. Our main objective of this work is to extract contributions from the scientific articles and extract scientific terms and relations from the contribution. These terms and relations are used to build the SKG. The SKG allows machines to navigate through prior knowledge in the literature, make meaningful comparisons, understand the novelty of a new research article, etc. The NCG challenges serve as the basis for this paper [22]. In order to address these challenges, we use the SciBERT [10] deep learning model.

In this paper, we propose deep learning-based approaches to solve four problems, *viz.* contribution sentence identification, phrase extraction, information unit classification, and triplet extraction. For this, we propose a neural network-based technique for automatically identifying contribution sentences in research articles. We develop a multitasking deep neural network architecture named *ContriSci*. Multitask learning can help to address the issue of limited training data by leveraging the data from related tasks to improve performance on the primary task [59]. We implement the following two scaffold tasks for *ContriSci* model: (1). *Section identification*, (2). *Citance classification*. Section identification refers to identifying the section headings or labels in a document. The goal is to automatically recognize the hierarchical structure of a document and to identify the headings, such as *introduction, methods, results, experiment* and *abstract*. The task is often approached as a classification problem, where the model is trained on labeled examples to predict the section label of each sentence. We use the *ACL Anthology Sentence Corpus* (AASC)[1] dataset to train the section identification scaffold task. Citance classification is a method for classifying research statements as either citances or non-citances. Citances are statements that reference previously published work, while non-citances do not. In our research, we use citance classification to identify and analyze citances in a large corpus of scientific articles. The citance classification task is trained using the *SciCite* dataset [18]. We use the *BERT-CRF* [67] model to extract phrases from the contribution sentences. The neural network model cannot be adequately trained with only 6,093 training sentences in the NCG dataset. So we use two additional datasets, i.e., *SciERC* [44] and *SciClaim* [47]. The NCG dataset contains annota-

[1] https://kmcs.nii.ac.jp/resource/AASC/AASC.html.

tions for various IUs, namely *ablation analysis, approach, baselines, experimental setup, experiments, hyperparameters, model, research problem, results, task, dataset, and code*. We classify the sentences into the IUs using a *BERT*-based multi-class classifier [88]. Inspired by Liu et al. [39], we reorganize the dataset into five categories, namely *A, B, C, D, E* where each category is defined based on similar syntactic or semantic properties, allowing for more efficient and accurate extraction of relevant triplets. We generate all the possible combinations of the triplets. We implement BERT-based classifiers for *A, B, C, D* types triplets. For the type *E* triplets, we extract the triplets using the rule-based approach. Our main contributions can be summarized as follows:

1. We propose a multitasking system for the identification of contribution statements from research articles with state-of-the-art results. This system can automatically identify and extract the contribution statements from research articles, which can help researchers quickly understand the main contributions of the paper.
2. We built a BERT-CRF-based system for phrase extraction from contribution statements. Our approach exhibits reduced complexity in comparison with the existing models. The system can accurately extract phrases related to contributions from the identified contribution statements.
3. We develop a multi-class BERT-based classifier for information unit classification with state-of-the-art results. The system can classify contribution statements into different information units.
4. We develop a BERT-based system for triplet extraction. The system can organize phrases into triplets with state-of-the-art results.
5. We propose a pipelined-based system for triplet extraction for building the KG with state-of-the-art results. The proposed system can automatically extract triplets to build the KG using the extracted information.

We structure the rest of our paper as follows: In Sect. 2, we provide a detailed description of the dataset used in our research. The related work is discussed in Sect. 3. We define the problem in Sect. 4. The problem is divided into four parts which are *contribution sentence identification, phrase extraction, information unit classification,* and *triplets extraction*. The IUs classification is the subtask of the triplet extraction task. It plays a vital role in the extraction of relevant triplets. Hence, both tasks are jointly discussed in the subsequent sections. In Sect. 5, we explain our dataset preprocessing steps to ensure the quality of our data. Section 6 is dedicated to the system overview in detail. We compare the performance of our proposed model to the baseline model and analyze the results along with addressing dataset annotation anomalies in Sect. 7. Finally, in Sect. 8, we conclude our findings and provide directions for future research.

2 Dataset description

We use the *NLPContributionGraph* (NCG) [21] dataset. The dataset is publicly available in three sets, i.e., *training set*[2], *trial set*[3], and *test set*.[4] The dataset is annotated at three distinct levels. The corpus contains two plain text formats for each article: (1). The PDF is converted to the plain text file using *(GROBID, 2008)*[5] parser. (2). The sentence is transformed into a tokenized form by utilizing Stanza [58]. The dataset is annotated into three levels, as shown in Fig. 1: (1) Contribution sentences, (2) Scientific terms and predicates from contribution sentences, and (3) The triplets *viz. (subject, predicate, and object)*. These triplets are organized into two levels of knowledge [23]. At the top level, there is a placeholder called *Contribution*. Underneath that, there are twelve IUs, encompassing categories like *ablation analysis, approach, baselines, experimental setup, experiments, hyperparameters, model, research problem, results, task, dataset,* and *code*. Scholarly article contributions are categorized under at least three IU nodes, determined by their relevance to the article. The first triplet of each IU includes the *Contribution* subject, which we classify as type E triplets. Figure 1 shows the example of triplets, belonging to the *ExperimentalSetup* IU. Moreover, D'Souza et al. [22] present five general annotation guidelines for identifying contribution sentences in the NCG scheme. (1) Identify sentences that describe or indicate the contribution of the paper, such as introducing a new method or achieving a breakthrough result. (2) Focus on the main contribution of the paper, which is often stated in the introduction or abstract. (3) Annotate sentences that provide evidence or support for the main contribution, such as experiments, results, or analysis. (4) Avoid annotating sentences that describe background knowledge or unrelated information. (5) Consider the context and purpose of the paper when identifying contribution sentences, as the contribution may vary depending on the research question or goal. By following these guidelines, annotators can consistently and systematically identify and annotate contribution sentences. The annotation scheme is evaluated on a dataset of 200 articles, which are purposefully selected from the *ACL Anthology*. Each of the five NLP tasks is represented equally with 40 articles. To ensure the quality of the annotations, two annotators independently annotated every sentence in the dataset, and disagreements were resolved through adjudication by a third annotator. The authors calculate the inter-annotator agreement score using Cohen's Kappa [70]. The obtained results indicated a substantial agreement of 0.75 for sentence-level annotation, indicating that the annotation

[2] https://github.com/ncg-task/training-data.

[3] https://github.com/ncg-task/trial-data.

[4] https://github.com/ncg-task/test-data.

[5] https://github.com/kermitt2/grobid.

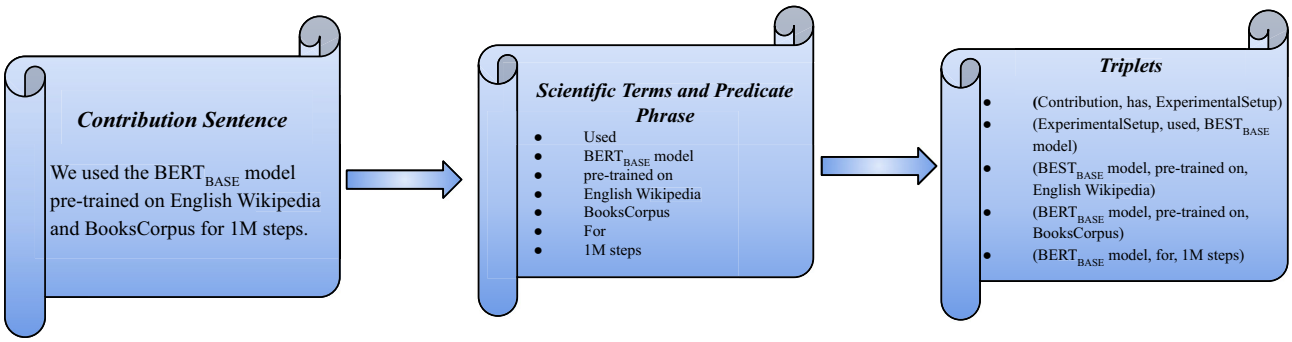

Fig. 1 Structure of NCG dataset

scheme is reliable. The overall objective of these tasks is to build a KG. The structure of the dataset is as follows:

1. The *sentence.txt* file contains the index number of contribution sentences.
2. The *entities.txt* file contains phrases with *paper id, starting index,* and *end index* of the phrases.
3. The *Grobit-out.txt* file contains the plain text of the article.
4. The *Stanza-out.txt* file contains articles' sentences in tokenized form with sentence numbers.
5. The *triplet* folder contains information unit-wise triplets of the papers.
6. The *info-unit* folder contains a *.json* file of information units, each containing respective contribution sentences.

Tables 1, 2, and 3 show the dataset statistics of the *contribution sentences*, *phrases*, and *triplets*, respectively. Due to the limited number of instances in the training set, we combine the trial set with the training set for the training deep learning model. Consequently, our training set encompasses both the original training and trial sets, while the test set evaluates the model's performance. We create a validation set by randomly selecting 10% of the samples from the training set. In Tables 1, 2 the columns *Avg. Length* and *Max. Length* refer to the average and maximum length of sentences in terms of the number of tokens. These metrics are calculated by counting the number of tokens in each sentence and then averaging or taking the maximum across all sentences in the dataset. However, we count the number of sentences per section per document. On average, there are approximately 10–15 sentences.

The evaluation of systems in this task is conducted in three distinct phases. The first phase, known as Evaluation Phase 1, focused on the end-to-end pipeline of the KG building task, testing the systems' ability to construct a KG comprehensively. The second phase, Evaluation Phase 2, is divided into two parts. Part 1 focused solely on the systems' capacity to extract phrases and organize them into triples. Part 2, on the other hand, tested the systems exclusively on their abil-

Table 1 Data statistics of NCG corpus for contribution sentences (CS) and non-contribution sentences (NCS)

Analysis	Train	Test
# Domains	29	10
# Document	287	155
# Sentences	66,397	33,645
# CS	6093	2720
# NCS	60,304	30,925
Avg Length (# tokens)	21.8	20
Max Length (# tokens)	389	377
Avg # sentence per section	39	36
CS having Citation	111	45
NCS having Citation	633	484

Table 2 Data statistics of NCG corpus for phrases

Analysis	Train	Test
# Phrases	35, 262	16, 433
Avg. # Phrases in Sentences	6	6
Avg. # Phrases in Doc	123	106
Max. Length of Phrases (# tokens)	32	18
Avg. Length of Phrases (# tokens)	3	2

Table 3 Data statistics of NCG corpus for triplets

Analysis	Train	Test
# IU in Doc	1,267	642
Avg. # IU in Doc	4.420	4.1419
# Triplets	21,603	10,623
# Unique Triplets	19,512	10,002
Subject	9600	4951
Predicate	5719	2447
Object	15,847	8282

ity to form triples. These three evaluation phases thoroughly assess the systems' performance in different aspects of the KG building tasks.

3 Related work

In this section, we discuss the previous work on identifying contribution sentences, phrase extraction, and triplet extraction. We also survey some of the literature on multitasking techniques.

3.1 Contribution sentence identification

The problem of contribution sentence identification has received very little attention in the past literature. Brack et al. [14] propose a list of generic scientific concepts (such as process, method, material, and data) identified by a rigorous annotation procedure. This set of ideas is used to annotate a corpus of scientific abstracts from ten different fields of knowledge. Furthermore, they suggest the active learning [89] technique for selecting the best instances from diverse data areas. The experimental results indicate that non-experts may reach a significant agreement after consulting with domain specialists. The baseline system has a high F1 score. As part of SemEval 2021 Task 11:NLPContributionGraph [23], Shailabh et al. [64] solve this challenge by constructing a system for a research paper contributions-focused KG over NLP literature. Bidirectional LSTM (BiLSTM) is stacked on top of SciBERT model layers in the first sub-task identifying contribution sentences from the research articles. Liu et al. [39] developed a BERT-based classifier for classifying contribution sentences. The authors also include position features in the classifier. Their system came in second place in the Phase 1 evaluation and first place in both parts of the Phase 2 evaluation. The system produced the best overall results after correcting a submission error in Phase 1. Ma et al. [45] employed a BERT-based system to identify contribution sentences. They utilized a pre-trained BERT model to generate word embeddings, which are 768-dimensional representations, for each word in the sentence. To classify sentences, they put forth a novel approach of using the word embeddings of the first token (i.e., [CLS]) in each sentence. By leveraging the semantic information the [CLS] token captured, their model successfully identified contribution sentences based on their meaning and context. Zhang et al. [84] introduced a framework for extracting sentences. They leveraged sentence context and section heading as additional features and used BERT as a binary classifier. This classifier determines if a sentence provides contribution information. The contribution information is not included in the majority of sentences in an annotated article. As a result, they used a strategy of under-sampling. The positive and negative samples ratio is fixed to an integer for each batch during the training process to guarantee that the model does not overfit negative samples. Martin et al. [50] propose a multi-class sentence classification model with 13 classes. Each of the 12 IUs represents a class and fine-tuned the *deBERTa* [29] base model using sentences from the training dataset to develop a 13 class sentence classification model. Arora et al. [5] proposed a BERT-based classification model to identify the contributing sentences in a research publication. Their approach utilized the BERT pre-trained weights which can support sequences of up to 512-word pieces. The authors addressed the issue of data imbalance between contribution and non-contribution sentences by filtering out most of the non-contribution sentences using simple bi-gram filtering. Their model achieved promising results in identifying the contribution sentences accurately.

Our proposed approach is similar to previous work, such as Shailabh et al. [64], Liu et al. [39], Ma et al. [45], Brack et al. [14], Zhang et al. [84], Martin et al. [50], and Arora et al. [5], which also develop a deep learning BERT-based model for sentence identification. However, our approach distinguishes itself by introducing a unique problem setting. In addition to sentence identification, we incorporate multitasking by including *section identification* and *citance classification* as supporting tasks. By leveraging these additional tasks, we aim to enhance the accuracy of contribution sentence identification in scholarly articles. This comprehensive approach allows us to address the challenges in identifying and extracting meaningful sentences more effectively.

3.2 Phrase extraction

There are some exciting works that focus on extracting phrases from research articles. Liu et al. [39] present a BERT-CRF model to recognize and characterize relevant phrases in contribution sentences. Shailabh et al. [64] used a combination of SciBERT, BiLSTM [30], and CRF for phrase extraction from contribution sentences. Zhang et al. [84] presented a BERT-based model. They trained 10 models by 10-fold cross-validation and used a voting count scheme to extract the phrases from contribution sentences. Ma et al. [45] used a pre-trained BERT model with softmax and argmax top layers, which are shared across all tokens. Martin et al. [50] trained a feature-based Maximum Entropy Markov Model (MEMM) to predict scientific terms in the contribution sentences. Zhu et al. [90] present a BiLSTM model. On top of the BiLSTM, a CRF layer is used to predict the label of sentences. Wang et al. [75] present PTR, a phrase-based topical ranking method for phrase extraction in scientific publications. Zhang et al. [86] proposed a novel deep recurrent neural network (RNN) [56] model to combine the keywords and context information. Alzaidy et al. [3] propose a model that jointly exploits the complementary strengths of CRF layers

that capture the label dependencies and BiLSTM networks that capture the hidden semantics in text. Sahrawat et al. [61] used contextual embeddings to the BiLSTM and CRF model using the BIO[6] labeling scheme.

