

# KnowGraph@IITK at SemEval-2021 Task 11: Building Knowledge Graph for NLP Research

**Shashank Shailabh\***    **Sajal Chaurasia\***    **Ashutosh Modi**

Indian Institute of Technology Kanpur (IIT Kanpur)

{shailabh, sajal}@iitk.ac.in

ashutoshm@cse.iitk.ac.in

## Abstract

Research in Natural Language Processing is making rapid advances, resulting in the publication of a large number of research papers. Finding relevant research papers and their contribution to the domain is a challenging problem. In this paper, we address this challenge via the SemEval 2021 Task 11: NLPContributionGraph, by developing a system for a research paper contributions-focused knowledge graph over Natural Language Processing literature. The task is divided into three sub-tasks: extracting contribution sentences that show important contributions in the research article, extracting phrases from the contribution sentences, and predicting the information units in the research article together with triplet formation from the phrases. The proposed system is agnostic to the subject domain and can be applied for building a knowledge graph for any area. We found that transformer-based language models can significantly improve existing techniques and utilized the SciBERT-based model. Our first sub-task uses Bidirectional LSTM (BiLSTM) stacked on top of SciBERT model layers, while the second sub-task uses Conditional Random Field (CRF) on top of SciBERT with BiLSTM. The third sub-task uses a combined SciBERT based neural approach with heuristics for information unit prediction and triplet formation from the phrases. Our system achieved F1 score of 0.38, 0.63 and 0.76 in end-to-end pipeline testing, phrase extraction testing and triplet extraction testing respectively.

## 1 Introduction

Given the advancements in Natural Language Processing (NLP), a large number of research papers are published every year. However, given the field's dynamic nature, keeping track of all the papers is a non-trivial task. This motivated the formulation

of an Open Research Knowledge Graph (Jaradeh et al., 2019), a knowledge graph of research contributions and the relation between them. Task 11 of SemEval 2021 (D'Souza et al., 2021) formalizes the building of a contributions-focused knowledge graph of NLP literature. The task is divided into three sub-tasks:

- **Sub-task A** Extracting sentences that posit contributions in a research paper.
- **Sub-task B** Extracting relevant phrases that include scientific terms and relational cues from the extracted sentences of the sub-task A.
- **Sub-task C** Triplet (subject phrase, predicate phrase, object phrase) formation from the extracted phrases of the sub-task B and classification of the triplet in one of the information units (IU). There are twelve information units (Research problem, Approach, Results, Model, Code, Dataset, Experimental setup, Hyperparameters, Baselines, Tasks, Experiments, and Ablation analysis), each focusing on different sections in a research paper. These information units can represent all the findings given in a research paper. Out of twelve, first three information units are present in each article.

Recently, transformer-based approaches have been popular for NLP applications (Liu and Lapata, 2019; Yao et al., 2019). For the sub-task A, we propose a sentence level classifier, leveraging SciBERT (Beltagy et al., 2019) and Bidirectional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997). SciBERT model has been trained on 1.14M scientific papers from Semantic Scholar corpus, which has 18% papers from the computer science domain. Our system for sub-task B also uses SciBERT based model using CRF

\* Authors equally contributed to this work.

