INNOVATORS at SemEval-2021 Task-11: A Dependency Parsing and BERT-based model for Extracting Contribution Knowledge from Scientific Papers

Hardik Arora†, Tirthankar Ghosal*, Sandeep Kumar†, Suraj Patwal†, Phil Gooch‡
†Indian Institute of Technology Patna, India
*Institute of Formal and Applied Linguistics, Charles University, Czech Republic
†(hardik_1901ce15,sandeep_1911mc12,suraj_1911mt12)@iitp.ac.in
*ghosal@ufal.mff.cuni.cz, ‡phil@scholarcy.com

Abstract

In this work, we describe our system submission to the SemEval 2021 Task 11: NLP Contribution Graph Challenge. We attempt all the three sub-tasks in the challenge and report our results. Subtask 1 aims to identify the contributing sentences in a given publication. Subtask 2 follows from Subtask 1 to extract the scientific term and predicate phrases from the identified contributing sentences. The final Subtask 3 entails extracting *triples* (subject, predicate, object) from the phrases and categorizing them under one or more defined information units. With the NLPContributionGraph Shared Task, the organizers formalized the building of a scholarly contributions-focused graph over NLP scholarly articles as an automated task. Our approaches include a BERT-based classification model for identifying the contributing sentences in a research publication, a rule-based dependency parsing for phrase extraction, followed by a CNN-based model for information units classification and a set of rules for triples extraction. The quantitative results show that we obtain the 5^{th}, 5^{th}, and 7^{th} rank respectively in three evaluation phases. We make our codes available at https://github.com/HardikArora17/SemEval-2021-INNOVATORS

1 Introduction

Thousands of papers are published by the scientific community every day. It is now increasingly becoming difficult to browse the huge pool of papers to identify relevant work and thereby keep up with the latest research findings. Scientific literature is growing at an exponential rate and researchers today face the problem to identify the latest state-of-the-art contributions. Keeping track of recent advancements is becoming a tedious exercise, if not practically impossible. The Open Research Knowledge Graph (ORKG) (Jaradeh et al., 2019) is posited as a solution to keeping track of research progress minus the cognitive overload that reading dozens of full papers imposes. It aims to build a comprehensive knowledge graph that publishes scholarly publications' research contributions per paper, where the contributions are interconnected via the graph even across documents.

As described in D'Souza et al. (2021), with the ORKG comparisons feature, researchers are no longer faced with the daunting cognitive ingestion obstacle from manually scouring through dozens of papers of unstructured content in their field. This process traditionally would take several days or months; using the ORKG contributions comparison tabulated view, the task is reduced to just a few minutes. Assuming the individual paper contributions are structured in the ORKG, they can then deconstruct the graph, tap into the aspects they are interested in, and can enhance it for their purposes. Further, they can select multiple such paper graphs and click a button to generate their tabulated comparison. This presents an opportunity to enhance content ingestion enabled via their fine-grained machine interpretability by transforming scholarly articles into knowledge-based information flows by representing and expressing information through semantically rich, interlinked knowledge graphs (Auer et al., 2018).

In this paper, we present our approach for the three sub-tasks in the NLP Contribution Graph Challenge. Our contribution are as follows:

1. Fine tuning BERT for contributing sentences (a set of sentences about the contribution in the article).

2. A rule-based approach for extracting scientific and phrases (a set of scientific terms and relational cue phrases extracted from the contributing sentences; for each paper) using

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 502–510
Bangkok, Thailand (online), August 5–6, 2021.

dependency parsing.

3. A CNN-based architecture for classifying sentences to 12 information units followed by rules to generate triples (semantic statements that pair scientific terms with a relation, modeled toward subject-predicate-object RDF statements for KG building).

The rest of this paper is organized as follows: Section 2 briefly summarizes some related works similar to this task followed by the problem statement of this task. Section 4 describes the details of the data provided by the organizers. Section 5 and 6 presents the details of our model for all three phases of the task, including the structure and its implementations, along with results and experimental details. The conclusions and the directions for the future research are provided in Section 8.