Phrase extraction also focuses on other domains, including news [72], meeting transcripts [38], and web text [26, 86]. The phrase extraction techniques can be divided into two categories, e.g., supervised learning [55] and unsupervised learning [17]. In supervised learning, train a classification model with some heuristic rules to predict the candidate phrases. They also use some features for this task [31, 51, 81], such as TF-IDF, position feature, and other resource-based features. The unsupervised method is usually formalized as a ranking problem [36]. Phrases are ranked-based on TF-IDF [28, 38, 87], term informativeness [79] and graph-based ranking [52, 72] as well. One approach in the graph-based ranking method involves creating a graph where nodes represent the phrases or sentences in the text, and then ranking the nodes based on their importance. There are some important methods incorporated into the graph to improve the performance, i.e., topic information [12, 13], semantic information from the knowledge base [66, 83], and pre-trained word embeddings [76], [48]. Gupta et al. [27] proposed an approach for describing a research work in terms of focus, application domain, and the techniques used. *FOCUS*: a research article's main contribution. *TECHNIQUE*: a research approach or instrument, such as expectation-maximization and conditional random fields. *DOMAIN*: the application domain of an article, such as *Machine Translation* [57] and *Natural Language Inference* [46]. They use semantic patterns to classify texts from the abstract into the above categories. The purpose of extracting the following concepts from scientific publications is to examine application domains, strategies used to solve domain challenges, and the focus of scientific papers in a community.

Our model shares similarities with the models proposed by Shailabh et al. [64], Zhang et al. [84], and Ma et al. [45]. All of these models utilize a BERT-based model for phrase extraction. Similarly, Zhu et al. [90], and Alzaidy et al. [3] have proposed models that use a CRF layer for phrase extraction, similar to our model. However, the key difference between our proposed model and the models proposed by Shailabh et al. [64], Zhang et al. [84], and Ma et al. [45] lies in the top layer of the BERT-based model. Gupta et al. [27] extract key aspects from the articles by matching semantic extraction patterns learned using bootstrapping [2] to the dependency trees of sentences in an article's abstract. In contrast, our model shares similarities with the model used by Liu et al. [39]. Both models use BERT to encode the input sentences and use CRF to extract relevant phrases. However, our model differs in its approach, as we employ the BERT-CRF model trained on multiple datasets, including NCG, SciClaim [47], and SciERC [44], whereas Liu et al. [39] only trained their model on the NCG dataset. Another difference is that Liu et al. [39] used an ensemble of 96 models for phrase extraction. Overall, our proposed model offers a simpler yet effective approach to phrase extraction compared to the other models mentioned above.

[6] B, I, and O denote the beginning, intermediate, and outside Named Entities (NEs).

3.3 Triplet extraction

There are several interesting studies that focus on extracting triplets from research articles. Rusu et al. [60] present a method for extracting triplets from English phrases in their work. First, four well-known English syntactical parsers are used to generate the parse trees from the phrases and then extract triplets from the parse trees using parser-dependent approaches. Jivani et al. [33] present a method for extracting multiple *subject-object* relations from natural language input, including one or more *subjects, predicates,* and *objects*. The visualization of the parse tree and the dependencies generated from the Stanford Parser [80] is used to extract the information from the given sentence. Jaiswal et al. [32] presented an algorithm with a modified approach for extracting various triplets from text using the Treebank structure and dependencies generated from the Stanford parser on sentences. The KG-Bert [32] used the BERT language model and utilized the entity and relation of a triplet to compute its score. Liu et al. [39] categorize the triplets into different types according to their composition and use separate BERT-based binary classifiers for each type. In their work, Shailabh et al. [64] developed a rule-based methodology for extracting triplets. Their approach involved using a SciBERT-BiLSTM-based binary classifier to identify the predicates. For phrase identification, the preceding phrase was assigned as the subject, while the subsequent phrase was designated as the object of the respective triplet. Zhang et al. [84] and Lin et al. [37] proposed a similar 2-step triplets generation followed by extraction procedure. For triplets generation, they used the combination of all the serial phrases. Then they classify these triplets using a BERT-based model. Martin et al. [50] proposed a rule-based approach using the part-of-speech tags and order of occurrence of phrases in a sentence for triplet extraction. Ma et al. [45] used a binary classifier to classify phrases into subject-predicate-object using the multi-label classifier. They organized these subjects, predicates, and objects into triplets in an iterative manner.

In contrast to the above-mentioned existing model, our approach involves a more in-depth analysis of the dataset and incorporates several modifications to enhance the performance of our model. We exclude the IUs from the training set that contain $< 2\%$ contribution sentences of the training

set. This step is taken to improve the quality of our dataset. After this, we separate the overlapping sentences.

3.4 Multitask learning

Multitask Learning (MTL) is now widely used in NLP tasks to take advantage of the interconnection between the related tasks. Caruana et al. [16] proposed MTL as an inductive transfer method that increases generalization by employing domain information included in related task training signals as an inductive bias. This is achieved by learning tasks in parallel and utilizing a shared representation. What is learned for one task can contribute to the learning of other tasks. Liu et al. [40] suggest three alternative RNN models for sharing information. All linked tasks are combined into a single system that is jointly trained. The first model has a single shared layer for all tasks. The second approach employs multiple layers for various tasks, with each layer able to read data from other layers. The third approach creates a shared layer for all the tasks and assigns one specialized layer to each work. In addition, the authors create a gating technique that allows the model to use shared data selectively. All of these tasks are trained for the entire network at the same time. The complete system is trained jointly on all these tasks. Liu et al. [41] propose an adversarial MTL framework. The framework incorporates orthogonality restrictions in an adversarial multitask setting, where the shared and private feature spaces are fundamentally discontinuous. They create a general shared private learning architecture to model the text sequence. The authors suggest two approaches to prevent interference between the shared and private latent feature spaces: adversarial training and imposing orthogonality requirements. The adversarial training ensures that the shared feature space only contains common and task-invariant data using the orthogonality constraint to remove redundant features from both the private and shared spaces. To incorporate knowledge into citations from the structure of scientific papers, Cohan et al. [18] offer a neural MTL framework. They propose two auxiliary tasks as structural scaffolding to improve citation intent prediction: (1). predicting the section title where the citation appears and (2). predicting if a phrase requires a citation. Unlike the primary objective of citation intent prediction, collecting large amounts of training data for scaffold tasks is simple because labels naturally appear during the writing process. Hence, no manual annotation is required. They show that the suggested neural scaffold model outperforms the existing approaches by a wide margin on two datasets. They classify citation intents based on structural information from research articles. We take advantage of the concept of multitask learning and apply it to identify contribution sentences in research papers automatically.

4 Problem definition

The problem is divided into four parts. Formally, given a scientific document D which consists of a list of n number of sentences $D = [S_1, S_2, \ldots, S_n]$. The problem is defined as:

1. *Contribution sentence identification* Contribution sentences are a set of sentences that show the contribution to the research article. We classify the sentences in a given document D into contribution $C = [c_1, c_2, \ldots, c_m]$ and non-contribution classes, where c_i denotes an ith contribution sentence and m is the total number of contribution sentences in the given document.
2. *Phrase extraction* Suppose C is the list of contribution sentences $C = [c_1, c_2, \ldots, c_m]$ from a given document D. We have to extract the list of the phrases P from the C, where $P = [p_1, p_2, \ldots, p_t]$ is denoted as the list of phrases in the contribution sentences C and each p_k represents one of the phrases extracted from the contribution sentence c_i. This is posed as a sequence learning problem where the task is to identify whether a phrase denotes a scientific term or a predicate phrase. The t represents the total number of phrases in the list of contribution sentences C of a document D. We use BIO tagging format to tag the tokens. BIO tagging format is a common annotation scheme used in NLP to label the entities in text. *BIO* stands for *Beginning, Inside, Outside*. This tagging scheme is used to indicate the position of each token in an entity (e.g., person, organization, or location) in a text.
3. *Information unit classification* IU serves as a way to categorize and organize the contribution triplets based on the content and context of the research article. The annotated contribution triplets of each scholarly article are categorized into at least three or more IUs. These IUs are as follows: *ablation analysis, approach, baselines, experimental setup, experiments, hyperparameters, model, research problem, results, task, dataset,* and *code*. These IUs contain triplets from multiple sentences, formatted in the .json file format. Our goal is to classify contribution sentences into their respective IUs, which will enable us to effectively categorize triplets. The classification of IUs is a crucial step for the triplet extraction task, as it helps us to extract relevant triplets. With the aid of sentence identifiers of contribution sentences, we categorize the triplets into their respective information units, provided that the sentence identifier of the triplet matches that of its corresponding contribution sentence.
4. *Triplet extraction* In this task, we are forming triplets of the phrases that are extracted from the contribution sentences and classifying them into one of the IU denoted as $U = [u_1, ...u_r, \ldots, u_x]$. In the given document, u_r represents one of the twelve information units, and x denotes the total number of information

units. For each u_r, there is a triplet set called $T^r = [(su^r_1, pr^r_1, ob^r_1), (su^r_2, pr^r_2, ob^r_2),\ldots,(su^r_j, pr^r_j, ob^r_j),\ldots, (su^r_O, pr^r_O, ob^r_O)]$, where (su^r_j, pr^r_j, ob^r_j) is a triplet representing the *subject, predicate* and *object*, respectively, where j represents the jth triplet in $u_r \in$ IU and O is the total number of triplets in $u_r \in$ IU. Triplet statements are organized under the information unit.

For example, given a contribution sentence from the *Introduction* IU from the *Machine Translation* domain.

1. *Contribution sentence: The NMT typically consists of two sub-neural networks.*
2. *Phrase extraction:* The*(O)* NMT*(B)* typically*(O)* consists*(B)* of*(I)* two*(B)* sub-neural*(I)* networks*(I)*.
   Phrases:
   NMT *(B)*
   consists of *(B, I)*
   two sub-neural networks *(B, I, I)*
3. *Triplets extraction:*
   Information Unit: Introduction
   Triplets NMT *(subject)*, consists of *(predicate)*, two sub-neural networks *(object)*

5 Dataset pre-processing

The NCG dataset includes three separate file types for each paper. These are: (i). The original paper is in PDF format. (ii). The plain text representation of the PDF is obtained by parsing it with the Grobid PDF parser. (iii). An additional text file containing the paper's tokenized sentences, generated using Stanza [58]. This tokenization helps break down the sentence into its constituent parts, making it easier for machine processing. In the Stanza file, each sentence is assigned a unique sentence number. These sentence numbers provided by Stanza facilitate accurate tracking of the contribution status for each sentence. However, we analyzed the title sentences, and out of the total 287 title sentences, 284 (98.9%) are contributing sentences, while the remaining 3 (1%) are split into two parts and these sentences are annotated as non-contributing. The following pre-processing steps are performed on the combined Stanza file of the training. However, these steps are not applied to the test set.

5.1 Combining incomplete sentences in the stanza file

We observe that the Stanza files in the dataset contain many incomplete sentences, which do not provide proper context, and the baseline model [64] fails to identify those incomplete sentences. As an example, under the topic *Paraphrase Generation*, paper number: 0, The following two lines are incorrectly terminated due to special symbols replaced by '?' in the *Stanza* file.

1. *164: The Critical Difference (CD) for Nemenyi test depends upon the given ?*
2. *165: (confidence level, which is 0.05 in our case) for average ranks and N (number of tested datasets).*

We join the sentences that terminate with any of the following symbols "?", ";", "?:", ",". Also, some sentences break on the citation. For example in *Natural Language Inference* paper number: *58* Stanza file,

1. *63. Chen et al.*
2. *64. propose using a bilinear term similarity function to calculate attention scores with pre-trained word embeddings.*

We also combine these types of sentences. We conduct manual sentence verification to identify and correct any instances of incorrect sentence combinations.

5.2 Extraction of main section and sub-section titles

We label the NCG dataset for section identification by extracting the section names to which a sentence belongs. Additionally, we extract the subtitles of the sentences to provide extra context. Rule-based heuristics are implemented for their identification using Grobid and Stanza files.

1. If the sentence length is $<= 4$ tokens and it contains a substring like *Abstract*, *Introduction*, *Related Work*, *Experiment*, *Implementation*, *Background*, etc. It is the main heading of the section.
2. Statements succeeding blank lines in the Grobid files are recognized as potential section titles and subtitles. The subsequent conditions are examined within these sentences:
   (a) Based on the analysis of the dataset, we find that sentences serve as main headings if their length is < 10 tokens and there is a substring (length > 2) of the paper title present in a sentence that does not end with any of the English stop words.
   (b) If the preceding criteria *(2a)* are not satisfied, the sentence is considered a subheading, and all such sentences are recognized subheadings if they do not terminate with a stopword like [*by*, *as*, *in*, *and*, *that*] nor if they consist only of digits and periods like "2.", "4.1.".

5.3 Extracting previous and next sentence

We concatenate the previous and following sentences of the current sentence to the input representation in addition to the original sentence and subsection to provide more context to the model. The previous sentence is blank if the sentence is the first sentence of the subsection. Similarly, if the sentence is the last one of the subsection, then the following sentence is blank.

6 System overview

In this section, we describe the methodologies for *contribution sentence identification, phrase extraction*, and *triplet extraction*. We show the architectural diagram with the corresponding task. Out of the various versions of the BERT model such as SciBERT [10], DistilBERT [62], we chose to use SciBERT for each task because it has been specifically pre-trained on scientific text, which is the type of text that we are dealing with in this task. Additionally, SciBERT has been shown to outperform general-purpose BERT models on several NLP tasks related to scientific text, such as citation intent classification [18] and scientific named entity recognition [69]. Therefore, we believe that SciBERT is a suitable choice for our task of contribution sentence identification, phrase extraction, information unit classification, and triplet extraction. DistilBERT is also a smaller and faster version of BERT, but it may not provide the same level of performance as SciBERT on scientific text [6].

6.1 Contribution sentences identification

We build a multitask model (ContriSci) to determine whether the sentence of the research article describes a contribution. After pre-processing, we analyze the dataset again.

6.1.1 Data analysis after pre-processing

After pre-processing, we analyze the training set and find that the number of contribution sentences distributed across sections (such as *Related Work*, *Background*, *Previous Work*, *Future Work*, *Conclusion*, or *Discussion*) is negligible. Therefore, we remove the sentences belonging to these sections from the dataset.