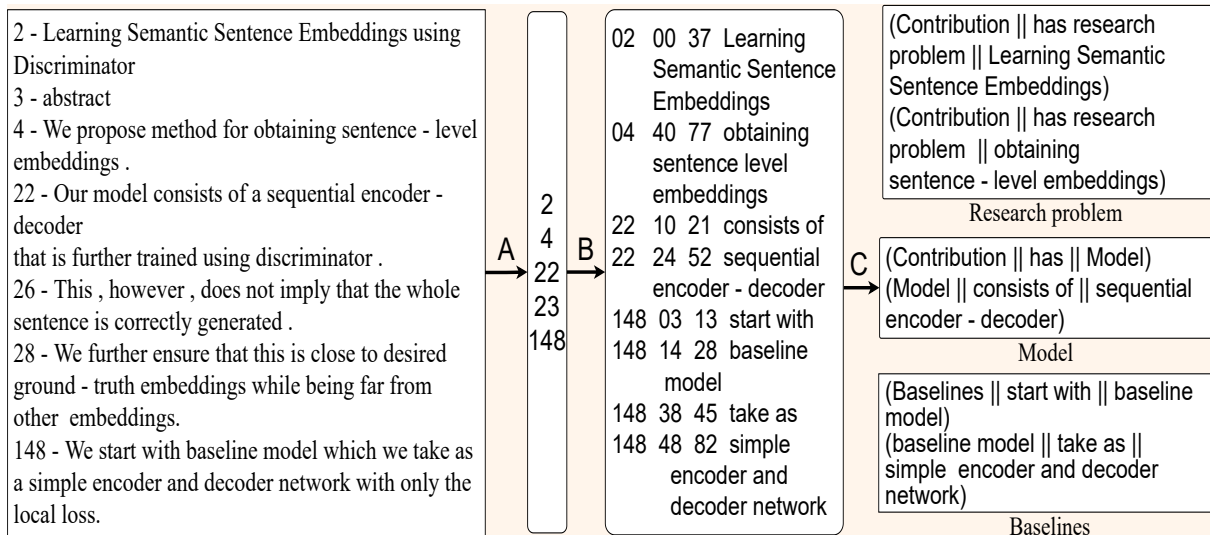


Figure 1: Dataset Example

(Lafferty et al., 2001) on top of BiLSTM layers using BILUO (B=start token of phrase, I=interior tokens of phrase, L=last token of phrase, U=single token phrase and O=Non-phrase tokens) labelling scheme for tokens. For sub-task C, we build a combination of neural and heuristic-based approach. The IU prediction for sub-task C is at document level where two information units use heuristic approach while others use a multi-label classifier based on BiLSTM stacked on top of SciBERT. For triplet formation in sub-task C, we use a separate SciBERT+BiLSTM classifier along with heuristics. These triplets are classified into one of the already predicted information units, i.e., the output unit having the maximum score among the predicted IU. However, some of the IU, such as Baselines, Ablation analysis, Code and Research problem triplets, perform well using heuristic, so we use heuristic instead of a neural approach for these IU. Our proposed system was ranked third in the overall end-to-end pipeline (A+B+C) testing and achieved F1 score of 0.38. Our proposed model was ranked fourth in phrase extraction ( $A_{GT}+B+C$ ) and triplet extraction testing ( $A_{GT}+B_{GT}+C$ ) with a F1 score of 0.63 and 0.76 respectively where  $A_{GT}$  and  $B_{GT}$  represent ground-truth for sub-task A and B respectively. Phrase extraction testing uses ground-truth labels for sub-task A, while triplet extraction testing uses ground-truth for both sub-task A and B. We found that the heuristic-based model for two IU (Research problem and Code) in sub-task C can achieve high performance and achieved F1 score of 0.98 and 1.00, respectively. Our system imple-

mentation code is made available via GitHub<sup>1</sup>.

## 2 Background

### 2.1 Problem Definition

Consider a document  $D = \{s_1, s_2, \dots, s_i, \dots, s_N\}$  having  $N$  sentences  $s_i$ . Sub-task A finds  $M$  contribution sentences denoted by  $S = \{s_1, s_2, \dots, s_M\}$  from  $D$ . Sub-task B selects phrases  $P = \{p_1, p_2, \dots, p_i, \dots, p_L\}$  where  $p_i$  is a phrase selected from a sentence  $s \in S$  and  $L$  is total number of phrases in  $D$ . Sub-task C is forming triplets of extracted phrases for IU denoted by  $U = \{u_1, \dots, u_i, \dots, u_X\}$  where  $u_i$  is one of the twelve IU and  $X$  is number of IU in document  $D$  ranging between three to twelve. For each  $u_i \in U$ , there is a triplet set called  $T^i = \{(su_1^i, pr_1^i, ob_1^i), (su_2^i, pr_2^i, ob_2^i), \dots, (su_j^i, pr_j^i, ob_j^i), \dots, (su_O^i, pr_O^i, ob_O^i)\}$  where  $(su_j^i, pr_j^i, ob_j^i)$  is a triplet representing subject, predicate (relation) and object respectively and  $O$  is total number of triplets in  $u_i$  IU in document  $D$ . An example dataset is given in Figure 1 for reference. Here, on moving from left to right is research paper, contribution sentences, phrases and IU along with triplets given in a research paper respectively.

### 2.2 Related Work

Knowledge graphs (Rebele et al., 2016; Hertling and Paulheim, 2018; Lehmann et al., 2015; Carlson et al., 2010) have shown to be helpful in several areas such as search, knowledge extraction, inter

<sup>1</sup><https://github.com/sshailabh/SemEval-2021-Task-11>

alia. However, only a handful are based on research articles (Xu et al., 2020). Typically, most knowledge graphs are created with rule-based approaches, hence, limiting their performance and generalization. However, some recent approach such as Sang et al. (2018); Wang et al. (2020b) uses neural approach in biomedical literature. To the best of our knowledge, there is no available contributions-focused knowledge graph over NLP literature using the neural approach.

**Sub-task A:** The sub-task of extracting contribution sentences can also be posed as an extractive summarization problem (Nallapati et al., 2016; Narayan et al., 2018; Liu, 2019; Cheng and Lapata, 2016; Zhou et al., 2018; Dong et al., 2018; Wang et al., 2020a). BERTSUM (Liu and Lapata, 2019) and MATCHSUM (Zhong et al., 2020) are the recent methods leveraging language models and uses ROUGE-1, ROUGE-2 and ROUGE-L scores (Lin, 2004) on DailyMail data-set (Hermann et al., 2015). However, this extractive summarization technique may not be applicable in our case due to a number of reasons. Firstly, extractive summarization alone will not give all the contribution sentences because some sentences may not be relevant to the summarization task. Secondly, extractive summarization models are not tested on large documents such as research articles due to the limitation of the input token length for transformer-based language models. Some long document transformer-based methods are proposed (e.g., Beltagy et al., 2020), and can consider documents up to a length of 4096 tokens, however, in our case, documents have on an average  $\sim 10,000$  tokens. Some of the extractive summarization methods (Liu and Lapata, 2019; Miller, 2019) take the number of contribution sentences as a hyper-parameter, but in our case, this is a trainable parameter in our model.