2 Related Work

Although this is a relatively new challenge, we found some related investigations in the literature. Vogt et al. (2020) proposed a novel semantic data model for modeling the contribution of scientific investigations of three domains, viz. Medicine, Computer Science, and Agriculture. The model includes a schema of relevant concepts highlighting six core information units, viz. Objective, Method, Activity, Agent, Material, and Result. They introduced the idea of building blocks called Knowledge Graph Cells for its knowledge graph application.

Gupta and Manning (2011) introduced a new categorization of key aspects of scientific articles, which is (1) FOCUS: main contribution, (2) TECHNIQUE: method or tool used, and (3) DOMAIN: application domain. They extracted the aspects by matching semantic patterns to dependency trees and learn the patterns using bootstrapping. They also present a case study on the computational linguistics community using the three aspects extracted from its articles, verifying our system's results and showing novel results for the dynamics and the overall influence of computational linguistics subfields.

Hayashi et al. (2020) introduced a new task of disentangled paper summarization to generate summaries for the paper contributions and the work context to help identify the key findings shared in articles.

Rusu et al. (2007) presented an approach to extracting subject-predicate-object triplets from English sentences. They used four different well-known syntactic parsers for English to generate parse trees from the sentences, followed by extraction of triplets from the parse trees using parser-dependent techniques. A machine learning approach has been used by Dali and Fortuna (2008) to extract subject-predicate-object triplets from English sentences. Support Vector Machine (SVM) is used to train a model on human annotated triplets, and the features are computed from three parser.

3 Problem Definition

The problems are defined by the shared task organizers.

1. For Phase-1 (End-to-end Pipeline), given a scientific paper we have to output contributing sentence S_1,S_2,$S_3 \ldots S_{|n|}$ (where n is the number of contributing sentences present in the document), scientific term and predicate phrases from the contributing sentences and finally triples information for particular information units.

2. For Phase-2, Part 1 (Phrases and Triples), we are provided with gold annotated contributing sentences and we have to output scientific terms and predicate phrases from the contributing sentences.

3. Lastly, for Phase 2, Part 2 (Triples Extraction), along with the gold annotated contributing terms, the gold-labeled scientific term and predicate phrases are also provided. We have to output the triplets information for particular information units.

Table 2 shows the three sub-tasks with an example.

4 Dataset Description

D'Souza et al. (2021) released the data for this task. We are provided with a training set of 55084 sentences, taken from 236 annotated papers from across 24 various fields in the NLP domain (such as natural language inference, question answering, sarcasm detection, etc.). Out of these, 5084 comes under the category of contributing sentences, and the remaining are non-contributing sentences. Triples are organized into three (minimum) up to 12 information units (Research Problem, Approach, Model, Code, Dataset, Experimental Setup, Hyperparameters, Baselines, Results, Tasks, Experiments, and Ablation Analysis). The detailed description of