1. *Analysis of contribution sentences in sections* Figure 2a shows the distribution of all the sentences, and Fig. 2b shows the distribution of contribution sentences across sections. The *Experiment* section consists of most of the contribution sentences, followed by the *Result* and *Introduction* sections. Around 20% of sentences in the *Experiment* section are contribution sentences, whereas only 5% of sentences in the *Method* section are contribution sentences. This asymmetric distribution aids us in identifying contribution sentences.
2. *Analysis of contribution sentences having citation* We analyze the cited sentences in the training set. The analysis of the cited sentences is shown in Figs. 3a and b. The total number of non-cited sentences in the dataset is 46,980, whereas there are 538 cited sentences. Only 109 of the 538 sentences are contribution sentences, accounting for less than 2% of the total contribution phrases. As a result, the number of cited contribution sentences is negligible across the dataset.

6.1.2 Data for scaffold tasks

We make use of the additional data for enhancing the weight of the scaffold tasks.

- *Section Identification:* We utilize the *ACL Anthology Sentence Corpus (AASC)*[7] dataset for section identification. The corpus originates from the scientific domain. The dataset constitutes a substantial corpus of sentences encompassing ten labeled sections, comprising over 2 million annotated instances. We use a subset of this corpus having 75K samples across five sections in the following class distribution: *abstract(8.6%)*, *introduction(20.3%)*, *result(20.9%)*, *background(7.3%)*, and *method(16.1%)*. Since there is no *experiment* section in the AASC dataset and the distribution of contribution sentences in *result* and *experiment* sections is the same. We combine the section label for these sentences into the *result* section, as we are only interested in the distribution of the sentences for the scaffold task. In the dataset, the average length of the sentence is 20 (in token).
- *Citance Classification:* The SciCite [18] dataset is used for the *citance classification* scaffold task. This dataset contains 73K sentences from publications. Sentences containing citances are categorized as positive instances, while those without citances are categorized as negative instances. The ratio of positive-to-negative samples is approximately 1:6.

6.1.3 Methodology

We propose a BERT-based multitask learning (ContriSci) model for extracting contribution sentences from the research articles. This multitask model has two scaffold tasks related to the structure of research articles. These scaffold tasks help the main task in identifying contribution sentences. We train

[7] https://kmcs.nii.ac.jp/resource/AASC/AASC.html.

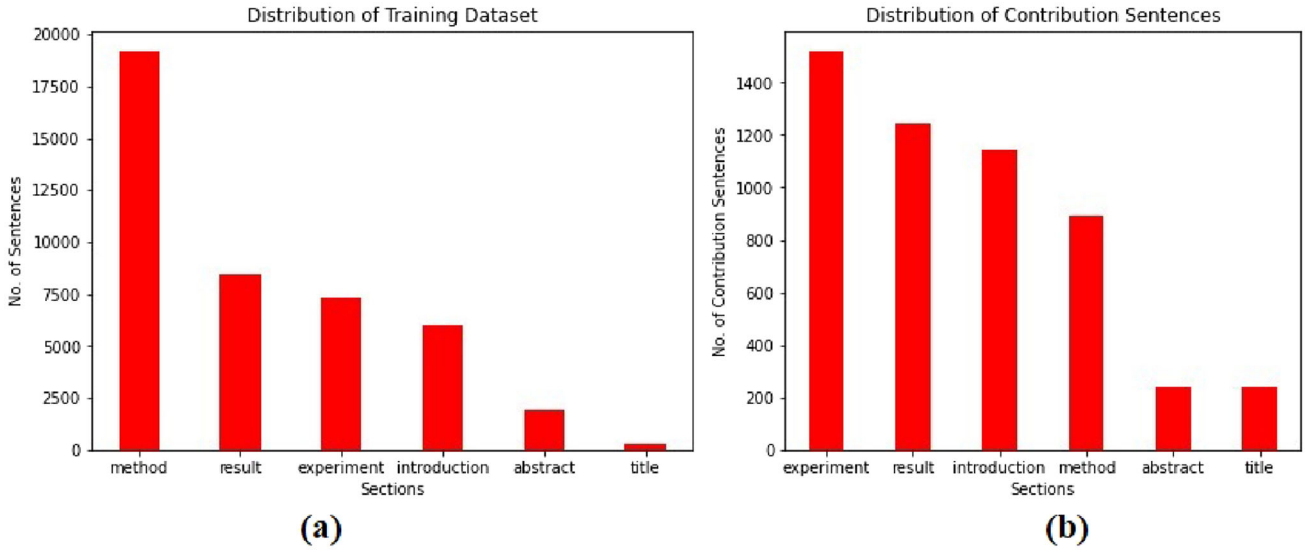

Fig. 2 Analysis graph of NCG training dataset. The training set consists of the combination of the original training and trial sets provided in the dataset. Figures *a* and *b* show the analysis of the distribution of all the sentences and contribution sentences in the section

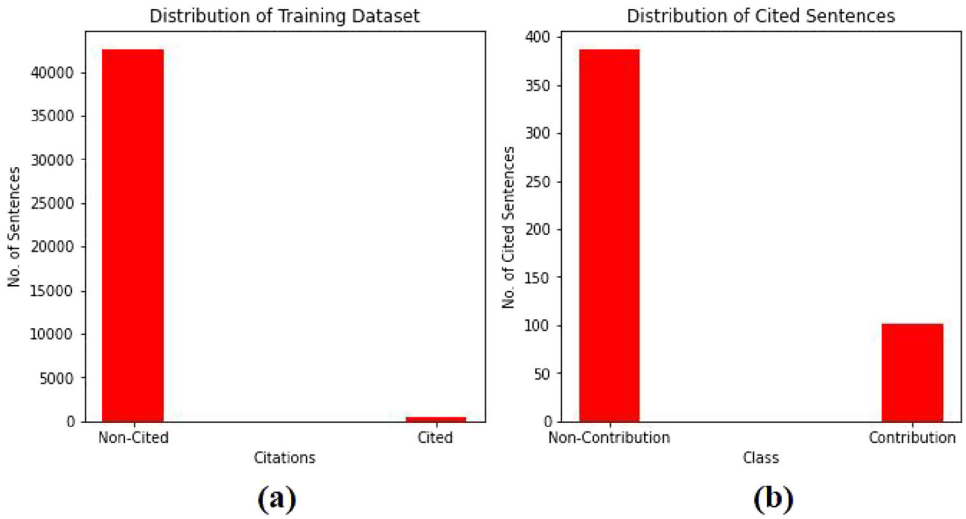

Fig. 3 Analysis graph of NCG data (training + trial) set. Figures *a* and *b* show the analysis of the cited sentences and non-cited sentences

both the scaffold tasks with NCG data as well as the additional data, i.e., *ACL Anthology Sentence Corpus (AASC)* and *SciCite* [18] dataset.

1. *ContriSci model* Figure 4 shows the architecture of the *ContriSci* model. All the tasks share the *SciBERT* [10] layer. We present inputs to the model in two ways. If the sentence belongs to the NCG dataset, it is in the form of *Current Sentence* + # + *Subheading* + # + *Previous Sentence* + # + *Next Sentence*, whereas, if the sentence belongs to the scaffold dataset, it is a single sentence. We tokenize the input and set the maximum length to 256. If the length exceeds the maximum length, we truncate those inputs from the right-hand side to the maximum length and add padding tokens to the shorter sentences to match the maximum length.

Section Identification: The first scaffold task is the identification of section headings of the sentences. The semantic structure and distribution of contribution sentences vary across different sections. In the *Introduction* section, contribution sentences primarily outline the research problem, while the *Experiments* section's sentences describe the methodology used in the article. Section identification aids the identification of contribution sentences by learning differences in linguistic patterns of sentences across sections.

Citance Classification: The second scaffold task is to classify whether or not a sentence includes a citation. The primary purpose of this scaffold task is to distinguish between cited and non-cited sentences. In the research

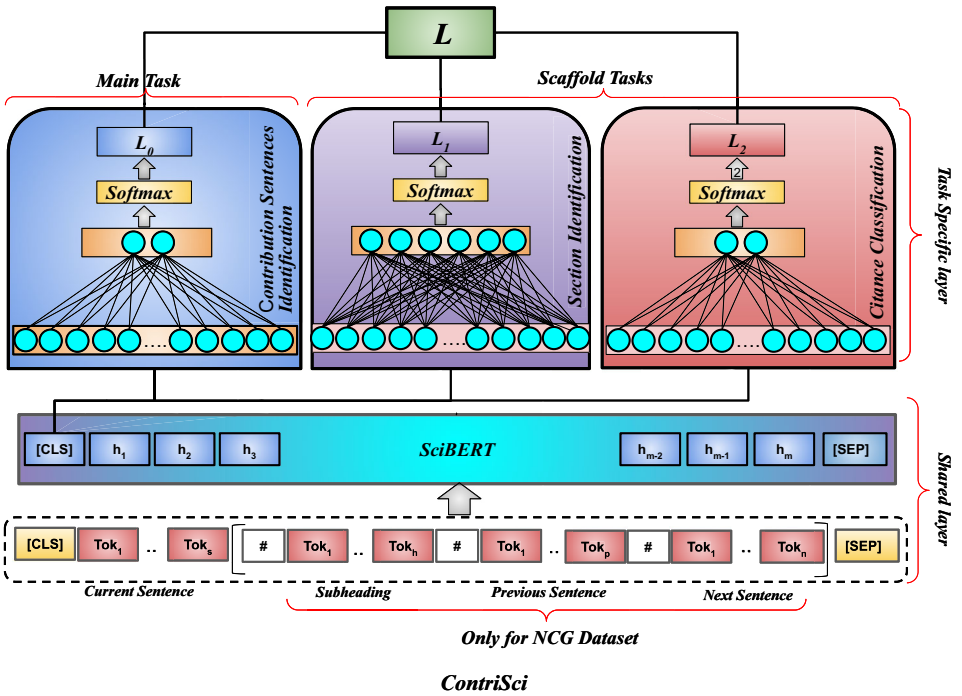

Fig. 4 Architecture of the *ContriSci*. Here, *ContriSci* model aims to identify contribution sentences. The main task is predicting contribution sentences, and two scaffolds predict the section title (section identification) and predict citation in the sentences (citance classification)

article, there are about 5% of cited sentences [24]. The number of cited contribution sentences among such sentences is minimal. This information is quite useful in distinguishing between contribution and non-contribution sentences.

2. *ContriSci architecture description* MTL [16] enhances the performance compared to the single-task setting by transferring knowledge from the related tasks. It has some parameters shared across all the tasks. In our ContriSci model, T_0 represents the main task in multitask learning, accompanied by (n-1) supplementary tasks T_i, where n = 3. We utilize a task-specific Multi-Layer Perceptron (MLP) layer for each task, with a Softmax layer positioned on top. Specifically, we take the SciBERT [CLS] output vector x and input it into n MLPs, followed by softmax to calculate the prediction probabilities for each task.

$$y^{(i)} = softmax(MLP^{(i)}(x)) \tag{1}$$

We focus on the main task output, denoted as $y^{(0)}$. The remaining output ($y^{(1)}$, $y^{(2)}$) is exclusively utilized during training to enhance the performance of the main task.

3. *Training procedure* We train the multitask model following the steps outlined in Algorithm 1. Each dataset (*NCG*, *SciCite*, and *AASC*) has its own data loader. A batch is sequentially selected from either of the three data loaders in a 3:5:5 ratio. The ratio is according to the size of each dataset. We cannot randomly build batches because of the task-specific layer train with the respective dataset. In the NCG dataset's batch-wise training process, we use the dataset batches to train both the main and scaffold tasks. On the other hand, the AASC dataset batches are used to train only the section identification task, and the SciCite dataset batches are used to train only the citation classification task. Batches are selected sequentially from the three data loaders in a ratio of 3:5:5, corresponding to each dataset's size.
4. *Loss* We use the categorical weighted cross-entropy loss for each of the tasks. Cross-entropy is defined as:

$$L = -\sum_{i=0}^{n-1} w_i t_i \log(P_i) \tag{2}$$

where n is a number of batches, w_i are class weights, t_i is truth label, and P_i is a softmax probability of i^{th}. Thus, the overall loss function is a linear combination of loss for each task defined by:

$$L = L_0 + \lambda_1 * L_1 + \lambda_2 * L_2 \tag{3}$$

Here, L_0, L_1, L_2 is a loss for the main task, section identification task, and citance classification task, respectively. The λ_1 and λ_2 are the hyperparameters. Each class is assigned equal weightage for L_1 loss since each label in the AASC dataset has an equal number of examples. The distribution of cited and non-cited sentences in the SciCite dataset is 1:6. So for L_2 loss, we set the class weight as the inverse ratio of the number of examples in each class in

the citance classification scaffold dataset. Finally, for the main loss L_0, we set the class weight as a hyperparameter.

Algorithm 1 Multi-task learning for contribution sentence identification

```
procedure MLTContriSci(D0, D1, D2)
    D0, D1, D2 are distinct data loaders for the three datasets.
    D0: NCG Dataset, D1: AASC Dataset, D2: SciCite Dataset
    Each loader processing mini-batches of size 16.
    for epoch in 1, 2, ..., epoch_max do
        N = length(D0)/3;
        for i in range(N) do
            Model train with 3 batches of D0
            Compute L0(θ), L1(θ), L2(θ)
            Compute Gradient grad
            Update Model (θ = θ − α ∗ grad)
            Model train with 5 batches of D1
            Compute L1(θ)
            Compute Gradient grad
            Update Model (θ = θ − α ∗ grad)
            Model train with 5 batches of D2
            Compute L2(θ)
            Compute Gradient grad
            Update Model (θ = θ − α ∗ grad)
        end for
    end for
end procedure
```

6.2 Phrase extraction

In this task, we extract relevant phrases from contribution sentences, which are essential for extracting triplets. However, this can be challenging as it requires identifying and extracting phrases that accurately denote entities and their relations. In the NCG training set, there are phrases of different lengths, and the number of training samples in the dataset is 6,093 only, which is insufficient to train the neural network model. Hence, we use two additional datasets *viz. SciERC* [44], *SciClaim* [47]. We utilize the *BERT-CRF* model to extract phrases from the contribution sentences. The architecture of the phrase extraction model is shown in Fig. 5. The model receives input sentences from the NCG dataset and additional datasets.

1. *SciERC dataset* The *SciERC* [44] dataset includes annotations for scientific entities, their relations, and coreference clusters. This dataset is annotated for 500 scientific abstracts. These abstracts are taken from 12 AI conference proceedings in four AI communities from the *Semantic Scholar* Corpus. *SciERC* expands existing datasets in scientific articles of *SemEval 17* [7], and *SemEval 18* [25] by adding entity and relation types. The *SciERC* dataset is paragraph label annotation. We split the paragraph into sentences, but the label remains the same and we check the correctness of the sentences manually. The entire dataset comprises a total of 2,382 sentences and 5,238 phrases. The average number of phrases per sentence is 2.19.
2. *SciClaim dataset* The *SciClaim* dataset is annotated by Magnusson et al. [47], involving 12,738 annotations across 901 sentences. These sentences were identified as expert-identified claims in SBS [4] papers, causal language in PubMed [82] papers, and claims and causal language heuristically discovered from CORD-19 [74] abstracts. The *Sciclaim* dataset has been labeled using the BIO annotation scheme on a sentence level to identify the beginning, inside, and outside of entities within each sentence. However, due to accessibility constraints, we can use a subset of 512 sentences for our experiments. The dataset contains 3,498 phrases, and on average, each sentence contains 6.19 phrases.