**Sub-task B:** Sub-task B closely resembles the phrase extraction problem and several neural methods (Zhu et al. (2020); Wang et al. (2016); Zhang et al. (2016), inter alia) and non-neural based methods (using n-grams and noun-phrases with certain Part-of-speech (POS) patterns (Hulth, 2003)) have been proposed. Gollapalli et al. (2017) have shown that CRF has the potential to improve the existing phrase extraction model. Alzaidy et al. (2019) jointly leverages CRF and BiLSTM to capture hidden semantics for phrase extraction. Zhu et al. (2020) extended the work of Alzaidy et al. (2019) with the idea of self-training and used word em-

beddings, POS embeddings, and dependency embeddings with a BILUO labelling scheme in the output. Our proposed model took inspiration from Zhu et al. (2020) and propose a SciBERT based model using CRF on top of BiLSTM layers using BILUO labelling scheme on tokens. Our model captures better semantics than the word embeddings based approach in Zhu et al. (2020) because of SciBERT, which is trained on the scientific corpus. Moreover, our model uses the WordPiece tokenizer and hence, robust to Out-of-Vocabulary (OOV) tokens. Sahrawat et al. (2020) used contextual embeddings to the BiLSTM and CRF model using BIO (B=start token of phrase, I=continuation tokens of phrase and O=Non-phrase tokens) labelling scheme. Ratnov and Roth (2009) discussed that the BILUO scheme is superior to the BIO scheme; hence we adopt the BILUO scheme for sub-task B. Recently, Lai et al. (2020) combined sequence labelling with joint learning inspired from self-distillation to boost model performance on unsupervised datasets. However, their model used BIO labelling scheme and gave a comparable or marginal improvement in a supervised setting.

**Sub-task C:** The sub-task C can be divided into two parts - information units (IU) prediction and triplet formation. The information unit prediction and triplet formation have been approached in the literature mainly using rule-based methods. Rusu et al. (2007) suggests using syntactic parsers for generating parse trees, followed by triplet extraction using parser dependent techniques. Jivani et al. (2011) proposed an algorithm that exhibits the relationship between subject and object in a sentence using Stanford parser. This rule-based algorithm can form multiple triplets from a sentence as compared to Rusu et al. (2007). Stanford OpenIE Relation triplet formation (Angeli et al., 2015) uses a classifier, which learns to extract self-contained clauses from longer sentences to form the final triplets using heuristics. Hamoudi (2016) and Jaiswal and George (2015) are also rule-based methods for triplet formation using Stanford dependency parser and constituency parser, respectively. KG-Bert (Yao et al., 2019) uses the BERT language model and utilize entity and relation descriptions of a triplet to compute its scoring function.

### 3 System Overview

The proposed system is shown in Figure 2 depicting the entire pipeline and its respective model.