Information unit	Description	Example
Research Problem	It determines the research challenge addressed by a contribution using the predicate has ResearchProblem.By definition, it is the focus of the research investigation, in other words, the issue for which the solution must be obtained.	A Question - Focused Multi- Factor Attention Network for Question Answering
Approach or Model	Essentially, this is the contribution of the paper as the solution proposed for the research problem.	"More specifically , unlike existing models where the query attention is applied either token - wise or sentence - wise to allow weighted aggregation , the Gated - Attention (GA) module proposed in this work allows the query to directly interact with each dimension of the token embeddings at the semantic - level , and is applied layer - wise as information filters during the multi-hop representation learning process ." & First , it is embedding - agnostic , meaning that one of the main (and perhaps most important) hyperparameters in NLP pipelines is made obsolete .
Code	It is the link to the software on an opensource hosting platform such as Gitlab or Github or on the author's website.	We compute a vector gate as a linear projection of the token features followed 1 Code is available at https://github.com/kimiyoung/fg-gating 1 ar Xiv: 1611.01724v2 [cs.CL] 11 Sep 2017
Dataset	This is another aspect of the contribution solution in the form of a dataset.	To address this , this paper introduces the Stanford Natural Language Inference (SNLI) corpus , a collection of sentence pairs labeled for entailment , contradiction , and semantic independence .
Experimental Setup or Hyperparameters	Includes details about the platform including both hardware (e.g., GPU) and software (e.g., Tensorflow library) for implementing the machine learning solution; and of variables, that determine the network structure (e.g., number of hidden units) and how the network is trained (e.g., learning rate), for tuning the software to the task objective. It is called Experimental Setup when hardware details are provided, otherwise Hyperparameters.	We used pre-trained 300D Glove 840B vectors to initialize the word embeddings . & This takes two days using Tensorflow and a single NVIDIA K80 GPU . provide an official evaluation script that allows us to measure F 1 score and EM score by comparing the prediction and ground truth answers .
Baselines	They are the listed systems that a proposed Approach or Model is compared against.	(5) BM25 : BM25 is a bag - of - words retrieval function that ranks a set of reviews based on the question terms appearing in each review .
Results	The main findings or outcomes reported in the article text for the ResearchProblem.	Overall , we observe a significant improvement with all three configurations , effectively showing the benefit of training a QA model in a semisupervised fashion with a large language model .
Tasks	The Approach or Model, particularly in multi-task settings, are tested on more than one task, in which case, we list all the experimental tasks. The experimental tasks are often synonymous with the experimental datasets since it is common in NLP for tasks to be defined over datasets. And where lists of Tasks are concerned, the Tasks can include the ExperimentalSetup as a sub information unit.	All the above subtasks have been modeled as binary classification problems : kernel - based classifiers are trained and the classification score is used to sort the instances and produce the final ranking .
Experiments	It is a container information unit that includes one or more of the previous discussed units as sub information units. Can be combination of lists of Tasks, ExperimentalSetup and Results, or a combination of Approach, ExperimentalSetup and Results.	The temperature parameter ? of Gumbel - Softmax is set to 1.0 , and we did not find that temperature annealing improves performance .
Ablation Analysis	It is a form of Results that describes the performance of components in an Approach or Model.	In , we removed dense connections over both co-attentive and recurrent features , and the performance degraded to 88.5.

Table 1: Information units and their corresponding definitions [1]

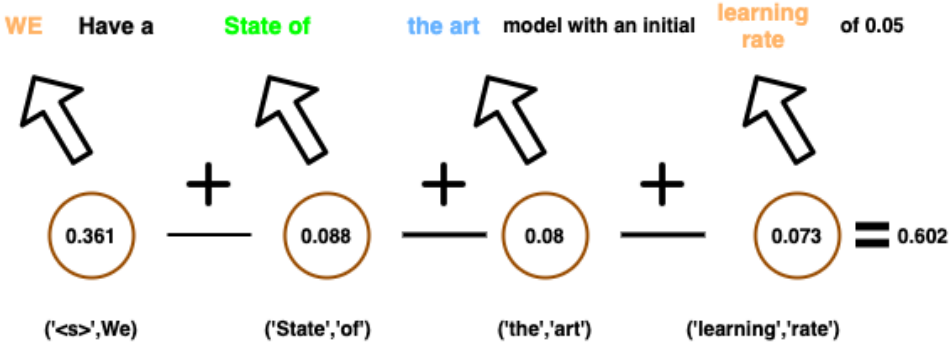

Figure 1: Bigram filtering to prune non-contributing sentences

Sentence:
We used the BERTBASE model pre-trained on English Wikipedia and BooksCorpus for 1M steps.
Scientific Term and Predicate Phrases:
used
BERTBASE model
pre-trained on
English Wikipedia
BooksCorpus
for
1M steps
Triples:
(C, has, ES)
(ES, used, BERTBASE model)
(BERTBASE model, pre-trained on, English Wikipedia)
(BERTBASE model, pre-trained on, BooksCorpus)
(BERTBASE model, for, 1M steps)

Table 2: Example of a Contributing Sentence, corresponding Scientific Term and Predicate Phrase, and extracted Triples

these information units is shown in Table 1. Overall, the annotated corpus contains 2631 triples (avg. of 52 triples per article). Its data elements comprise 1033 unique subjects, 843 unique predicates, and 2182 unique objects. Of all tasks, relation classification has the highest number of unique triples (544) and named entity recognition the least (473).