6.2.1 Methodology

We use BERT-CRF [68] model to extract the phrases from the contribution sentences. Figure 5 shows the proposed model. The BERT-based model is efficient for contextual representations of sentences. We use the CRF layer to identify the scientific terms from the contribution sentences. The CRF layer leverages the contextual information from the surrounding context to assign labels to tokens within a sequence.

$$y^{(i)} = CRF(FC^{(i)}(x)) \tag{4}$$

The fully connected layer is between the BERT output and the CRF layer, where x is the input and y is the model's output.

6.3 Information unit classification and triplet extraction

Here, we classify contribution sentences into one of the IUs. To form triplets, we initially separate predicates and non-predicates. Subsequently, we generate all possible combinations of the triplets and classify them as valid or invalid according to their respective type: *A, B, C,* or *D*. Rules are applied for type *E* triplets.

6.3.1 IU classification

This task aims to categorize the contribution sentences into one of the IUs. In the NCG dataset, contribution sentences are annotated in IU, which aims to extract the most important information units in scientific papers related to NLP tasks and classify the contribution sentences. Figure 6 illustrates the multi-class classification model for IUs classification.
Before analysis During our analysis of the NCG training set, we noticed that certain IUs are highly infrequent. Due to

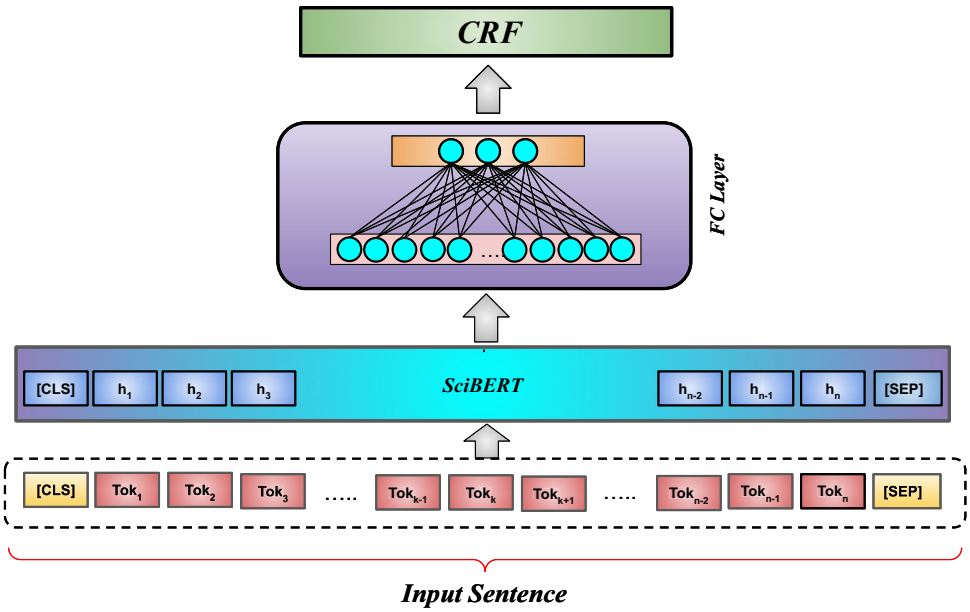

Fig. 5 Proposed phrase extraction architecture

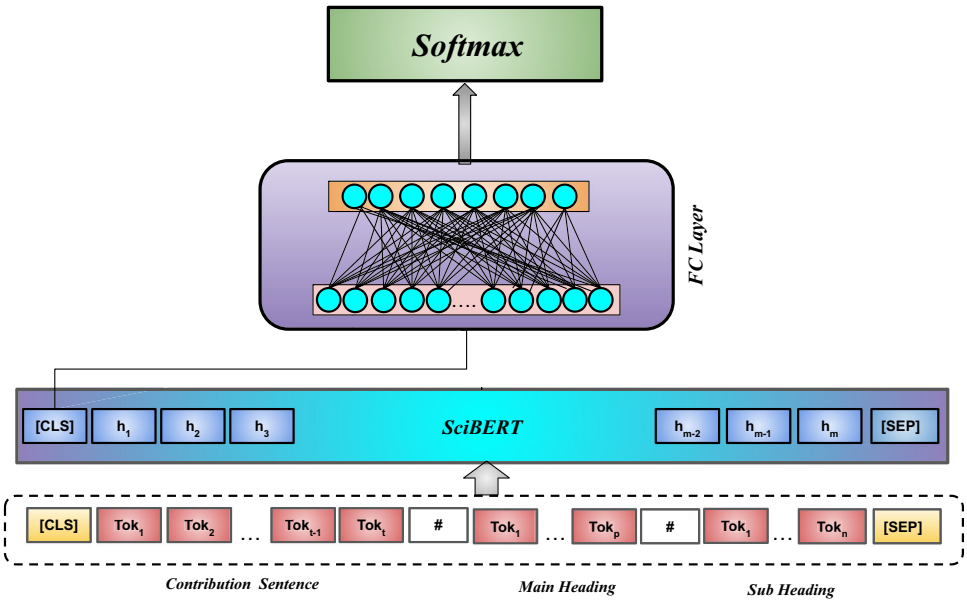

Fig. 6 Multi-class classification architecture for IU

infrequent occurrences of IUs in the research article, these IUs consist of a low number of contribution sentences, as shown in Fig. 7. To enhance the effectiveness of our IU classification model, we opted to exclude IUs that comprise $< 2\%$ of the contribution sentences of the training set. As a result, we eliminated the *Task, Code,* and *Dataset* IUs from consideration. For the remaining nine IUs, we initially train the BERT sequence multi-class classifier with nine classes. We achieved the F1 score of 82.02% on the test dataset. We observe that the sentences containing IU *Code* can be identified using a simple rule-based approach, as they are characterized by the presence of a *Uniform Resource Locator* (URL) in their content.

After analysis After analyzing the results, we find that the IUs *Experimental-Setup* and *Hyperparameters* have many sentences overlapping. In the NCG training set, there is no instance of a single paper containing both *Experimental-Setup* and *Hyperparameters* information units. The decision of which unit to choose is made at the document level. Therefore, we merge these two labels under the name *hyper-setup* and train the multi-class classifier with eight classes, namely *research-problem, model, approach, experiments, results, hyper-setup, baselines, ablation-analysis*. After classification, we train another BERT-based binary classifier to classify sentences with *hyper-setup* labels into *experimental-setup* and *hyperparameters*. The input to the binary classifier is an entire paragraph instead of sentences. We achieve an F1

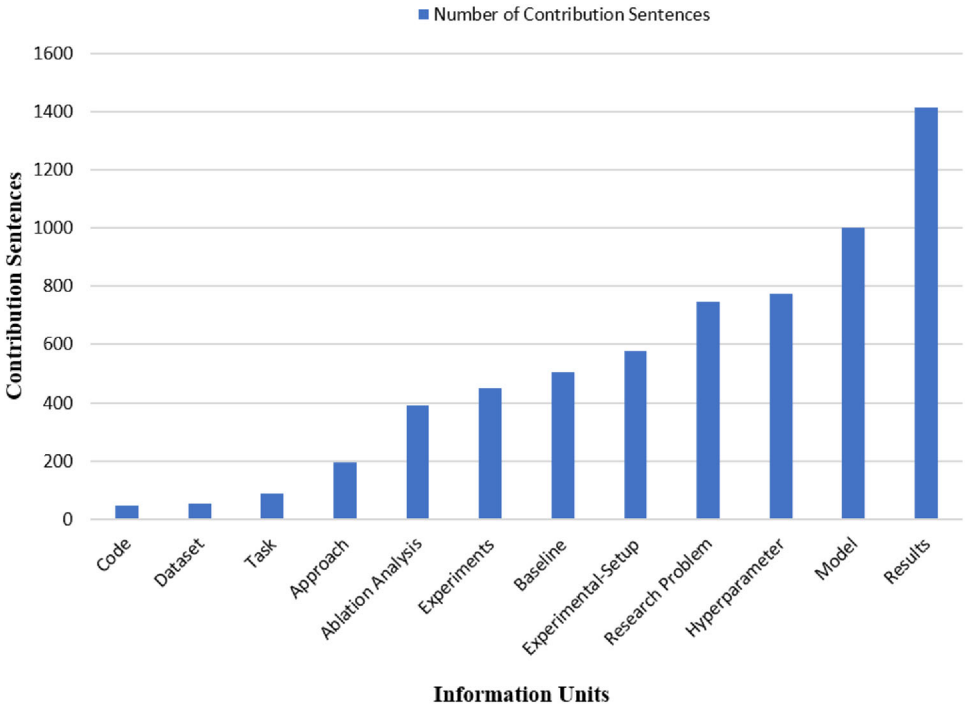

Fig. 7 Distribution of Contribution Sentences in Information Units

score of 78.82% using this binary classifier. After classifying overlapping sentences using a binary classifier, we achieve an overall F1 score of 84.52%, a 2.5% improvement in the IU classification model.

6.3.2 Predicate classification

To generate triplets from sentences, we need to understand the differences between *subject*, *object*, and *predicate* phrases. We know *predicate* is unique in triplets. On this basis, the *predicate* can be distinguished from the *subject* and *object*. To classify the *predicate*, we create a BERT-based binary classifier. In this BERT-based classifier, one indicates a *predicate*, while zero indicates a *non-predicate*. An F1 score of 93% is achieved when phrases are input into the binary classifier. Introducing a sentence with a phrase marker into the binary classifier results in an even higher F1 score of 98%. An example of an input sentence with a phrase marker is as follows:

We also present a version of our model that uses a << character LSTM >>, which performs better than other lexical representations even if word embeddings are removed from the model.

6.3.3 Triplet extraction

From the phrase extraction model, the discrete phrases are obtained. We have to organize these phrases. The phrases have to be formed as a triplet (subject, predicate, and object). Liu et al. [39] categorize the triplets according to their composition in order to achieve a deeper understanding of the properties of the triplets. The triplets categorization is shown in Table 4. We explain each type of triplet using an example.

1. *Type A* In type *A*, triplets occur in the same sentence, and these triplets have the highest number among all other triplet types. We generate all the possible triplets of type A and train the classifier. Let there be *m* number of *predicates* and *n* number of *subjects/objects*, then the total number of triplets generated per sentence *[m*n*(n-1)/2]*. The total number of generated triplets of type *A* is 157K in the NCG training set. The following is the input example for the *A* type triplets classifier:
   All models are implemented using [[TensorFlow 3]] and << trained on >> the [[SQUAD training set]] using the ADAM optimizer with a mini-batch size of 4 and trained using 10 asynchronous training threads on a single machine.
2. *Type B* In the type *B* triplets, the *subject* and *object* come under the sentence and use *'has'* as the *predicate*. We generate all possible triplets of type *B*, in which the subject comes before the object in the sentence. Let there be *n* number of *subjects/objects*, then *[n*(n-1)/2]* triplets are generated per sentence. The following is the input example for the *B* type classifier:
   We can see that our << transfer learning approach >> [[consistently improved]] over the non-transfer results.
3. *Type C* The sentence is linked to its IU in a type *C* triplets, so the *subject* is always the *information unit*, and the *predicate* and *object* are under the sentence. If the number of

Table 4 The triplet types, their respective example, and the frequency of the triplets in the dataset. To address type *A* to *D*, BERT-based binary classifiers are used, while rules are used to address Type *E*. Here *S, O,* and *P* represent the subject, predicate, and object, respectively

	Composition	Triplets	# Triplets
Type A	All three phrases belongs to sentence itself	*S:-different contexts* *P:-in the form of* *O:-Location, Caption, and Part of Speech tags*	10,646
Type B	Subject and Object in a sentence related by *"has"* predicate	*S:-Adagrad optimizer* *P:-has* *O:-optimizing algorithm*	1,110
Type C	Information Unit connected to the object by a predicate present in a sentence	*S:-approach* *P:-consider* *O:-different contexts*	1,790
Type D	Information Unit connected to the object present in a sentence by *"has"* predicate	*S:-baselines* *P:-has* *O:-NQG*	1,546
Type E	Subject = *Contribution* Predicate = *has*, Object = *Code* or *has research problem*	*S:-Contribution* *P:-has research problem* *O:-Visual Question Generation*	1,449

predicates in a sentence is *m* and the *object* is *n*, the total number of triplets generated per sentence *[m * n]*. The following is the input example for the *C* type classifier:
[[approach]]: Our method implicitly << uses >> a [[differential context]] obtained through supporting and contrasting exemplars to obtain a differentiable embedding.

4. *Type D* Triplets of type *D* are similar to type *C*, there is always an *information unit* in the *subject*, and the *predicate* also does not come under the sentence. They include the non-sentence *predicate* word *'has'*. If there are *n* objects in a sentence, the total number of triplets generated per sentence *n*. The following is the input example for the *D* type classifier:
[[hyperparameters]]: Parameter optimization is performed with [[mini batch stochastic gradient descent (SGD)]] with batch size 10 and momentum 0.9.

For *A, B, C, D* types of triplet, we construct a unique classifier and validate them. We extract type *E* triplets by a rule-based approach. In this triplet type, the subject is always a *Contribution*, and the predicate can be *has research problem, Code, has*. If the predicate is *has research problem*—it covers all triplets belonging to the IU *Research Problem*. The object is the phrase extracted from the sentence belonging to the *Research Problem* IU. If the predicate is *Code*—it covers all triplets belonging to IU *Code*, and the object is the URL extracted from the sentence. If the predicate is *has* this triplet type is the link between paper and IU. For example, if a paper has at least one sentence belonging to *Results* IU, then the triplet *(Contribution || has || Results)* is added to the paper. It signifies the presence of the particular IU in that paper. We achieve an F1 score of 1.00 for type *E* triplets using these rules. Approximately 3% of the triplets identified in the dataset belong to more than one sentence, known as cross-sentence triplets. Our work does not involve categorizing these types of triplets.

7 Evaluation

In this section, we discuss the experiment setup, baseline model, results, error analysis, annotation anomalies, and ablation analysis.