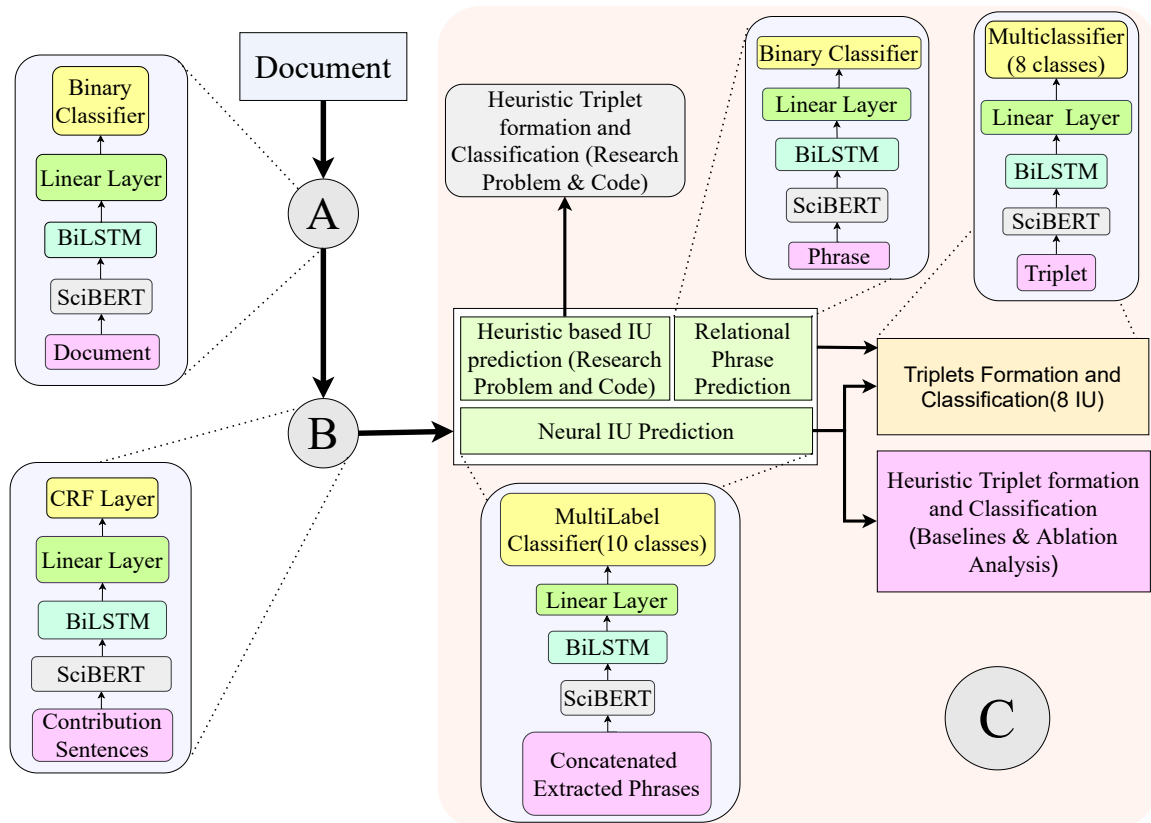


Figure 2: Overall system architecture for end-to-end pipeline showing all the sub-tasks and their respective model.

### 3.1 Sub-Task A

**Initial Experimentation:** We experimented with SciBERT, a language model trained on research papers to build a binary classifier. The classifier encodes the sentence using feature representation corresponding to [‘CLS’] token from the last layer of the pre-trained SciBERT model to predict a binary label. Since the fine-tuning dataset is small, linear layers were not able to capture contextual information well. We experimented by adding a Convolution Neural Network (Zhang and Wallace, 2015) layer on the top of SciBERT. The CNN architecture use three kernels of size two, three and four. The model boosted the performance; however, the model cannot classify long sentences due to their ambiguous nature and lack of CNN’s capacity to capture long semantic dependency among tokens.

**Proposed Approach:** We propose the SciBERT+BiLSTM model (a sentence-level binary classifier where BiLSTM is stacked on top of the SciBERT model). This helps to encode hidden semantics and long-distance dependency. Consider the training dataset as  $Tr = \{D_1, D_2, \dots, D_i, \dots, D_Z\}$  comprising of  $Z$  documents. Each  $D_i$  can be represented as  $D_i = \{s_{i1}, s_{i2}, \dots, s_{ij}, \dots, s_{iN}\}$  where  $N$

is the number of sentences in the document and  $s_{ij}$  is the  $j^{th}$  sentence of document  $D_i$ . Each sentence is assigned a ground-truth label where label “1” represents a contribution sentence and label “0” a non-contribution sentence. The sentences are processed using a SciBERT model followed by stacked BiLSTM layers whose output is further processed through linear layers with ReLU (Nair and Hinton, 2010) non-linearity. We add dropout layers to avoid overfitting. The last linear layer consists of two units corresponding to label “0” and label “1”. The final output label is the label whose corresponding unit has a higher score in the last linear layer. Our loss function is weighted binary cross-entropy loss, where weights are in according to the number of samples in each class.

### 3.2 Sub-Task B

**Initial Experimentation:** We built our initial method, the BiLSTM+CRF model, on the lines of Zhu et al. (2020). The model uses a BiLSTM layer along with CRF for sequence to sequence (BILUO) labelling in order to mark the phrases in a sentence. We introduced SciBERT (by replacing BiLSTM layers) to fine-tune and better generalise with an increase in performance. We tested the significance



of CRF by replacing it with a softmax layer, which gave poor performance since it is unable to learn the constraints in the BILUO scheme.

**Proposed Approach:** We further improved the SciBERT+CRF model to improve semantic information. Our proposed model stack BiLSTM layers on top of SciBERT, followed by CRF (see Figure 2). The word-level representation  $\{x_1, x_2, \dots, x_N\}$  from the input sentence passes through the SciBERT tokenizer. We used the representation of the first sub-token for every word as the input to the SciBERT. The tokenized input is passed into the SciBERT layer, followed by the BiLSTM layer. The final feature output is mapped to a hidden linear layer to get the score matrix  $Z$ , which is passed into the CRF layer for label prediction  $y$ . The CRF layer is the same as described in Lample et al. (2016).