5 Proposed Approach

We describe our approach for all three phases of the competition as follows:-

5.1 Phase-I: Identifying Contributing Sentences

5.1.1 Sentence Filtering

Initially, we use the Scholarcy API[2] to do some preliminary data analysis to understand some essential information (key concepts, highlighted sentences) in the challenge corpus.

To reduce the data-imbalance ratio of contributing and non-contributing sentences, we filter out most non-contributing sentences. We employ a simple bi-gram filtering to achieve this. We extract all the bi-gram pairs from the entire training corpus and assign each bi-gram pair a score(number of times the bi-gram pairs occurs in the corpus divided by 1000), based on which we set a threshold[3] and filter out the sentences. After filtering, 37.4% non contributing sentences are removed while only 7.07% of contributing sentences are filtered. The example in Figure 1 explains our approach.

We have a state of the art model with an initial learning rate of 0.05,

[2] https://www.scholarcy.com/

[3] see our Github link mentioned in the abstract for details

Bigram tuples	Score
('<s>', 'We')	0.361
('<s>', 'The')	0.267
('of', 'the')	0.262
('on', 'the')	0.174
('in', 'the')	0.148
('<s>', 'In')	0.147
('to', 'the')	0.11
('with', 'the')	0.101
('and', 'the')	0.098
('state', 'of')	0.088
('the', 'art')	0.088
('with', 'a')	0.081
('the', 'model')	0.08
('our', 'model')	0.079
('learning', 'rate')	0.073
('<s>', 'Our')	0.071
('for', 'the')	0.068
('We', 'use')	0.068
('set', 'to')	0.064
('<s>', 'For')	0.063
('In', 'this')	0.058
('that', 'the')	0.058
('of', '%')	0.053
('word', 'embeddings')	0.053
('rate', 'of')	0.051
('natural', 'language')	0.05
('we', 'propose')	0.05
('is', 'a')	0.0.049
('from', 'the')	0.048
('this', 'paper')	0.047
('the', 'performance')	0.047
('based', 'on')	0.0.046
('<s>', 'To')	0.046
('is', 'set')	0.046
('question', 'answering')	0.044
('which', 'is')	0.044
('number', 'of')	0.044
('and', 'a')	0.042
('use', 'the')	0.042
('<s>', 'This')	0.04
('of', '< e >')	0.04
('batch', 'size')	0.04
('the', 'best')	0.039

Table 3: The topmost Bigram scores in non-increasing order; here(<s> denotes start token <e> denotes end token

From our generated list of bi-gram scores:

(< s >, 'We'), has a score of 0.308, ('state', 'of ') has a score of 0.088 and so on. The total score is then calculated by summing the score of individual bi-grams.

5.1.2 Classification Model

Input is the sequence of filtered sentences S = S_1, ..., S_n, where n<=10. We set a threshold of 10 sentences on the number of sentences per sequence as released BERT pretrained weights support sequences of up to 512 word-pieces (Wu et al., 2016). The standard [CLS] is inserted as the first token of the sequence, and another delimiter token [SEP] is used for separating the segments.

After processing each sentence, we feed it into a SciBERT model (Beltagy et al., 2019) which is a variant of BERT, trained on scientific papers, as shown in Figure 2. The pre-training task of BERT (Devlin et al., 2019a) depends on two unsupervised sub-tasks: masked language modeling