7.1 Experimental setup

We implement our proposed *ContriSci* model with (*SciBERT* has word piece vocabulary (scivocab)) $allenai/scibert\_scivocab\_uncased$.[8] Each task has its own *multilayer perceptron* (MLP) [85] layer, which consists of a fully connected dimension (number of neurons) of 768 layers followed by a classification layer. The batch size is set to 16, and the *AdamW* [43] optimizer is used to train the model. On the fully connected layer and classification layer, we use the activation functions [65] *Tanh* and *Softmax*, respectively. Using the *PyTorch* [9] framework. In trial experiments, we used epochs = 2, 3, and 4. We found that the best validation F1

[8] https://huggingface.co/allenai/.

score is achieved with epochs = 2, and further training leads to overfitting. So we set epochs = 2 while tuning other hyperparameters. We are tuning the following hyperparameters for the best model: *Learning rate(α), λ_1, λ_2, dropout,* and *class weights*. We use α between the range [1e-6 and 2e-5] and using *dropout* [54] of 0.1 and 0.2. λ_1: weightage for section identification loss varied between [0 and 0.3]. λ_2: weightage for citance classification loss varied between [0 and 0.3]. In the main task, the loss weights for each class ranged from [0.5 to 0.88]. The *Contrisci* model has 5 hyperparameters for tuning. Due to a large number of combinations of the variable, we apply the random search algorithm. On the following parameters, the *ContriSci* model performs the best: $\lambda_1 - 0.18$, $\alpha - 1e - 5$, $\lambda_2 - 0.09$, *dropout* - 0.2, and *class weights* - 0.75. We get the best result, when we input the sentences in the order *(Current Sentence + # + Subheading + # + Previous Sentence + # + Next Sentence)* in *ContriSci* model.

We implement the *SciBERT-CRF*-based phrase extraction model using the *PyTorch* framework. We initialize the *SciBERT* layers with pretrained weights from *allenai/scibert_scivocab_uncased*. On the top of *SciBERT*, there are fully connected layers followed by CRF. These weights are initialized randomly. The batch size is set to 1. We use *AdamW* [43] optimizer, *Tanh* activation function, and *linear* scheduler to train the model for five epochs. We use a grid search algorithm to tune learning rates and dropout hyperparameters. *SciBERT* learning rate is varied between [5e-6, 1e-5, 2e-5, 5e-5], and the learning rate for the remaining layers is varied between [1e-4, 2e-4], and dropout is varied between [0.1, 0.2]. The best model has the following hyperparameters: SciBERT learning rate is 2e-5, the learning rate for remaining layers is 1e-4, and dropout is 0.2.

We use SciBERT for sequence classification[9] model for IU classification and triplet extraction. In each model, we use the *Tanh* activation function in between layers and *Softmax* activation function in the final layer. We train the model for ten epochs with a learning rate of 1e-5 and a dropout of 0.1. We use *AdamW* optimizer and *polynomial decay* [42] scheduler with the number of warmup steps set to 500 and decay power of 0.5.

7.2 Baseline

In this section, we explore the top-performing models for each task. We compare our proposed models with these models.

[9] https://huggingface.co/docs/transformers/model_doc/bert#dtatransformers.BertForSequenceClassification.

7.2.1 ContriSci model

We compare our model with the baseline as proposed in Shailabh et al. [64] and Liu et al. [39]. Shailabh et al. [64] used the pre-trained *SciBERT* with BiLSTM [30] as a sentence-level binary classifier. Sentences of the Stanza file are input into the *SciBERT* model. The last layer of *SciBERT* is the input into the stacked *BiLSTM* layers. Liu et al. [39] present a SciBERT-based binary sentence classifier with features to handle the sentence characteristics. They also process the topmost and innermost section header and position in the articles.

7.2.2 Phrase extraction

We compare our model with the baseline Shailabh et al. [64] model. They add a BiLSTM layer on top of the *SciBERT*. They use the CRF layer on the top of the SciBERT+BiLSTM and use the BILOU scheme to mark the boundary of the phrases.

7.2.3 Information units classification and triplet extraction

Liu et al. [39] use the BERT-based multi-class classifier for IU classification task. They merged the two special pairs (*Model* vs *Approach* and *Experimental-Setup* vs *Hyperparameters*) in the multi-class classifier. After classification, they used lexical rules to differentiate among these units. Liu et al. [39] classified the triplets into different types depending on whether and how their components were expressed in text and then validate each type triplet using independent BERT-based classifiers and a rule-based approach. Instead of developing traditional neural network open information extraction (OIE) architectures for the triplets extraction, ECNUICA et al. [37] constructed potential triplets using manually developed rules and developed a binary classifier to distinguish positive from the negative ones. Zhang et al. [84] developed a BERT-based binary classifier for true or false candidates by forming all feasible triplets candidates from the classified scientific phrases. Binary classifier down samples negative candidate triplets by artificially producing them with random replacement (RR) of one of the actual triplets arguments with a false argument and random selection (RS) of triplets with no argument that is a valid pair of another. They also use the *adversarial training* approach.

7.3 Results and analysis

We compare the performance of our proposed models to the SemEval 2021 results.

We evaluate the performance of our model and compare it with the results of the existing model on the CodaLab leader-

board system.[10] Our team name is *IITP* on the leaderboard. To ensure the robustness of our models, we conduct significance tests aimed at comprehensively evaluating the ability of the models to generalize their performance. Through five separate experiments, we use Welch's t test [77] with a significance level of 5% (0.05). With unbalanced classification datasets having unequal group sample sizes, Welch's t test is a better fit compared to the other significance test [53]. We conduct tests to ensure the normality of the data, as this is a prerequisite for this test. The objective is to demonstrate that the improved F1 score achieved by our proposed approach is not a random occurrence but rather statistically significant.

7.3.1 ContriSci model

In Table 5, the initial three results are taken from the leaderboard of SemEval 2021. The first comparison involves the *SciBERT+BiLSTM* model proposed by Shailabh et al. [64]. They utilized the *BiLSTM* layer on top of *SciBERT*. The highest achieved F1 score for this model is 46.80%. The second comparison involves the *SciBERT + Positional feature* [39], which achieves an F1 score of 57.27%. The third model achieves a score of 59.41%. Our multitask model surpasses all these previous state-of-the-art scores by 4.8%, achieving the F1 score of 64.21%. Figure 8a shows the precision, recall, and F1 score of the main task, i.e., identification of the contribution sentences across different sections. In the NCG training dataset, most of the titles of the paper correspond to the contribution sentences, and hence the title section has *recall* 1.0. The sentences in the *Method* section are highly skewed (1:19) toward the non-contribution class. This could potentially be attributed as one of the underlying causes for the inability of our model to effectively discriminate between the contribution and non-contribution sentences within this particular section. Figure 8b shows the results of cited sentences indicating that such kinds of sentences have a higher F1 score of around 0.7. Therefore, *citance classification* scaffold task plays an important role. As a result, we achieve statistical significance compared to the existing models with p-values of 0.024, 0.015, and 0.009.

7.3.2 Phrases extraction

We utilize a BERT-CRF model to extract scientific phrases and their relations. The NCG dataset contains the total of 6,093 training samples. Additionally, we incorporate the *SciERC* and *SciClaim* datasets from the NLP domain as additional data for our model. We achieve F1 score is 77.47%, with the best model scoring 78.57%, as presented in Table 6. We are only 1.1% behind the best model. When we compare our model with the existing models, then the complexity of those models is much more than our model. For instance, Liu et al.'s [39] model, which is regarded as the top phrase extraction model, uses an ensemble of 96 models. In contrast, we use *SciBERT* [10] coupled with additional datasets. The simplicity of our model stands out, as it significantly reduces complexity compared to other models. Our model is the best in terms of complexity. We are not conducting any significant tests since our phrase extraction model is not outperforming.

[10] https://competitions.codalab.org/competitions.

Table 5 Results on NCG test set. The table also shows the comparison of the proposed *ContriSci* model with the top-performing models' results reported in the SemEval 2021 competition. Here, P: Precision, and R: Recall. Results of the task using gold-standard annotations available for CS identification task. The reported results are found to be statistically significant ($p < 0.05$) based on a t test [35]

Models	F1	P	R
KnowGraph@IITK [64]	0.4680	0.3669	0.5701
UIUC BioNLP [39]	0.5727	0.5361	0.6146
Malteos [15]	0.5941	0.5519	0.6433
Proposed Model(ContriSci)	**0.6421**	0.5943	0.6943

Table 6 Results of phrase extraction model. Results of the task using gold-standard annotations available for phrase extraction task

Models	F1	P	R
UIUC BioNLP [39]	0.7857	0.7686	0.8035
ECNUICA [37]	0.7774	0.7655	0.7896
ITNLP [84]	0.7843	0.7795	0.7891
KnowGraph@IITK [64]	0.7452	0.7292	0.7619
Proposed Model (Phrase Extraction)	0.7747	0.7564	0.7939

7.3.3 Information units classification and triplet extraction

First, we classify the predicate for the triplet extraction model using a BERT-based binary classifier. We achieve the F1 score of 98% using this classifier. We use a BERT-based multiclass classifier for IU classification. As shown in Table 7, our proposed model for the IU classification is 2.03% ahead in the F1 score compared to the existing best model. The F1 score of our IU classifier is 84.52%. We obtain statistical significance compared to the existing models with p-values of 0.029, 0.018, and 0.015.

Table 8 compares our results with the triplet extraction performance. The first result in the table is *UIUC BioNLP* [39]. They divide the triplet into six categories and create the neural network BERT-based six different classifiers for each triplet type. They achieve the F1 score of 61.29%. The second result in the table is *ECNUICA* [37]. Based on the scientific term sequence order in the sentences, *ECNUICA* formed triplet candidates. Then, the triplets are classified as true or false candidates using a BERT-based binary classi-

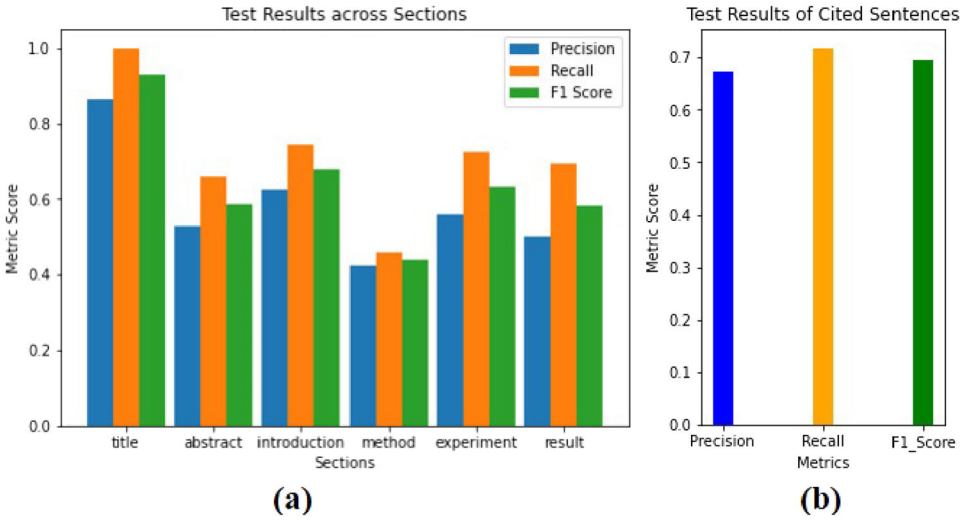

Fig. 8 **a** Evaluation results w.r.t F1, Precision, Recall for section-wise contribution sentence identification in *ContriSci* model. **b** Evaluation results w.r.t. F1, Precision, Recall for the identification of contributed cited sentences in *ContriSci* model

Table 7 Results of IU classification model. Results of the task using gold-standard annotations available for IU classification task. The reported results are found to be statistically significant ($p < 0.05$) based on a t test [35]

Models	F1	P	R
UIUC BioNLP [39]	0.8249	0.7684	0.8902
ECNUICA [37]	0.8108	0.7282	0.9146
ITNLP [84]	0.7640	0.7083	0.8293
Proposed Model (IU)	0.8452	0.8256	0.8659

Table 8 Results of Triplet Extraction (IU + Triplets) model. Results of the task using gold-standard annotations available for Triplet Extraction task. The reported results are found to be statistically significant ($p < 0.05$) based on a t test [35]

Models	F1	P	R
UIUC BioNLP [39]	0.6129	0.6519	0.5782
ECNUICA [37]	0.4473	0.4920	0.4100
ITNLP [84]	0.4082	0.4468	0.3757
Proposed Model (Triplet Extraction)	0.6271	0.6724	0.5874

fier. They achieve the F1 score of 44.73%. The third result in the table is *ITNLP* [84]. Using some rule-based approach, they formed all the possible triplets from the classified scientific terms and then classified them using a BERT-based classifier. They achieve the F1 score of 40.82%. Our triplet extraction proposed model outperforms all the comparing models with the F1 score of 62.71%. *UIUC BioNLP* [39] applies the BERT-based multi-class classifier for IU classification. One of the drawbacks of their model is overlapping between (experimental-setup vs. hyperparameter) and (model vs. approach) pairs. We process overlapping sentences. Initially, we classify the IUs with the BERT-based multi-class classifier. After classifying them into one of the 8-class multi-class classifiers, we reclassified the *hyper-setup* IU sentences. The *hyper-setup* IU is a merged label of the *experimental setup* and *hyperparameter* IU pair, so we classified these sentences with the BERT-based binary classifier. We feed the paragraph as input to the classifier. Our classifier achieves the F1 score of 84.52%, an increment of 2.03% from the previous state-of-the-art performance. As a result, compared to competitive models, we achieve statistical significance with p-values of 0.037, 0.039, and 0.023.

Figure 9 shows the confusion matrix of the IUs classification. There is considerable overlap between approach and model IUs because most of these sentences belong to the *Introduction* section of the paper and hence have a similar semantic structure. Another pair (experiments vs. results) also shows overlap with 59% sentences belonging to the *experiments* IU predicted as results. We tried to resolve these issues using independent binary classifiers for both pairs. These binary classifier performances are worse than the multi-class classifier. So we decided to stick with the multi-class classification model. Although our model classifies 9 classes, we compare our proposed model fairly and the existing results. For the classification of *Code* IU's sentence, we use a rule-based approach to handle these sentences successfully. Additionally, it is important to note that *Dataset* IU sentences are misclassified and assigned to the model IU, as shown in Fig. 9. Despite this, our model's performance continues to outperform the established benchmarks. Furthermore, the absence of the *Task* IU from the test set is reflected in the confusion matrix. We believe our model's performance remains comparable with the existing models, due to the availability of conclusive results for twelve individual IUs.