The output produced by the SciBERT + BiLSTM + Hidden Layer corresponds to a scoring matrix  $Z^{(n \times l)}$  where  $n$  denotes the number of words in the input sentence, and  $l$  is the number of labels ( $l=5$ ). The score of an output sequence  $y$  using CRF is given by:

$$Scr(s, y) = \sum_{i=0}^n (Z_{i, y_i} + T_{y_{i-1}, y_i}) \quad (1)$$

where  $Z_{i,j}$  denotes the score of word  $w_i$  with the  $j$ th label,  $T_{y_{i-1}, y_i}$  is the transition score from the label  $y_{i-1}$  to  $y_i$ ,  $y = \{y_1, y_2, \dots, y_n\}$  is the sequence of true labels and  $Scr(s, y)$  corresponds to output score for sentence  $s$  and true labels  $y$ . A softmax over all possible label sequences yields a probability for true labels  $y$ :

$$P(y|s) = \frac{\exp(Scr(s, y))}{\sum_{y' \in Y(s)} \exp(Scr(s, y'))} \quad (2)$$

where  $Y(s)$  corresponds to all the possible label sequences for sentence  $s$ . Now during training, our task is to maximize the log-probability of the correct label sequence  $y$ . Model loss is defined as follows:

$$L(\Theta) = -(1/M) \sum_{i=1}^M \log(P(y_i|s_i)) + \frac{\lambda}{2} \|\Theta\|^2 \quad (3)$$

where  $s_i$  is the input sentence,  $y_i$  is the corresponding true label sequence,  $\Theta$  denotes the model parameters,  $\lambda$  is the regularization hyperparameter and  $M$  is the train set size. Output label prediction is made by:

$$y^* = \operatorname{argmax}_{y' \in Y(s)} P(y'|s) \quad (4)$$

Here  $y^*$  represents the final output label sequence,  $s$  is the input sentence,  $Y(s)$  is the set of possible label sequences and  $P(y'|s)$  denotes the probability of getting  $y'$  label sequence from sentence  $s$ . The phrases are extracted using BILUO scheme based on the prediction outputs.

### 3.3 Sub-Task C

**Initial Experimentation:** We used a combination of neural and rule-based approach for sub-task C. An IU triplet has three phrases - subject, predicate and object. Research problem and Code IU triplets have fixed subject and predicate in triplets. We employed a heuristic to scan the phrases of the first thirty lines of each document and select only those sentence's phrases that have only a single phrase extracted out. These phrases form object in Research problem IU triplets. A regex expression is used to extract all the sentences in the article that contain any URLs for Code IU triplet's object. However, only those URL sentences are selected which have token such as "our" or "code" or "our code".

Our initial approach was to form the triplets for all other IU and classify them into one of the ten remaining information units using SciBERT + BiLSTM multi-class classifier (BiLSTM layers stacked on top of SciBERT). The heuristic for triplet formation is based on orthographic visualization of the document - firstly, the phrases are arranged in the exact order as they appear in the original sentence. Then, every three consecutive pair of phrases present within the same sentence are considered as one triple. This approach gave us decent results since most of the research paper is written in the active voice; hence the subject phrase should occur first, then its corresponding predicate and last should be the object phrase. The SciBERT + BiLSTM multi-class classifier takes concatenated triplets as the input and their corresponding information unit as the ground truth. The loss is a cross-entropy loss, with the final softmax layer having ten classes corresponding to 10 information units. One of the drawbacks of the model is that the IU prediction depends on triplet formation's correctness. Further, a single triplet does not have enough context to be correctly classified into the correct information unit. We visualized triplet formation for feature extraction and found that some IU triplet's, such as Baselines and Ablation analysis, can be better formed using heuristics. Our proposed approach

is a better triplet formation model and eliminates these limitations.

**Proposed Approach:** Our proposed approach has some information unit such as Research problem and Code, whose model for prediction and triplet formation is entirely heuristic-based and the same as initial experimentation. Our proposed method for the rest of ten IU is divided into two parts -

**IU Prediction** - We propose a SciBERT+BiLSTM multi-label classifier (BiLSTM layers stacked on top of SciBERT) (refer to Figure 2) whose input is the concatenated phrases and predicts the IU of the document. The concatenation is in the order of the occurrence of phrases in the document. Moreover, these concatenated extracted phrases represent the whole document since it includes all relevant keywords of the research article necessary for information unit prediction. Hence, our proposed model encodes information at the document level, which makes it superior to the initial experimentation method.

**Triplet Formation and Classification** - In general, a phrase is either a relational phrase or a scientific term (Figure 1). A specific trend observed in ground truth triplets is that a triplet’s predicate is unique for a triplet. This contrasts with the subject and object phrase, which can be used multiple times in other triplets. Hence, a one-to-one relation between predicate phrase and triplet exists. We identify all predicate phrases from the extracted phrases of sub-task B; then, a corresponding triplet will be formed for each predicate phrase. We train a SciBERT+BiLSTM based binary classifier (BiLSTM layers stacked on top of SciBERT model) to identify the phrases which act as predicates (relational phrases). The model is fine-tuned on our dataset and uses the weighted binary cross-entropy loss. In labelling, “1” denote as predicate while “0” denote as non-predicate. To form triplets, we use a simple heuristic to arrange the phrases in the exact order as they appear in the original sentence. For every phrase predicted as a predicate, we take its previous phrase as the subject and its next phrase as the triplet object.