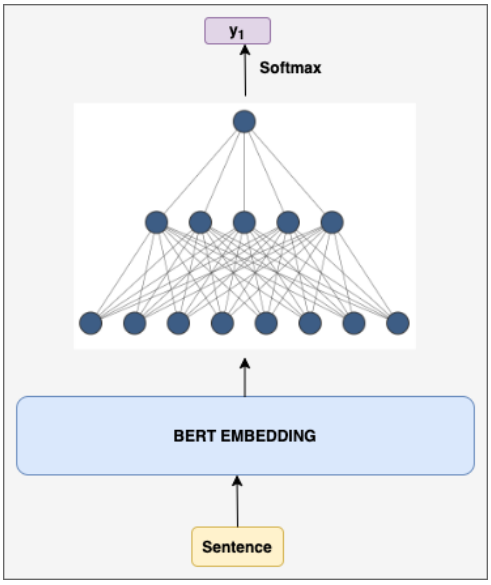

Figure 2: Classification of Contributing and Non-Contributing using a pre-trained SciBERT model

Model	F1 (with filtering)	F1 (w/o filtering)
CNN+Glove	0.3123	0.1347
$Bert_{base}$	0.3578	0.1681
Our model	0.3987	0.1872

Table 4: Result of classifiction to contributing sentences, all are F1 scores

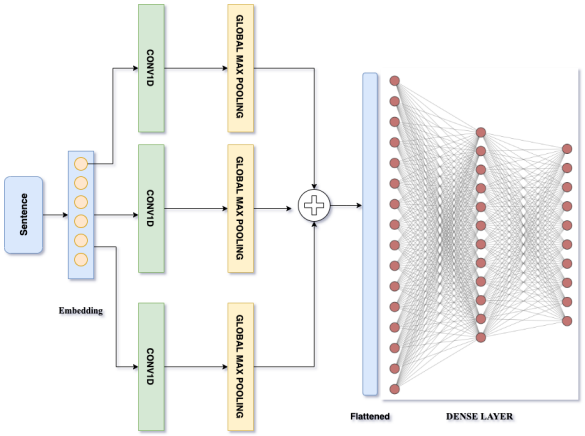

Figure 3: Architecture of classification of sentence into corresponding information units

(MLM) and next sentence prediction(NSP). These two sub-tasks use the same model architecture but with different input patterns and different output layer. In MLM, a fixed amount of tokens of the input sequences is masked, and the model is trained for predicting the original tokens of the masked tokens. In NSP, the model has to predict whether two sequences of text are naturally following each other or not. 50% data is generated automatically by taking sentence pairs next to each other, and the other 50% is generated by taking sentence pairs randomly from the unlabeled corpus. The initial input embedding(E_{Tok}) is calculated by summing up the token, sentence and positional embedding. In the case of MLM, the final hidden vector of each of the masked tokens is passed to a softmax classifier(output layer) to predict the original token. On the other hand, during NSP, the final hidden vector(C) of the [CLS] token is fed to a binary classifier(output layer) to predict whether the input pair is following each other or not.

In the fine-tuning part for downstream tasks, we use the encoding of the [SEP] tokens to classify each sentence. The transformer layers (Devlin et al., 2019a) allows the model to fine-tune the weights of these special tokens according to the task-specific training data. We use a multi-layer feed forward network on top of the [SEP] representations of each sentence to classify them to the categories(is contributing or not?). During fine-tuning, the model learns appropriate weights for the [SEP] token to capture contextual information and learn sentence structure and relations between continuous sentences (through the next sentence objective). Further, we use a softmax classifier on top of the MLP to predict the label's probability.

5.1.3 Experimental Setup

We perform all our experiments on a GPU (GeForce RTX 2070) with 8 GB of memory. In phase-1,for contribution and non-contributing sentence classification task, we use the AllenNLP (Gardner et al., 2018) toolkit for the model implementation. As in prior work (Devlin et al., 2019b),for training we use the dropout of 0.1, the Adam optimizer for 2-5 epochs, and learning rates of 5e-6, 1e-5, 2e-5, or 5e-5.