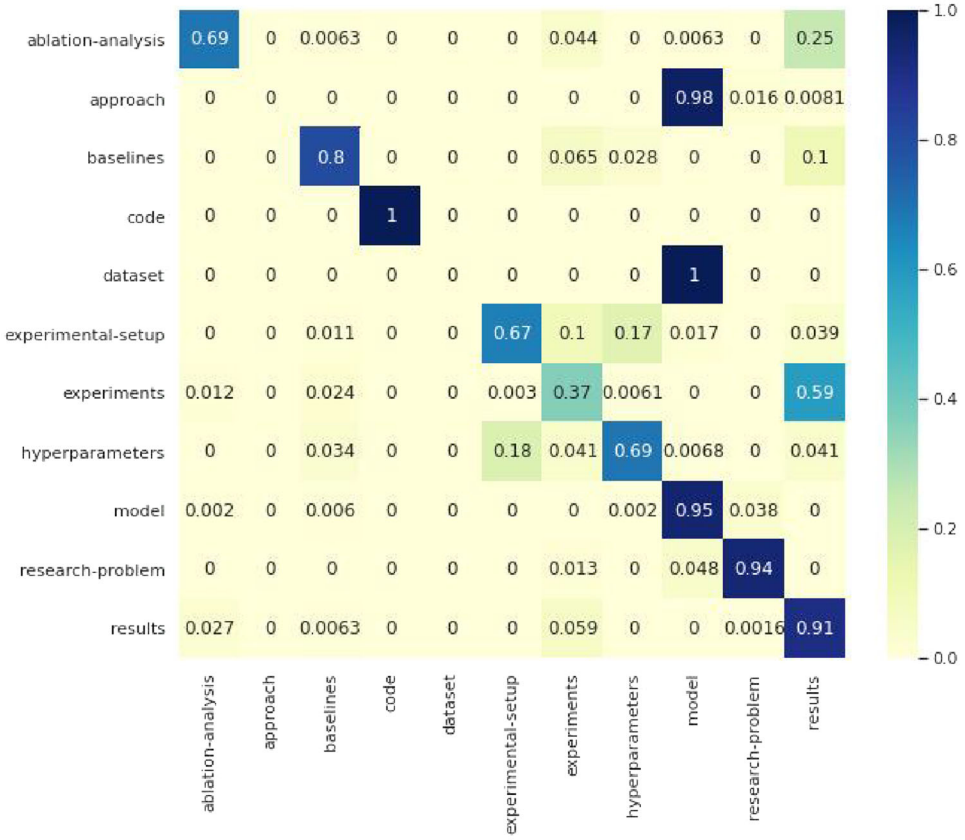

Fig. 9 Confusion matrix for IU classification

7.3.4 Pipeline results

In the pipeline, four models are connected sequentially. We only provide the positive examples obtained from the previous model as inputs to the next model. Table 9 shows the comparison of our end-to-end model performance with the inter-annotator agreement (IAA) [20] and Liu et al. [39] model on each subtask. Although we observe that our system performance for contribution sentence identification is lower than human performance (64.21% vs. 67.44% F1), for the phrase extraction, our model outperforms the previous best Liu et al. [39] (51.30% vs. 46.41% F1) in the pipeline results. For the IU classification, our model outperforms the human annotators (80.00% vs. 79.73% F1). Triplet extraction achieves the best results compared to all existing models (34.63% vs. 22.28% F1). Even our overall pipeline F1 score is 4% ahead of IAA. Our overall pipeline F1 score is 57.54%. However, we also assess the impact of imbalanced document distribution in two sub-tests by selecting 10 evenly distributed articles and another 10 unevenly distributed articles. The number of sentences, unevenly distributed in the articles, is 2,387, and in the evenly distributed articles is 2,330. The model is tested using both sets, and the difference in F1 score is only 0.005, which is negligible. Therefore, we can conclude that the imbalanced distribution did not significantly affect the model's performance.

7.4 Error analysis

Although our proposed models outperform existing approaches, they still have some errors. This section is dedicated to discussing the specific errors encountered in our proposed methods.

7.4.1 Error analysis in ContriSci model

Table 10 shows the misclassified sentences by the ContriSci model. Among these examples, 1 and 2 are classified as *false negatives*, while examples 3 and 4 are identified as *false positives*. Example 1 is identified as a contribution sentence, but it is erroneously classified as a non-contribution sentence. The maximum length of input sequences for SciBERT is 512 tokens, but we chose to use a maximum sequence length of 256 tokens to ensure that the model would fit in the available memory of their GPU. However, this choice of sequence length may affect the model's ability to learn from longer sentences. In our experiments, we observed that our proposed model is not able to effectively learn from longer sentences, and we attribute this to the sequence length limitation of

Table 9 Results of end-to-end model. IAA: inter-annotator agreement

	Avg F1	Information Unit			Sentence			Phrases			Triplets		
		F1	P	Recall	F1	P	Recall	F1	P	Recall	F1	P	Recall
Our system	**0.5754**	**0.8000**	0.7527	0.8537	0.6421	0.5973	0.6943	**0.5130**	0.4581	0.5830	**0.3463**	0.3323	0.3615
UIUC	0.4972	0.7293	0.6667	0.8049	0.5727	0.5361	0.6146	0.4641	0.4269	0.5083	0.2228	0.2230	0.2226
IAA	0.5282	0.7973	0.7883	0.8065	**0.6744**	0.6725	0.6763	0.4184	0.4536	0.3883	0.2228	0.2376	0.2097

Table 10 Error analysis in our proposed model *(ContriSci)*

S.No	Misclassified Sentences
1	The key attributes of our approach are the following: (1) to jointly predict short and long answers in a single model rather than using a pipeline approach, (2) to split each document into multiple training instances by using overlapping windows of tokens, like in the original BERT model for the SQuAD task, (3) to aggressively down sample the null instances (i.e., instances without an answer) at training time to create a balanced training set, (4) to use the " [CLS] " token at training time to predict null instances and rank spans at inference time by the difference between the span score and the " [CLS] " score
2	(1) Part of Speech Tagging (2) Evaluation on FDDB Database
3	Therefore, we propose two different methods for building this subset and we call them sense vocabulary compression methods
4	The model was implemented using Python and Theano

Table 11 The scaffold task is analyzed separately and compared with the proposed model. Where PM—Proposed Model, CC—Citance Classification, SI—Section Identification

S. no.	Misclassified sentences	PM-(CC+SI)	PM-CC	PM-SI	PM
1	Language model pretraining has recently been shown to provide significant performance gains for a range of challenging language understanding problems	Yes	Yes	Yes	No
2	(2) Larger layer size, hidden state dimension, and beam size have little impact on the performance ; our setting, L = 2, H = 200, and B = 5 looks adequate in terms of speed/performance trade-off	Yes	No	No	No
3	An ensemble of 5 LSTM+ A models further improves this score to 92.8	Yes	No	Yes	No
4	The computational complexity of this network is bounded to be no more than twice that of one convolution block	Yes	Yes	No	No

SciBERT. While it is a known limitation of the SciBERT model, we acknowledge that it may impact the performance of our proposed model on tasks that require the understanding of longer sentences. The given sentence has about 100 tokens, and the sentence is also divided into periods, in which scientific words are less, and these words are not relevant to the proposed model of the respective paper. In example 2, smaller sentences usually subheadings, our model cannot correctly classify due to lack of contributing contextual information in these subtitles. In example 3, the sentence given in this example is *False Positive* because the word *we propose* has come in this sentence, so it has been declared as a contribution sentence. However, we are not getting any necessary information about the paper from this sentence, so it is a non-contribution sentence. In example 4, this is also a non-contribution sentence. The reason for its misclassification is that it gives information about the model, but the contribution of this sentence in the article is significantly less. The sentence, whether the contribution in the article is less or more, the proposed model is not known well.

In Table 11, we conduct a separate analysis of the scaffold task. In the first example, a basic *SciBERT* model and a multitask model with a single scaffold incorrectly predict it as non-contribution. In contrast, our proposed model accurately makes the correct prediction. The basic *SciBERT* model struggles to correctly classify the second example. In the table, the third and fourth examples, one of the models with a single scaffold makes an incorrect prediction. In general, a

Table 12 Error analysis in our proposed model (phrase extraction)

S.No	Ground Truth	Predicted
1	: 0.2 (discriminative) : 0.3 (generative)	: 0.2 (discriminative) : and 0.3 (generative)
2	: Our bidirectional : transformer architecture	: Our : bidirectional tran sformer architecture
3	: adversarial and virtual : adversarial training	: adversarial and virtual adversarial : training (0.159-0.331)

model with *section identification* scaffold performs better on sentences with more numerical information, as evident by the second and third examples. Our proposed model *ContriSci* correctly classified all the examples shown in Table 11.

7.4.2 Error analysis in phrase extraction model

The frequent error that occurs in our proposed model is combining two phrases into one, as seen in Table 12 and example no 1. The second error is illustrated in Table 12, the model is unsure whether or not to include starting pronouns and adverbs in a single phrase. Therefore the phrases are split into two parts. The model could not learn which kind of information in parenthesis belongs to a phrase and which is not. Our model predicts the text in parenthesis as part of a phrase in Table 12 and example no 2.

7.4.3 Error analysis in triplet extraction model

In Table 13, examples 3 and 4 are *false positive* and *false negative*. After analyzing both sentences, it is found that the model is not learning semantics appropriately. The model is confused because the word *instead of* is in this sentence. In example 5 our model fails to recognize the relation between subject and object separated by words like *which, are, is, that, can*. Example 5 is a *false negative*. For every training sentence, we generate all the possible combinations of the triplet. We get consecutive triplets and non-consecutive triplets, in which a total of 97% of non-consecutive triplets are non-valid. Hence, a significant majority of the non-consecutive triplets contribute to the category of *false negatives*.

7.5 Annotation anomalies

In this section, we explore the anomalies found in the annotation of the NCG dataset.

7.5.1 Annotation anomalies in contribution sentences

When we analyze the NCG training set and identify numerous annotation anomalies in the contribution sentence identification training dataset. In Table 14, we describe some anomalies of the NCG dataset.

7.5.2 Annotation anomalies in phrases

Table 15 shows two sentences, each containing identical phrases. However, while the phrase in the first sentence lacks a citation, the phrase in the second sentence includes one. This inconsistency in the annotation within the NCG dataset highlights a notable issue. Another type of anomaly in the NCG training set is shown in Table 15. Conventionally, the *predicate* acts as a relation between *subject* and *object*. Hence the sequence of triplets we have always seen *subject predicate* then *object* as far as we know. The instance in Table 15 is extracted from paper number 69 within the *Natural Language Inference* domain of the NCG training set. In this example, we can observe that the triplet does not adhere to the standard sequence. Additionally, the annotated label index of the *predicate* is demonstrated in Table 15. Total 99 such sentences are found in the NCG training set. In these sentences, the sequence is of *predicate subject* and *object*. In comparison, the triplet sequence of this sentence should be as in example 2.

7.6 Ablation analysis

In this section, we present a discussion of the ablation analysis conducted on our proposed models.

7.6.1 Analysis of ContriSci model

We show ablation studies of our proposed model for the identification of contribution sentences on the NCG testing set. In the NCG training set, the length of 93.91% of the training input sentences is < 128, while the length of 99.8% of the training input sentences is < 256. Table 16 shows the results on input sequences 128 and 256 in *ContriSci* Model. When training our model, we conducted experiments both with and without surrounding sentences as well as without scaffold tasks. The comparison revealed a significant difference in F1 scores of 2.40%. The model utilizing surrounding sentences achieved an F1 score of 58.16%, while the model without surrounding sentences obtained an F1 score of 55.76%, with both evaluations performed on 128 tokens. The model's performance is boosted by both scaffold tasks: *section identification* and *citance classification*. As a result, there is a slight improvement when the maximum input length is set to 256.

Table 13 Error analysis in proposed model *(Triplet Extraction)*

S.No	Model Prediction	Example
1	[False Negative]	The conversion accuracy is better for nouns (? 50 % error), and [[much better]] for determiners (30%) particles (6%) << with respect to >> the [[Collins head rules]]
2	[False Positive]	As often demonstrated in the NMT literature, using subword split as input token unit instead of << standard tokenized word unit >> has [[potential]] to improve the performance
3	[False Negative]	As often demonstrated in the NMT literature, using subword split as << input token unit >> instead of standard tokenized word unit has [[potential]] to improve the performance
4	[False Negative]	Importantly, the RNN performance is significantly better than that of the << Avg baseline >>, which [[barely improves]] over mention—ranking, even with oracle history

Table 14 Annotation Anomalies in NCG Training Set

S.No	Annotation Anomaly	Example
1	Citation Removed in Stanza and Grobid File	**Annotated Sentence:** Attentive neural networks have recently demonstrated success in a wide range of tasks ranging from handwriting synthesis, digit classification, machine translation, image captioning, speech recognition and sentence summarization, to geometric reasoning
	Explanation : When we compared the sentences in PDF file with Grobid and Stanza file, we noticed that some sentences have been removed. In the example 1 sentence one is taken from Grobid, and sentence two is taken from the original sentence PDF. All the citations (such as (graves 2013) and (mhil 2014)) are not in sentence one.	**Original sentence:** Attentive neural networks have recently demonstrated success in a wide range of tasks ranging from hand writing synthesis (Graves, 2013), digit classification (Mnih et al., 2014), machine translation (Bahdanau et al., 2015), image captioning (Xu et al., 2015), speech recognition (Chorowski et al.,2015) and sentence summarization (Rush et al., 2015), to geometric reasoning (Vinyals et al., 2015)
2	Citation Break the Sentence into Two Parts	**Annotated Sentence:** (1) For example, Yu et al. (2)used CNN representations as feature inputs to a logistic regression model
	Explanation The sentence in example 2 is taken from the Stanza file. This sentence is divided into two parts by the dot of citation in the Stanza file. Similarly, there are many sentences in which citation is there. In the Stanza, the file is divided into two parts.	**Original sentence:** For example, Yu et al. [36] used CNN representations as feature inputs to a logistic regression model
3	Sentences are Break with Question Mark	**Annotated Sentence:** (1) Furthermore, let e L? (2) R L be a vector of 1s and h N be the last output vector after the premise and hypothesis were processed by the two LSTMs, respectively
	Explanation : In some sentences in Grobid, special symbols have changed into question marks, and in the Stanza file, the sentences in which the question mark has come in the middle, those sentences are divided into two parts from there. Example 3 shows two parts of the same sentence and the original sentence below	**Original sentence:** Furthermore, let e $L \in R$ L be a vector of 1s and h N be the last output vector after the premise and hypothesis were processed by the two LSTMs, respectively
4	Some Sentences are Wrongly Annotated	On SQuAD, our single model obtains an exact match (EM) score of 79.5 % and F1 score of 86.6%, while our ensemble model further boosts the result to 82.3% and 88.5%, respectively
	Explanation: In example 4, the sentence given in example 4 tells about the result of the Single Model and Ensemble Model and shows how much improvement is in Ensemble Model, so this sentence has a high probability of being a contribution. However, in the NCG dataset, it is the non-contribution is annotated	
5	Issues in Length of < 4 Sentences	(1) Sudoku