Now, a multi-class classifier for triplet classification for 8 IU (corresponding to all IU except Research problem, Code, Baselines and Ablation analysis) (refer to Figure 2) is built, which is similar to the one described in the initial experimentation. Since we have already predicted the information units, during inference, the triplet can be assigned

only to one of the already predicted information units, i.e., the output unit having the maximum score among the predicted IU.

We used rule-based heuristics for triplet formation of Baselines and Ablation analysis IU. The target sentences, whose phrases belong to baselines IU, are identified by selecting all the headings (i.e. lines having no punctuation) with words such as “baseline”, “comp” using a regex expression. Then, we took all the sentences between selected headings and their consecutive headings as the target sentences. The phrases associated with extracted sentences are used for triplet formation via the rule of three consecutive phrases present within the sentence. The same method is followed for Ablation analysis IU triplets with only change that the relevant headings are found using the regex expression that identifies if that heading contains the word such as “ablation”, “analysis”.

## 4 Experimental Setup

### 4.1 Data

The dataset annotation scheme is as per [D’Souza and Auer \(2020\)](#). The pre-processed dataset con-

Token Length	% of sentences less than token length
50	94.57
100	99.74
150	99.93
200	99.96

Table 1: Token length statistics on train set

sists of 287 annotated NLP research documents in the English language with ground truth for each sub-task. Train, dev and test set have 237, 50 and 155 documents, respectively. We have chosen 100 as the maximum token length in a sentence with WordPiece tokenizer since 99.7% sentences in the train set have less than or equal to 100 tokens. The Table 1 shows token length and percentage(%) of sentences less than that length. In the sub-task C, if there is no suitable predicate available in the extracted phrases, then the triplet’s predicate is chosen from the predefined set of predicates, i.e. “has”, “on”, “by”, “for”, “has value”, “has description”, “based on”, “called”. Table 2 shows the dataset statistics related to sub-task A and B. The dataset statistics for sub-task C is given in Table 3. Tasks information unit has no triplets in train set while

Statistics	Train	Dev
# Documents	237	50
# Contribution sentences	5096	1032
# Non-contribution sentences	50105	10451
# Avg. Sentences in doc.	232.915	229.7
# Avg. Tokens in sentence	20.622	21.06
# Avg. Contribution Sentences in doc.	21.38	20.24
# Avg. Phrases in doc.	128.53	92.52
# Avg. IU in doc.	4.43	4.46
# Max Tokens in sentence	396	193

Table 2: Dataset Statistics

# Information Unit Triplets	Train	Dev
Research problem	635	164
Approach	529	233
Model	3548	570
Code	40	9
Dataset	240	8
Experimental setup	1928	302
Hyperparameters	2267	254
Baselines	1625	146
Results	4989	657
Tasks	0	277
Experiments	1472	149
Ablation analysis	1407	155

Table 3: Number of triplets in each Information Unit on train and dev set.

Code and Dataset information units have very few triplets in dev set.

## 4.2 Hyperparameters

We have used the dev set to tune our hyperparameters. In every neural model, we are fine-tuning SciBERT.<sup>2</sup> We tried different batch sizes and learning rates for fine-tuning (Dodge et al., 2020). We found the best results using the AdamW optimizer in the neural models.

**Sub-task A** : We used batch size = 32, learning rate = 1e-05, epoch = 2, two layers of BiLSTM with hidden dimension of 400 and three linear layers (size = 800, 400, 100) with dropout = 0.1. Over-sampling of minority class (contribution sentences) counters the skewness in data.

**Sub-task B** : We used batch size = 1, learning rate = 2e-05, epoch = 4, single layer of BiLSTM with

<sup>2</sup><https://github.com/huggingface/transformers>

hidden dimension = 200 and linear layer with CRF. **Sub-task C** : We used batch size = 4, max-tokens = 512, learning rate = 2e-05, epoch = 16, threshold on sigmoid output = 0.5, two layers of BiLSTM with hidden dimension of 400 and three linear layers (size = 800,400,100) with dropout = 0.2 for multi-label classification of information units (ten out of twelve). We have used batch size = 32, max tokens = 25, learning rate = 2e-05, epoch = 4, two BiLSTM layers with hidden dimension = 400 and three linear layers (size = 800, 400, 100) with dropout = 0.1 for relational phrase prediction model. We have used batch size = 16, max tokens = 50, learning rate = 2e-05, epoch = 2, two BiLSTM layers with hidden dimension = 400 and three linear layers (size = 800, 400, 100) with dropout = 0.2 for triplet classification (eight out of twelve info-units).