5.2 Phase-II, Part 1: Phrase and Triples Extraction

Our system used an unsupervised rule-based system for extracting scientific entities and their predicates from the contributing sentences. As part of our initial experiments, we tried existing keyword extraction models such as RAKE, but they did not produce good results (e.g. F1 of 0.1062), as they are not tuned to this dataset. Given the paucity of training data for this task, we built a rule-based model for phrase extraction. We used the spacy

Phase	Avg F1	Sentences	Phrases span only	Information units	Triples + all units
Phase-1	0.3205	0.3987	0.1563	0.7172	0.0097
Phase-2, Part-1	0.5252	1.00	0.3740	0.7172	0.0097
Phase-2, Part-2	0.5971	1.00	1.00	0.3472	0.0413
Top-performing System					
Phase-1	0.4703	0.5941	0.4522	0.7293	0.1379
Phase-2, Part-1	0.7612	1.00	0.7857	0.8249	0.4344
Phase-2, Part-2	0.8594	1.00	1.00	0.8249	0.6129

Table 5: Our results for the three phases. Note: All scores in the table are F1 scores

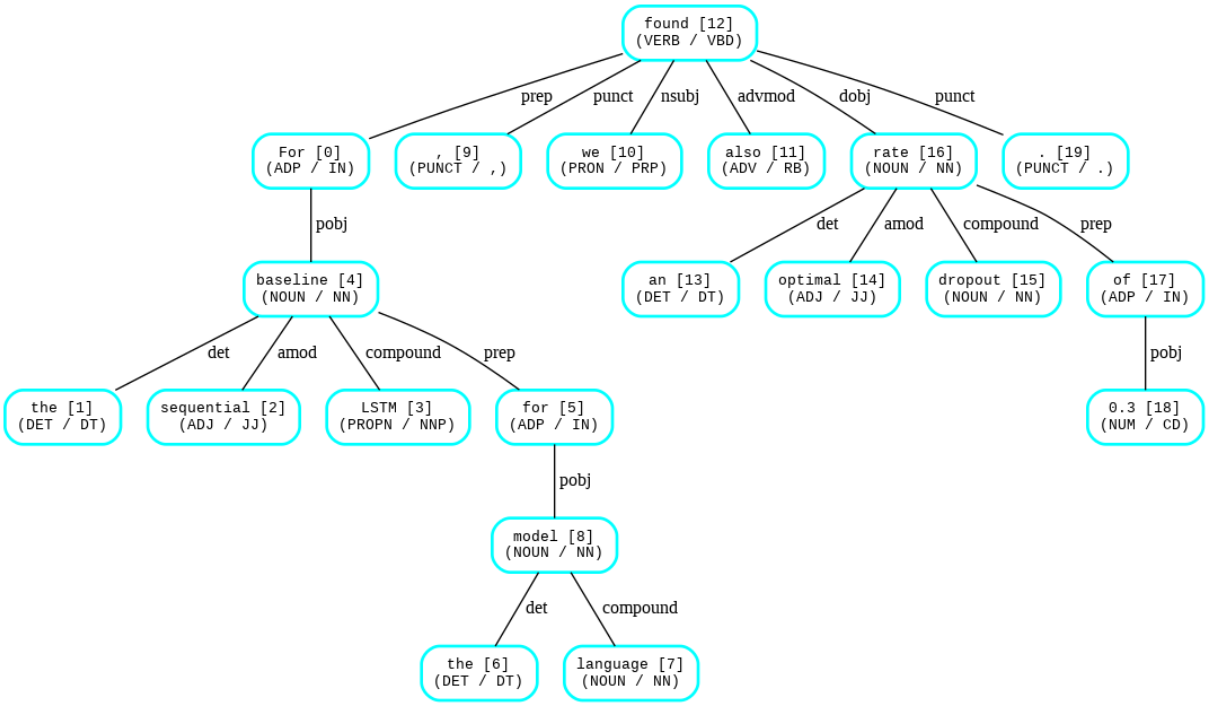

Figure 4: Example showing dependency parsing tree of a sentence

Non-contributing sentences misclassified as contributing sentences.	Reason
A model only achieved an F 1 score of 86.5 on our development set , that is over 2 points lower than the 88.7 of a LSTM + A+D model.	Our model fails to differentiate between a general sentence that gives essential information but has no relation to the paper's model
As a by-product of our investigation , a variant of the RNNG without ensembling achieved the best reported supervised phrase - structure parsing (93.6 F 1 ; English PTB) and , through conversion , dependency parsing (95.8 UAS , 94.6 LAS ; PTB SD).	Our model misclassified those sentences as contributing sentence which contains a large number of scientific terms
The elements of the stack that comprise the current constituent (going back to the last 2 https://github.com/clab/rnng/tree/	It misclassified those sentences as contributing, which contains links that are not explicitly related to the paper