Table 14 continued

S.No	Annotation Anomaly	Example
	Explanation : In the NCG dataset, the length of less than 4 sentences in length is wrongly annotated. In example 5, the first two contribution sentences have only word, then how will the model recognize from a single word that this sentence contributes to the article and the last three sentences are abbreviations. If the model does not even know the full form of these short-form keys, how will the model recognize whether it is a contribution sentence or a non-contribution sentence?	(2) Subtask A
		(3) NQG *(abbreviation)*
		(4) ATSA *(abbreviation)*
		(5) s 2 s+ att *(abbreviation)*
6	Inconsistency in Labeling the same Subtitles in different Articles:	In *Natural Language Inference* Paper number 10, Line 139 Label is one, but in Natural Language Inference Paper 60, Line 93 Label is zero
	Explanation : The subheading *Natural Language Inference* exists in a total of six papers. Four of them have been annotated as one, and two of them have been annotated as zero	
7	Section Title Removed and some Sentences are Jumbled	The order of sentences is incorrect in the Experiment, Document Modeling section of the Grobid file of *Natural language Inference* papers number 18 in the NCG dataset and Introduction heading in Natural language Inference paper number 21
	Explanation : Section titles are missing in the Grobid file of some articles. That is why our section identification scaffold task is mispredicting the sections of some sentences. Apart from this, in some sections in some Grobid files, there are jumbled sentences in the section; if compared with PDF, then the order of sentences in the Grobid file is wrong	

Table 15 Phrases Annotation Anomalies in NCG Training Set

1Sentences	Phrase
All experiments use the << **Adam optimizer** >> (Kingma and Ba, 2015) with gradient norms clipped at 5.0	Adam optimizer
Training is performed using the << **Adam optimizer (Kingma and Ba, 2015)** >> with a learning rate of 10 ?3	Adam optimizer (Kingma and Ba, 2015)
2Sentences	**Label Index**
The Document—cue baseline can predict more than a third of the samples correctly, << **for** >> both datasets, even after sub-sampling **[[frequent document - answer pairs]]** for **[[WIKIHOP]]**	85 to 88
The Document—cue baseline can predict more than a third of the samples correctly, for both datasets, even after sub-sampling **[[frequent document - answer pairs]]** << **for** >> **[[WIKIHOP]]**	161 to 164

7.6.2 Ablation analysis of phrase extraction

Table 17 presents the ablation analysis of our phrase extraction model using additional datasets. When utilizing the NCG dataset solely as input, we achieve an F1 score of 76.28%. Similarly, when incorporating either the *SciClaim* or *SciERC* dataset, the resulting F1 scores are 76.70% and 76.72%, respectively. Moreover, the Recall value with the *SciClaim* dataset is 79.18%, showing its higher performance compared to the *SciERC* dataset. Conversely, Precision is higher with the *SciERC* dataset, at 75.62%, indicating a complementary relation between the two datasets. When both additional datasets are combined as input, we achieve the highest F1 score of 77.47%.

7.6.3 Ablation analysis of triplet extraction

We conduct an analysis of triplets on IU-wise basis, as presented in Table 18, and categorize the analysis based on types, as illustrated in Fig. 10. All the triplets of type E belong to *Research Problem*, and *Code* IU is extracted by the rule-based approach and achieves an F1 score of 100%. The *Experiments*

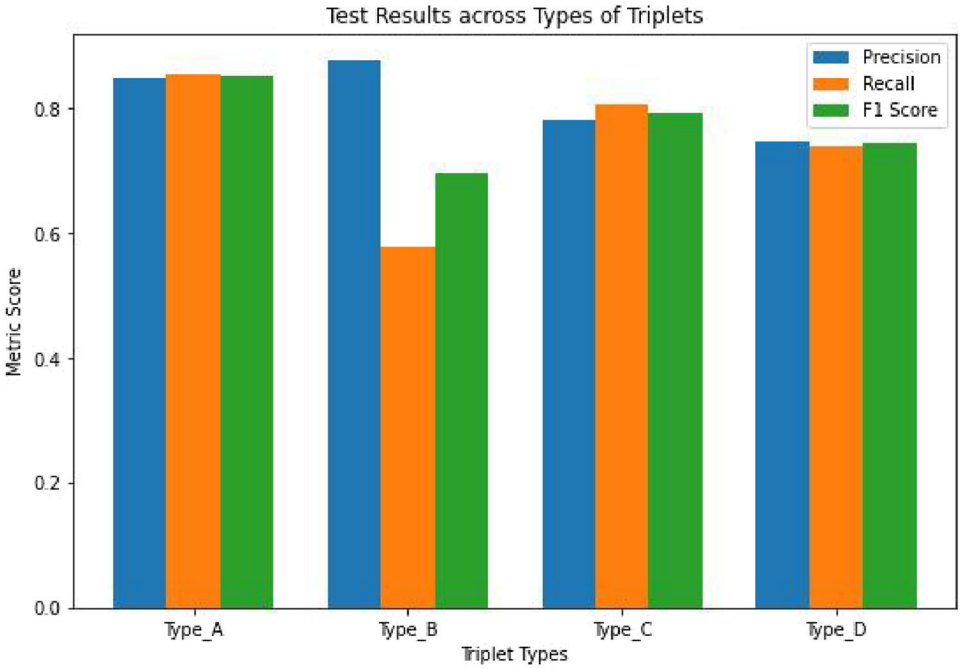

Fig. 10 Typewise (A-D) triplets results

Table 16 Performance of *ContriSci* model along with individual scaffold tasks when the model is trained with 128 and 256 tokens, respectively

Model(128 Tokens)	F1	P	R
Proposed Model—Both Scaffold Tasks	0.5816	0.5629	0.6014
Proposed Model—Section Identification	0.5989	0.5917	0.6063
Proposed Model—Citance Classification	0.5996	0.5862	0.6136
Proposed Model	0.6327	0.5977	0.6720
Model(256 Tokens)	F1	P	R
Proposed Model—Both Scaffold Tasks	0.5694	0.5199	0.6294
Proposed Model—Section Identification	0.5998	0.5534	0.6548
Proposed Model—Citance Classification	0.6052	0.5799	0.6327
Proposed Model	**0.6421**	0.5943	0.6943

Table 17 Performance of *Phrase Extraction* model along with individual dataset

Models	F1	P	R
NCG	0.7628	0.7498	0.7768
NCG + SciERC	0.7672	0.7562	0.7784
NCG + SciClaim	0.7670	0.7437	0.7918
NCG + SciERC + SciClaim	0.7747	0.7564	0.7939

IU has 1,273 triplets in the test dataset out of which in this distribution in types (A = 546, B = 284, C = 3, D = 4, E = 32, Others = 404). Approximately 31.74% of triplets do not align with any specific category and remain unextracted. Triplets belonging to Class B make up 22.30% of the

Table 18 The Triplets IU-wise Results

Information Unit	F1	P	R
Research Problem	1.0000	1.0000	1.0000
Approach	0.7919	0.8550	0.7375
Model	0.8757	0.8980	0.8545
Code	1.0000	1.0000	1.0000
Dataset	0.0000	0.0000	0.0000
Experimental Setup	0.8091	0.8412	0.7794
Hyperparameters	0.7595	0.8333	0.6977
Baselines	0.7957	0.9024	0.7115
Results	0.7461	0.8038	0.6961
Experiments	0.5909	0.8125	0.4643
Ablation Analysis	0.7273	0.8235	0.6512

total, and they exhibit the lowest performance, as indicated in Fig. 10. Due to these reasons, our experiment IU result for triplets is 59.09%. In 77% of Type B triplets, the *object* is immediately followed by the *subject*, (without any phrase in between). As a result, our model struggles to capture the long-term relationship between the *subject* and *object*.

8 Conclusion and future work

In this paper, we propose a pipeline neural network model for extracting the NLP contribution from scientific articles. The proposed model is divided into four tasks: (1) identification of the contribution sentences, (2) phrase extraction from

contribution sentences, (3) IU classification, and (4) Triplet extraction. We introduce a multitasking architecture with two supporting tasks for the identification of contribution sentences—*Section Identification* and *Citance Classification*. Our multitask model achieves an F1 score of 64.21%. We utilize a BERT-CRF model for the phrase extraction task and achieve an F1 score of 77.47%. To classify the IUs, we propose a multi-class classifier based on BERT. Additionally, we develop a binary classifier to distinguish between the *hyperparameters* and *experimental-setup* IUs. Our IUs classification model achieves an F1 score of 84.52%. Finally, for triplet extraction, we achieved an F1 score of 62.71%. We achieve state-of-the-art results in contribution sentence identification, IU classification, and triplet extraction tasks. We obtain state-of-the-art results in the end-to-end pipeline, achieving an F1 score of 57.54%. Our phrase extraction model performance is not good. In the future, we will improve the performance of our phrase extraction model. Another exciting work would be to add new scaffold tasks in our *ContriSci* model for further performance improvement. Moreover, we will improve the performance of the IU classifier by classifying the *Model* and *Approach* IU conflicting sentences.

References

1. Abbas, A., Zhang, L., Khan, S.U.: A literature review on the state-of-the-art in patent analysis. World Patent Inf. **37**, 3–13 (2014)
2. Abney, S.: Bootstrapping. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 360–367 (2002)
3. Al-Zaidy, R.A., Caragea, C., Giles, C.L.: Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In: Liu, L., White, R.W., Mantrach, A., et al. (eds.), The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019. ACM, pp. 2551–2557. https://doi.org/10.1145/3308558.3313642 (2019)
4. Alipourfard, N., Arendt, B., Benjamin, D.J., et al.: Systematizing confidence in open research and evidence (score) (2021)
5. Arora, H., Ghosal, T., Kumar, S., et al.: INNOVATORS at semeval-2021 task-11: A dependency parsing and bert-based model for extracting contribution knowledge from scientific papers. In: Palmer, A., Schneider, N., Schluter, N., et al. (eds.), Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event/Bangkok, Thailand, August 5-6, 2021. Association for Computational Linguistics, pp. 502–510. https://doi.org/10.18653/v1/2021.semeval-1.61 (2021)
6. Arslan, Y., Allix, K., Veiber, L., et al.: A comparison of pre-trained language models for multi-class text classification in the financial domain. In: Leskovec, J., Grobelnik, M., Najork, M., et al. (eds.), Companion of the Web Conference 2021, Virtual Event/Ljubljana, Slovenia, April 19-23, 2021. ACM/IW3C2, pp 260–268. https://doi.org/10.1145/3442442.3451375 (2021)
7. Augenstein, I., Das, M., Riedel, S., et al.: Semeval 2017 task 10: Scienceie—extracting keyphrases and relations from scientific publications. In: Bethard, S., Carpuat, M., Apidianaki, M., et al. (eds.), Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017. Association for Computational Linguistics, pp. 546–555. https://doi.org/10.18653/v1/S17-2091 (2017)
8. Báez, M., Birukou, A., Casati, F., et al.: Addressing information overload in the scientific community. IEEE Int. Comput. **14**(6), 31–38 (2010). https://doi.org/10.1109/MIC.2010.107
9. Basha, C.Z., Pravallika, B.N.L., Shankar, E.B.: An efficient face mask detector with pytorch and deep learning. EAI Endorsed. Trans. Pervasive Health Technol. **7**(25), e4 (2021). https://doi.org/10.4108/eai.8-1-2021.167843
10. Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. In: Inui, K., Jiang, J., Ng, V., et al. (eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. Association for Computational Linguistics, pp. 3613–3618. https://doi.org/10.18653/v1/D19-1371 (2019)
11. Bordons, M., Fernández, M.T., Gómez, I.: Advantages and limitations in the use of impact factor measures for the assessment of research performance. Scientometrics **53**(2), 195–206 (2002). https://doi.org/10.1023/A:1014800407876
12. Boudin, F.: Unsupervised keyphrase extraction with multipartite graphs. In: Walker, M.A., Ji, H., Stent, A. (eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers). Association for Computational Linguistics, pp. 667–672. https://doi.org/10.18653/v1/n18-2105 (2018)
13. Bougouin, A., Boudin, F., Daille, B.: Topicrank: Graph-based topic ranking for keyphrase extraction. In: 6th International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013. Asian Federation of Natural Language Processing/ACL, pp. 543–551. https://aclanthology.org/I13-1062/ (2013)
14. Brack, A., D'Souza, J., Hoppe, A., et al.: Domain-independent extraction of scientific concepts from research articles. In: Jose, J.M., Yilmaz, E., Magalhães, J., et al. (eds.), Advances in Information Retrieval—42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I, Lecture Notes in Computer Science, vol 12035. Springer, pp. 251–266. https://doi.org/10.1007/978-3-030-45439-5_17 (2020)
15. Codalab—competition (2021). https://competitions.codalab.org/competitions/25680#results
16. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997). https://doi.org/10.1023/A:1007379606734
17. Celebi, M.E., Aydin, K.: Unsupervised Learning Algorithms, vol. 9. Springer, Berlin (2016)
18. Cohan, A., Ammar, W., van Zuylen, M., et al.: Structural scaffolds for citation intent classification in scientific publications. In: Burstein, J., Doran, C., Solorio, T. (eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics, pp. 3586–3596. https://doi.org/10.18653/v1/n19-1361 (2019)
19. Dawes, M., Sampson, U.: Knowledge management in clinical practice: a systematic review of information seeking behavior in physicians. Int. J. Med. Inform. **71**(1), 9–15 (2003). https://doi.org/10.1016/S1386-5056(03)00023-6
20. D'Souza, J., Auer, S.: Graphing contributions in natural language processing research: intra-annotator agreement on a trial dataset. arXiv:2010.04388 (2020)
21. D'Souza, J., Auer, S.: Nlpcontributions: An annotation scheme for machine reading of scholarly contributions in natural language processing literature. In: Zhang, C., Mayr, P., Lu, W., et al.