## 4.3 Evaluation Metrics

In this task, organizers used Precision, Recall and F1 score metrics. In sub-task A, the predicted and ground-truth contribution sentences of the document calculate the metrics score. The sub-task B output has predicted phrase, contribution sentence number, starting and ending character number of the predicted phrase (Figure 1). These outputs are the basis for sub-task B metric calculations. The sub-task C has two groups of metrics. First is the Information Units prediction of a document (regardless of triplets). The second group calculates using both Information Units and triplets (the predicted triplet is correct only if it exactly matches the ground truth triplet and the ground-truth IU; otherwise, it is incorrect). The final score is the average of all the four F1 scores on the dataset. The participating team rankings are according to this score.

## 5 Results

Table 4 shows all the participating teams average F1 score. In end-to-end pipeline testing, the input is the documents. In phrase extraction testing ( $A_{GT}+B+C$ ), the input is the document and ground-truth for sub-task A. Here,  $A_{GT}$  represents the true label of sub-task A, and the F1 score for sub-task A is 1.00. In the triplet extraction phase ( $A_{GT}+B_{GT}+C$ ), the input is the document and ground truth labels (F1 score = 1.00) for both sub-tasks A and B to the system. Our team “Know-Graph@IITK” achieved an F1 score of 0.3783 and ranked third in end-to-end pipeline testing.

Team Name	A+B+C	A <sub>GT</sub> +B+C	A <sub>GT</sub> +B <sub>GT</sub> +C
BioNLP@UIUC	0.3828	<b>0.7612</b>	<b>0.8594</b>
ecnuica	0.3335	0.7113	0.8145
ITNLP	<b>0.4703</b>	0.6863	0.7931
<b>KnowGraph@IITK</b>	<b>0.3783</b>	<b>0.6318</b>	<b>0.7600</b>
INNOVATORS	0.3205	0.5252	0.5971
DuluthGrad	0.2838	0.4921	0.7579
YNU-HPCC	-	0.4562	0.6541
DFKI-SLT	0.2651	-	0.7137
NLP_IITGN	-	0.3522	-

Table 4: Average F1 score on test set of participating teams in end-to-end pipeline testing(A+B+C), phrase extraction testing(A<sub>GT</sub>+B+C) and triplet extraction testing(A<sub>GT</sub>+B<sub>GT</sub>+C).

In phrase extraction and triplet extraction testing phases, our system ranked fourth with an F1 score of 0.6318 and 0.76, respectively. Our heuristic-based triplet extraction for Code and Research problem information unit achieved excellent performance with an F1 score of 1.00 and 0.9756, respectively, on the test set.

## 5.1 Ablation and Error Analysis

Methods	Dev F1 score
Sub-task A	
SciBERT + CNN	0.440
SciBERT + BiLSTM	<b>0.451</b>
Sub-Task B	
BiLSTM + CRF	0.361
SciBERT + CRF	0.444
SciBERT + BiLSTM + CRF	<b>0.480</b>

Table 5: F1 score of several methods of sub-task A and sub-task B on development set.

We built several models using SciBERT, LSTM and CNN for each sub-task to understand each method’s significance (Table 5). We found that language models are knowledge-rich and boost the existing models. On top of language models, BiLSTM based model performs better than the CNN-based model due to long semantic dependency in sequential models. In sub-task B, BiLSTM+CRF based model performed inferior to the same model built on top of SciBERT. In triplet formation in sub-task C, our rule-based approach of Research problem and Code information unit yield excellent results (highest on the leaderboard). A significant improvement in the F1 score for Baseline and Ablation Analysis IU suggests that the rule-based approach can boost neural models since specific

patterns are present for these information units’ triplets. In sub-task A, the dataset is highly skewed between minority and majority classes (1:10), making the training of a neural model difficult. On visualization of sub-task B outputs, we found some ambiguous phrases that our model fails to predict correctly. Extracting both scientific and relation cue phrases with high precision and recall in a single model is difficult. Sometimes, our model predicts scientific phrases correctly but fails to predict relation cue phrase in the same sentence.

In sub-task C, some IU triplet’s such as Model, Hyperparameters, Results were present in large number comparatively and while Tasks and Dataset IU triplets are scarce in number. This skewness resulted in biasing of our multi-label classification model for IU prediction. In triplet formation, our predicate approach fails when the predicate is not present in the sentence and selected from the organizer’s closed set of predicates. The proposed system also fails in the case of branching between the triplets, i.e. multiple triplets share the same subject phrase. Our triplet formation is unable to predict Approach, Dataset and Tasks triple. On visualizing, we found that Model triplets are predicted instead of Approach triplets. Further, the triplets lack enough features to be classified into an information unit using a neural-based method. Our system performance on sub-task C is poor in the end-to-end pipeline and phrase extraction testing because we submitted our initial experimentation model.

## 6 Conclusion

In this paper, we have presented our system for NLPContributionGraph task of SemEval 2021. We found that neural models combined with heuristics



can build a knowledge graph by dividing it into small tasks. The heuristic-based model can outperform the neural approach in the triplet formation of some Information Units. In future work, neural models can use sparse transformers to encode long documents without increasing much memory. The pre-defined predicate incorporation could also be a future direction of work for our system.