Table 6: Error analysis of Phase-1 (contributing sentence)

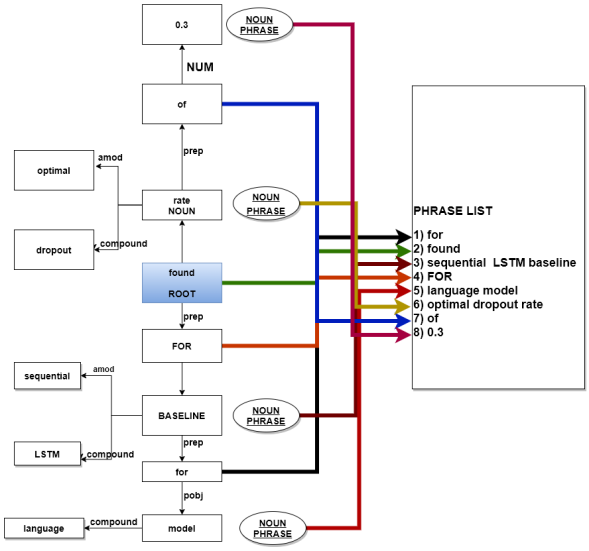

Figure 5: Illustration of addition of phrases to phrase list following rules for the sentence in Figure 4

True phrases	Predicted phrases
both models	models
tuned	tuned
dropout rate	dropout rate
to maximize	maximize
validation set likelihood	validation set likelihood
obtaining	
optimal rates	optimal rates
of	For
0.2 (discriminative)	
0.3 (generative)	

Table 7: Error analysis of Phase-2 (Phrase Extraction)

dependency parser[4] to generate a dependency tree for each contributing sentence. An example tree for the sentence, *"For the sequential LSTM baseline for the language model, we also found an optimal dropout rate of 0.3,"* is shown in Figure 4. The rules specified are in such a format that considers some of the minute details, like if a word with "nsubj" dependency tag is a pronoun occurring at the start of the sentence, then the word should not be included in the phrase. We observed that the "ROOT" of the parsing tree(here: found) is primarily a constituent of the phrase list for that sentence, so we appended it. Next, we started exploring some particular child nodes of the "ROOT" node (mainly with dependency tags "dobj," prep," "advmod") as we tried to extract and form a proper noun, verb phrases, as shown in Figure 4. As soon as a child node with "NOUN" tag is found, complete noun phrase, is appended to the phrase list. At any level, if the child node(Cn) is a modifier(dependency tag' prep'), then we will individually append it to the phrase list, followed by the subsequent exploration of the following hierarchy child nodes of Cn. This procedure continues recursively until we reach the leaf nodes for a branch. More details of various rules created by us for our unsupervised algorithm can be found in our shared source code.

5.3 Phase-II, Part 2: Triples Extraction

As there was little explicit section information in the provided parsed corpus, we classified each sentence into the 12 information units described above. We used a simple CNN-based neural architecture, and the model structure is shown in Fig 3. The first layer is the data input layer, followed by the embedded layer, which creates a numerical representation of the textual data. Then there are three parallel CNN models, each of which has a double convolution means a combination of a convolution layer and a pooling layer. To fully consider each word's information before and after, extract the size of the local characteristics of different sizes of 2x100,3x100,4x100. After the processing of the convolution layer, the characteristics of text classification are more advantageous. Based on this, the pooling layer is further screened from the global perspective where max pooling is used, followed by the merged layer, dense layer, dropout layer with a rate of 0.2. Finally, the text data is passed to a softmax function for outputting the classification

[4] https://spacy.io/

result.