(eds.), Proceedings of the 1st Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents co-located with the ACM/IEEE Joint Conference on Digital Libraries in 2020, EEKE@JCDL 2020, Virtual Event, China, August 1st, 2020, CEUR Workshop Proceedings, vol. 2658. CEUR-WS.org, pp. 16–27. http://ceur-ws.org/Vol-2658/paper2.pdf (2020)
22. D'Souza, J., Auer, S.: Sentence, phrase, and triple annotations to build a knowledge graph of natural language processing contributions - A trial dataset. J. Data Inf. Sci. **6**(3), 6–34 (2021). https://doi.org/10.2478/jdis-2021-0023
23. D'Souza, J., Auer, S., Pedersen, T.: Semeval-2021 task 11: Nlpcontributiongraph—structuring scholarly NLP contributions for a research knowledge graph. In: Palmer, A., Schneider, N., Schluter, N., et al. (eds.), Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event / Bangkok, Thailand, August 5-6, 2021. Association for Computational Linguistics, pp. 364–376. https://doi.org/10.18653/v1/2021.semeval-1.44 (2021)
24. Enduri, M.K., Sankar, V.U., Hajarathaiah, K.: Empirical study on citation count prediction of research articles. J. Scientometr. Res. **11**(2), 155–163 (2022)
25. Gábor, K., Buscaldi, D., Schumann, A., et al.: Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In: Apidianaki, M., Mohammad, S.M., May, J., et al. (eds.), Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018. Association for Computational Linguistics, pp. 679–688. https://doi.org/10.18653/v1/s18-1111 (2018)
26. Grineva, M.P., Grinev, M.N., Lizorkin, D.: Extracting key terms from noisy and multitheme documents. In: Quemada, J., León, G., Maarek, Y.S., et al. (eds.), Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009. ACM, pp. 661–670. https://doi.org/10.1145/1526709.1526798 (2009)
27. Gupta, S., Manning, C.D.: Analyzing the dynamics of research by extracting key aspects of scientific papers. In: 5th International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011. The Association for Computer Linguistics, pp. 1–9. https://aclanthology.org/I11-1001/ (2011)
28. Hasan, K.S., Ng, V.: Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In: Huang, C., Jurafsky, D. (eds.), COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China. Chinese Information Processing Society of China, pp. 365–373. https://aclanthology.org/C10-2042/ (2010)
29. He, P., Liu, X., Gao, J., et al.: Deberta: decoding-enhanced bert with disentangled attention. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net. https://openreview.net/forum?id=XPZIaotutsD (2021)
30. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
31. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, Sapporo, Japan, July 11-12, 2003. https://aclanthology.org/W03-1028/ (2003)
32. Jaiswal, A., George, V.: A modified approach for extraction and association of triplets. In: International Conference on Computing, Communication & Automation. IEEE, pp. 36–40 (2015)
33. Jivani, M.A.G., Shingala, M.A.H., Virparia, P.V.: The multi-liaison algorithm. IJACSA Int. J. Adv. Comput. Sci. Appl. **2**(5) (2011)
34. Johnson, R., Watkinson, A., Mabe, M.: The STM report. An overview of scientific and scholarly publishing 5th edition October (2018)
35. Kim, T.K.: T test as a parametric statistic. Korean J. Anesthesiol. **68**(6), 540–546 (2015)
36. Klementiev, A., Roth, D., Small, K.: Unsupervised rank aggregation with distance-based models. In: Cohen, W.W., McCallum, A., Roweis, S.T. (eds.), Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008, ACM International Conference Proceeding Series, vol. 307. ACM, pp. 472–479. https://doi.org/10.1145/1390156.1390216 (2008)
37. Lin, J., Ling, J., Wang, Z., et al.: ECNUICA at semeval-2021 task 11: Rule based information extraction pipeline. In: Palmer, A., Schneider, N., Schluter, N., et al. (eds.), Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event/Bangkok, Thailand, August 5-6, 2021. Association for Computational Linguistics, pp. 1295–1302. https://doi.org/10.18653/v1/2021.semeval-1.185 (2021)
38. Liu, F., Pennell, D., Liu, F., et al.: Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31–June 5, 2009, Boulder, Colorado, USA. The Association for Computational Linguistics, pp. 620–628. https://aclanthology.org/N09-1070/ (2009)
39. Liu, H., Sarol, M.J., Kilicoglu, H.: Uiuc_bionlp at semeval-2021 task 11: A cascade of neural models for structuring scholarly NLP contributions. In: Palmer, A., Schneider, N., Schluter, N., et al. (eds.), Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event/Bangkok, Thailand, August 5-6, 2021. Association for Computational Linguistics, pp. 377–386. https://doi.org/10.18653/v1/2021.semeval-1.45 (2021)
40. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. In: Kambhampati, S. (ed.), Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. IJCAI/AAAI Press, pp. 2873–2879. http://www.ijcai.org/Abstract/16/408 (2016)
41. Liu, P., Qiu, X., Huang, X.: Adversarial multi-task learning for text classification. In: Barzilay, R., Kan, M. (eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. Association for Computational Linguistics, pp. 1–10. https://doi.org/10.18653/v1/P17-1001 (2017)
42. Liu, Z., Rao, B.: Characterization of polynomial decay rate for the solution of linear evolution equation. Zeitschrift für angewandte Mathematik und Physik ZAMP **56**, 630–644 (2005)
43. Llugsi, R., El Yacoubi, S., Fontaine, A., et al.: Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In: 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), IEEE, pp. 1–6 (2021)
44. Luan, Y., He, L., Ostendorf, M., et al.: Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In: Riloff, E., Chiang, D., Hockenmaier, J., et al. (eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. Association for Computational Linguistics, pp. 3219–3232. https://doi.org/10.18653/v1/d18-1360 (2018)
45. Ma, X., Wang, J., Zhang, X.: YNU-HPCC at semeval-2021 task 11: Using a BERT model to extract contributions from NLP scholarly articles. In: Palmer, A., Schneider, N., Schluter, N., et al., (eds.), Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event / Bangkok, Thailand, August 5-6, 2021. Association for Computational Lin-

guistics, pp. 478–484. https://doi.org/10.18653/v1/2021.semeval-1.58 (2021)
46. MacCartney, B.: Natural Language Inference. Stanford University, Stanford (2009)
47. Magnusson, I.H., Friedman, S.E.: Extracting fine-grained knowledge graphs of scientific claims: Dataset and transformer-based results. In: Moens, M., Huang, X., Specia, L., et al. (eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 7-11 November, 2021. Association for Computational Linguistics, pp. 4651–4658. https://doi.org/10.18653/v1/2021.emnlp-main.381 (2021)
48. Mahata, D., Kuriakose, J., Shah, R.R., et al.: Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In: Walker, M.A., Ji, H., Stent, A. (eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers). Association for Computational Linguistics, pp. 634–639. https://doi.org/10.18653/v1/n18-2100 (2018)
49. Mansuri, I.R., Sarawagi, S.: Integrating unstructured data into relational databases. In: Liu, L., Reuter, A., Whang, K., et al., (eds.), Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA. IEEE Computer Society, p. 29. https://doi.org/10.1109/ICDE.2006.83 (2006)
50. Martin, A., Pedersen, T.: Duluth at semeval-2021 task 11: Applying deberta to contributing sentence selection and dependency parsing for entity extraction. In: Palmer, A., Schneider N., Schluter, N., et al. (eds.), Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event / Bangkok, Thailand, August 5-6, 2021. Association for Computational Linguistics, pp. 490–501. https://doi.org/10.18653/v1/2021.semeval-1.60 (2021)
51. Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, pp. 1318–1327. https://aclanthology.org/D09-1137/ (2009)
52. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain. ACL, pp. 404–411. https://aclanthology.org/W04-3252/ (2004)
53. Mishra, S., Mishra, D.: SVM-BT-RFE: an improved gene selection framework using Bayesian T-test embedded in support vector machine (recursive feature elimination) algorithm. Karbala Int. J. Modern Sci. **1**(2), 86–96 (2015)
54. Molchanov, D., Ashukha, A., Vetrov, D.P.: Variational dropout sparsifies deep neural networks. In: Precup, D., Teh, Y.W. (eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, Proceedings of Machine Learning Research, vol. 70. PMLR, pp. 2498–2507. http://proceedings.mlr.press/v70/molchanov17a.html (2017)
55. Nguyen, G.H., Bouzerdoum, A., Phung, S.L.: A supervised learning approach for imbalanced data sets. In: 19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA. IEEE Computer Society, pp. 1–4. https://doi.org/10.1109/ICPR.2008.4761278 (2008)
56. Pascanu, R., Gülçehre ,Ç., Cho, K., et al.: How to construct deep recurrent neural networks. In: Bengio, Y., LeCun, Y. (eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. arXiv:1312.6026 (2014)
57. Poibeau, T.: Machine Translation. MIT Press, Cambridge (2017)
58. Qi, P., Zhang, Y., Zhang, Y., et al.: Stanza: A python natural language processing toolkit for many human languages. In: Celikyilmaz, A., Wen, T. (eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020. Association for Computational Linguistics, pp. 101–108. https://doi.org/10.18653/v1/2020.acl-demos.14 (2020)
59. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv:1706.05098 (2017)
60. Rusu, D., Dali, L., Fortuna, B., et al.: Triplet extraction from sentences. In: Proceedings of the 10th International Multiconference" Information Society-IS, pp. 8–12 (2007)
61. Sahrawat, D., Mahata, D., Zhang, H., et al.: Keyphrase extraction as sequence labeling using contextualized embeddings. In: Jose, J.M., Yilmaz, E., Magalhães, J., et al. (eds.), Advances in Information Retrieval—42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II, Lecture Notes in Computer Science, vol. 12036. Springer, pp. 328–335. https://doi.org/10.1007/978-3-030-45442-5_41 (2020)
62. Sanh, V., Debut, L., Chaumond, J., et al.: Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 (2019)
63. Schafer, J.B., Konstan, J.A., Riedl, J.: E-commerce recommendation applications. Data Min. Knowl. Discov. **5**, 115–153 (2001)
64. Shailabh, S., Chaurasia, S., Modi, A.: Knowgraph@iitk at semeval-2021 task 11: building knowledgegraph for NLP research. arXiv:2104.01619 (2021)
65. Sharma, S., Sharma, S., Athaiya, A.: Activation functions in neural networks. Towards Data Sci. **6**(12), 310–316 (2017)
66. Shi, W., Zheng, W., Yu, J.X., et al.: Keyphrase extraction using knowledge graphs. Data Sci. Eng. **2**(4), 275–288 (2017). https://doi.org/10.1007/s41019-017-0055-z
67. Souza, F., Nogueira, R.F., de Alencar Lotufo, R.: Portuguese named entity recognition using BERT-CRF. arXiv:1909.10649 (2019)
68. Sutton, C., McCallum, A., Rohanimanesh, K.: Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. J. Mach. Learn. Res. **8**, 693–723 (2007). https://doi.org/10.5555/1314498.1314523
69. Trewartha, A., Walker, N., Huo, H., et al.: Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. Patterns **3**(4), 100488 (2022). https://doi.org/10.1016/j.patter.2022.100488
70. Vieira, S.M., Kaymak, U., Sousa, J.M.C.: Cohen's kappa coefficient as a performance measure for feature selection. In: FUZZ-IEEE 2010, IEEE International Conference on Fuzzy Systems, Barcelona, Spain, 18-23 July, 2010, Proceedings. IEEE, pp. 1–8. https://doi.org/10.1109/FUZZY.2010.5584447 (2010)
71. Wade, A.D.: The semantic scholar academic graph (s2ag). Companion Proc. Web Conf. **2022**, 739–739 (2022)
72. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Fox, D., Gomes, C.P. (eds.), Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008. AAAI Press, pp. 855–860. http://www.aaai.org/Library/AAAI/2008/aaai08-136.php (2008)
73. Wang, K., Shen, Z., Huang, C., et al.: Microsoft academic graph: when experts are not enough. Quant. Sci. Stud. **1**(1), 396–413 (2020). https://doi.org/10.1162/qss_a_00021
74. Wang, L.L., Lo, K., Chandrasekhar, Y., et al.: CORD-19: the COVID-19 open research dataset. arXiv:2004.10706 (2020)
75. Wang, M., Zhao, B., Huang, Y.: PTR: phrase-based topical ranking for automatic keyphrase extraction in scientific publications. In: Hirose, A., Ozawa, S., Doya, K., et al. (eds.), Neural Information

Processing—23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16-21, 2016, Proceedings, Part IV, pp. 120–128. https://doi.org/10.1007/978-3-319-46681-1_15 (2016)
76. Wang, R., Liu, W., McDonald, C.: Using word embeddings to enhance keyword identification for scientific publications. In: Sharaf, M.A., Cheema, M.A., Qi, J. (eds.), Databases Theory and Applications—26th Australasian Database Conference, ADC 2015, Melbourne, VIC, Australia, June 4-7, 2015. Proceedings, Lecture Notes in Computer Science, vol. 9093. Springer, pp. 257–268. https://doi.org/10.1007/978-3-319-19548-3_21 (2015)
77. Welch, B.L.: The generalization of 'student's' problem when several different population varlances are involved. Biometrika **34**(1–2), 28–35 (1947)
78. Wise, C., Ioannidis, V.N., Calvo, M.R., et al.: COVID-19 knowledge graph: accelerating information retrieval and discovery for scientific literature. arXiv:2007.12731 (2020)
79. Wu, Z., Giles, C.L.: Measuring term informativeness in context. In: Vanderwende III, L. H.D, Kirchhoff, K. (eds.), Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA. The Association for Computational Linguistics, pp. 259–269. https://aclanthology.org/N13-1026/ (2013)
80. Xu, H., AbdelRahman, S., Jiang, M., et al.: An initial study of full parsing of clinical text using the Stanford parser. In: 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW), IEEE, pp. 607–614 (2011)
81. Yih, W., Goodman, J., Carvalho, V.R.: Finding advertising keywords on web pages. In: Carr, L., Roure, D.D., Iyengar, A., et al. (eds.), Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006. ACM, pp. 213–222. https://doi.org/10.1145/1135777.1135813 (2006)
82. Yu, B., Li, Y., Wang, J.: Detecting causal language use in science findings. In: Inui, K., Jiang, J., Ng, V., et al. (eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. Association for Computational Linguistics, pp. 4663–4673. https://doi.org/10.18653/v1/D19-1473 (2019)
83. Yu, Y., Ng, V.: Wikirank: Improving keyphrase extraction based on background knowledge. arXiv:1803.09000 (2018)
84. Zhang, G., Su, Y., He, C., et al.: ITNLP at semeval-2021 task 11: Boosting BERT with sampling and adversarial training for knowledge extraction. In: Palmer, A., Schneider, N., Schluter, N., et al. (eds.), Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event/Bangkok, Thailand, August 5-6, 2021. Association for Computational Linguistics, pp. 485–489. https://doi.org/10.18653/v1/2021.semeval-1.59 (2021)
85. Zhang, J., Li, C., Yin, Y., et al.: Applications of artificial neural networks in microorganism image analysis: a comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer. Artif. Intell. Rev. **56**(2), 1013–1070 (2023). https://doi.org/10.1007/s10462-022-10192-7
86. Zhang, Q., Wang, Y., Gong, Y., et al.: Keyphrase extraction using deep recurrent neural networks on twitter. In: Su, J., Carreras, X., Duh, K. (eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. The Association for Computational Linguistics, pp. 836–845. https://doi.org/10.18653/v1/d16-1080 (2016)
87. Zhang, Y., Milios, E.E., Zincir-Heywood, A.N.: A comparative study on key phrase extraction methods in automatic web site summarization. J. Digit. Inf. Manag. **5**(5), 323–332 (2007)
88. Zhang, Z., Han, X., Liu, Z., et al.: ERNIE: enhanced language representation with informative entities. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers. Association for Computational Linguistics, pp. 1441–1451. https://doi.org/10.18653/v1/p19-1139 (2019)
89. Zhang, Z., Strubell, E., Hovy, E.H.: A survey of active learning for natural language processing. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. Association for Computational Linguistics, pp. 6166–6190. https://doi.org/10.18653/v1/2022.emnlp-main.414 (2022)
90. Zhu, X., Lyu, C., Ji, D., et al.: Deep neural model with self-training for scientific Keyphrase extraction. PLoS ONE **15**(5), e0232,547 (2020)