## Acknowledgments

We thank Shubham Kumar Nigam for his guidance and SemEval-2021 organizers for running the competition. We are also very grateful to the Pytorch developers.

## References

- Rabah Alzaidy, Cornelia Caragea, and C Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The world wide web conference*, pages 2551–2557.
- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. [Leveraging linguistic structure for open domain information extraction](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110.
- Jianpeng Cheng and Mirella Lapata. 2016. [Neural summarization by extracting sentences and words](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.
- Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. [BanditSum: Extractive summarization as a contextual bandit](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium. Association for Computational Linguistics.
- Jennifer D’Souza, Sören Auer, and Ted Pedersen. 2021. SemEval-2021 task 11: Nlpcontributiongraph - structuring scholarly nlp contributions for a research knowledge graph. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*, Bangkok (online). Association for Computational Linguistics.
- Jennifer D’Souza and S. Auer. 2020. Nlpcontributions: An annotation scheme for machine reading of scholarly contributions in natural language processing literature. *ArXiv*, abs/2006.12870.
- Sujatha Das Gollapalli, Xiao-Li Li, and Peng Yang. 2017. Incorporating expert knowledge into keyphrase extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Yassine Hamoudi. 2016. Extracting rdf triples using the stanford parser.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems* 28, pages 1693–1701. Curran Associates, Inc.
- Sven Hertling and Heiko Paulheim. 2018. Dbkwik: A consolidated knowledge graph from thousands of wikis. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 17–24. IEEE.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Anette Hulth. 2003. [Improved automatic keyword extraction given more linguistic knowledge](#). In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223.
- A. Jaiswal and V. George. 2015. A modified approach for extraction and association of triplets. *International Conference on Computing, Communication and Automation*.
- Mohamad Yaser Jaradeh, Allard Oelen, Kheir Ed-dine Farfar, Manuel Prinz, Jennifer D’Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*, pages 243–246.

- Ms Anjali Ganesh Jivani, Ms Amisha Hetal Shingala, and Paresh V Virparia. 2011. The multi-liaison algorithm. (*IJACSA International Journal of Advanced Computer Science and Applications*, 2(5).
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Tuan Lai, Trung Bui, Doo Soon Kim, and Quan Hung Tran. 2020. [A joint learning approach based on self-distillation for keyphrase extraction from scientific documents](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 649–656, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#).
- Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.
- Yang Liu and Mirella Lapata. 2019. [Text summarization with pretrained encoders](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.
- Derek Miller. 2019. Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.
- Vinod Nair and Geoffrey E. Hinton. 2010. [Rectified linear units improve restricted Boltzmann machines](#). In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, Haifa, Israel. Omnipress.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *arXiv preprint arXiv:1611.04230*.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Ranking sentences for extractive summarization with reinforcement learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.
- Lev Ratinov and Dan Roth. 2009. [Design challenges and misconceptions in named entity recognition](#). In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.
- Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, pages 177–185. Springer.
- Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. 2007. Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference “Information Society-IS*, pages 8–12.
- Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. In *European Conference on Information Retrieval*, pages 328–335. Springer.
- Shengtian Sang, Zhihao Yang, Lei Wang, Xiaoxia Liu, Hongfei Lin, and Jian Wang. 2018. Sematyp: a knowledge graph based literature mining method for drug discovery. *BMC bioinformatics*, 19(1):1–11.
- Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020a. [Heterogeneous graph neural networks for extractive document summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.
- Minmei Wang, Bo Zhao, and Yihua Huang. 2016. Ptr: phrase-based topical ranking for automatic keyphrase extraction in scientific publications. In *International Conference on Neural Information Processing*, pages 120–128. Springer.
- Qingyun Wang, Manling Li, Xuan Wang, Nikolaus Parulian, Guangxing Han, Jiawei Ma, Jingxuan Tu, Ying Lin, Haoran Zhang, Weili Liu, et al. 2020b. Covid-19 literature knowledge graph construction and drug repurposing report generation. *arXiv preprint arXiv:2007.00576*.
- J. Xu, S. Kim, M. Song, M. Jeong, D. Kim, J. Kang, J. F. Rousseau, X. Li, W. Xu, V. I. Torvik, Y. Bu, C. Chen, I. A. Ebeid, D. Li, and Y. Ding. 2020. Building a PubMed knowledge graph. *Sci Data*, 7(1):205.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. [KG-BERT: BERT for Knowledge Graph Completion](#). *arXiv e-prints*, page arXiv:1909.03193.

- Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. [Keyphrase extraction using deep recurrent neural networks on Twitter](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845, Austin, Texas. Association for Computational Linguistics.
- Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.
- Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. [Extractive summarization as text matching](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.
- Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. [Neural document summarization by jointly learning to score and select sentences](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.
- X. Zhu, C. Lyu, D. Ji, H. Liao, and F. Li. 2020. Deep neural model with self-training for scientific keyphrase extraction. *PLoS One*, 15(5):e0232547.