For triples creation, we created some rules. For research problems, triples are mainly in the form: "Contribution" followed by "has research problem" followed by the statement's scientific entity. For example:

Contribution||has research problem||Text Comprehension, is formed from the sentence "Gated - Attention Readers for Text Comprehension" belonging to a research problem.

We generate the triples in this format with the already extracted scientific phrases for the research problem information unit. Code triples are mainly in the form: "Contribution" followed by "code" followed by the URL of the available source code. For example:

Contribution||Code||https://github.com /mandarjoshi90/coref

We extract the URL using regular expression and generat the triples in the above format. We identified subjects along with their directly linked objects and predicates describing the relation between them for other triples. For example, if there is a sentence,

ST Gumbel - Softmax estimator relaxes the discrete sampling operation to be continuous in the backward pass, thus our model can be trained via the standard backpropagation

we form the triples as:

(Contribution||has||Model), (Model||use||Straight-Through (ST) Gumbel-Softmax estimator)

6 Evaluation

We report the evaluation results of all the three phases on Table 5. Average F1 score was to rank the participants in the competition. Precision, recall, and other details of the competition are available on leaderboard[5] of the competition. We also report the ablation analysis and the effect of filtering sentences in the extraction of contributing sentences of phase-1 in Figure 4. For phases-2 part-1, as were already provided by the gold label contributing sentences, the average F1-score of the contributing sentence is 1. Similarly, for phases-2 part-2, as were already further offered the gold label phrases, the average F1-score of the contributing sentence and the phases is 1. We reported a 0.32 average F1 for phase-1(end to end pipeline), 0.52 for phase-2(phrases and triples extraction) and 0.59 for phase-3(triples extraction).

[5] https://competitions.codalab.org/competitions/25680#results

7 Error Analysis

As shown in Table 6 these sentences are non-contributing but misclassified as contributing sentences.

In phase 2, our model is based on rules, so it fails to capture unique or very different tree structures. For example as shown in Table 7, in sentence:-

"For both models, we tuned the dropout rate to maximize validation set likelihood , obtaining optimal rates of 0.2 (discriminative) and 0.3 (generative).".

The phrases such as *"0.3 (generative)"* and *"0.2 (discriminative)"* was not captured. Our model also captured the single word phrase *"maximize"* instead of the true phrase *"to maximize"*.

8 Conclusion and Future Work

With the NLPContributionGraph Shared Task, we have attempted to formalize the building of a scholarly contributions-focused graph over NLP scholarly articles as an automated task. Results and analysis on the gold test dataset show that our approach performed reasonably well in identifying contributing sentences and phrase extraction. However, we didn't perform well in triples extraction. In the future, we plan to improve the system, especially the triples extraction phase.

References

Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria-Esther Vidal. 2018. Towards a knowledge graph for science. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS 2018, Novi Sad, Serbia, June 25-27, 2018*, pages 1:1–1:6. ACM.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics.

Lorand Dali and Blaz Fortuna. 2008. Triplet extraction from sentences using svm. abs/2011.03161.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: pre-training of

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding.

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.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform.

Sonal Gupta and Christopher D. Manning. 2011. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 1–9. The Association for Computer Linguistics.

Hiroaki Hayashi, Wojciech Kryscinski, Bryan McCann, Nazneen Fatema Rajani, and Caiming Xiong. 2020. What's new? summarizing contributions in scientific literature. *CoRR*, abs/2011.03161.

Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine 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, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*, pages 243–246. ACM.

D. Rusu, Lorand Dali, B. Fortuna, M. Grobelnik, and D. Mladení. 2007. Triplet extraction from sentences.

Lars Vogt, Jennifer D'Souza, Markus Stocker, and Sören Auer. 2020. Toward representing research contributions in scholarly knowledge graphs using knowledge graph cells. In *JCDL '20: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, Virtual Event, China, August 1-5, 2020*, pages 107–116. ACM.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.