Extracting Keyphrases and Relations from Scientific Publications

Project Final Report (Team No. 55)

Team Members -

Aakash Singh – 2021201087

Mayank Mukundam – 2021201057

Sourabh Patidar – 2021201089

Problem Statement -

Scientific publications are essential sources of information for researchers in various fields. However, with the increase in the number of publications, it has become challenging for researchers to keep up with the latest advancements in their respective fields. Key phrase and relation extraction from scientific publications is an essential task in natural language processing, which can help researchers quickly identify relevant articles and extract meaningful insights from them.

The problem can be divided into 3 sub-problems -

1. Key phrase Extraction
2. Classifying the keywords
3. Identifying relationships

The first step is to identify and extract the keywords or key phrases from the plain text. After identifying the keywords or phrases, we need to classify each of them into one of the following 3 categories (PROCESS, TASK and MATERIAL). Once these keywords or phrases are classified into the above categories, the final step is to identify the relationship (if any) between them.

Dataset -

We have used the SemEval 2017 Task 10 dataset. The dataset is a corpus of texts used in the SemEval 2017 shared task on automatic keyword extraction. The dataset contains a collection of scientific articles which contains a wide range of topics including Chemistry, Computer Science and Physics.

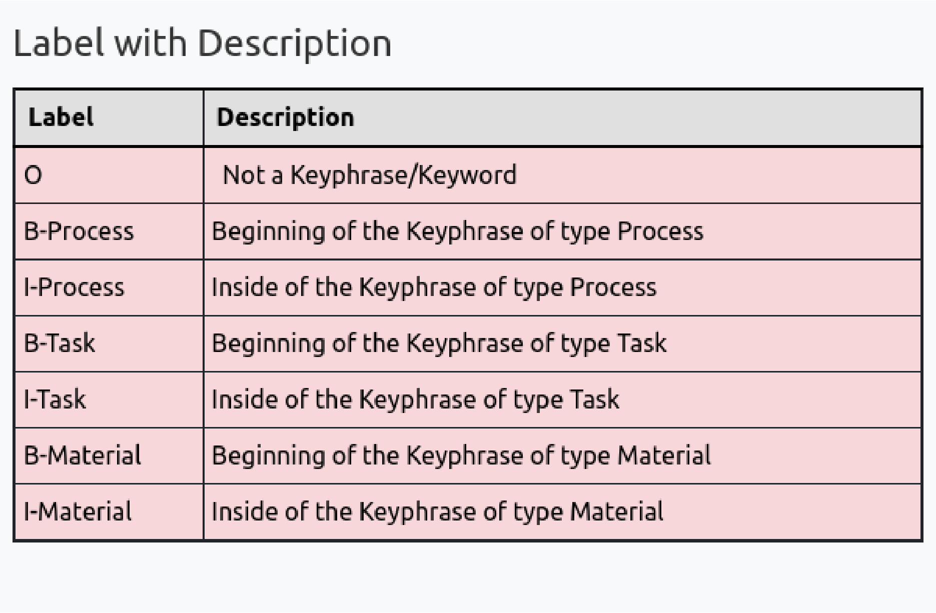
The corpus is divided into three parts: a training set, a development set, and a test set. The training set contains 350 files of each .ann (annotations) and text (.txt) types, while the development set and the test set contain 100 and 50 files of each type respectively. The ".txt" files contain text extract from scientific publications and annotations file contains the keyword or keyphrase boundary and the type of keyword or keyphrase. In addition to this, the annotation file also contains the relationship between a few of the keywords or keyphrases, these relationships are of 2 types, Synonym-of and Hyponym-of.

Methodology -

1. Data pre-processing:

a. For subtask1 and subtask2

In subtask1, we have to extract or identify whether a given token is a Keyword or not and in subtask 2, we need to classify the keywords in one of the 3 types. These three types are Task, Material and Process. We have combined subtask1 and subtask2 as both the tasks are classification tasks and each task is computationally heavy so to utilize the resources efficiently, the decision has been taken and also training 2 models does not make much sense. We have borrowed the BIO scheme of chunk representation to pre-process the dataset for subtask1 and subtask2. BIO in BIO scheme stands for Beginning, Intermediate and Outside. We have taken the whole text from the text file, and we have word tokenized the text. Each word is assigned a label out of the maximum 7 possible labels. The seven possible labels and their description is given below.

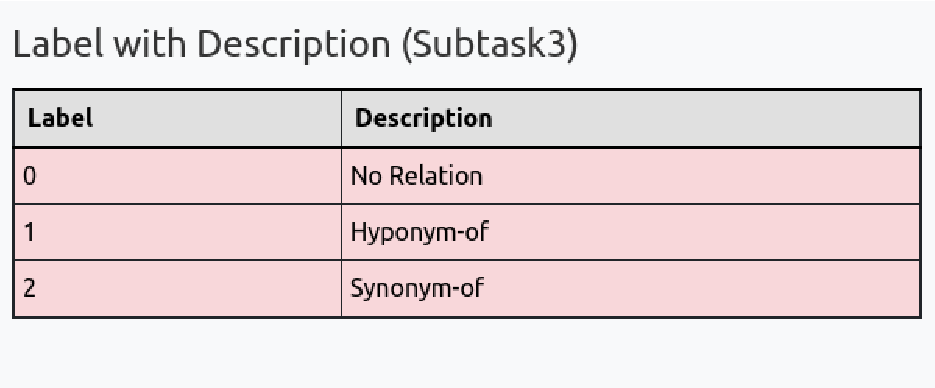


Label with Description

Label	Description
O	Not a Keyphrase/Keyword
B-Process	Beginning of the Keyphrase of type Process
I-Process	Inside of the Keyphrase of type Process
B-Task	Beginning of the Keyphrase of type Task
I-Task	Inside of the Keyphrase of type Task
B-Material	Beginning of the Keyphrase of type Material
I-Material	Inside of the Keyphrase of type Material

b. **For subtask3 -**

For the task of relationship identification, we have extracted the relationship between keyphrases from the annotation file. These relationships are of two types, Synonym-of and Hyponym-of. Here, Synonym-of is two-way relationship and Hyponym-of is only unidirectional so we have considered the reverse relationship between Hyponym as no relation. Hence, we got 3 labels as shown in table.



Label with Description (Subtask3)

Label	Description
0	No Relation
1	Hyponym-of
2	Synonym-of

We have done this preprocessing for each of the training, development and test set.

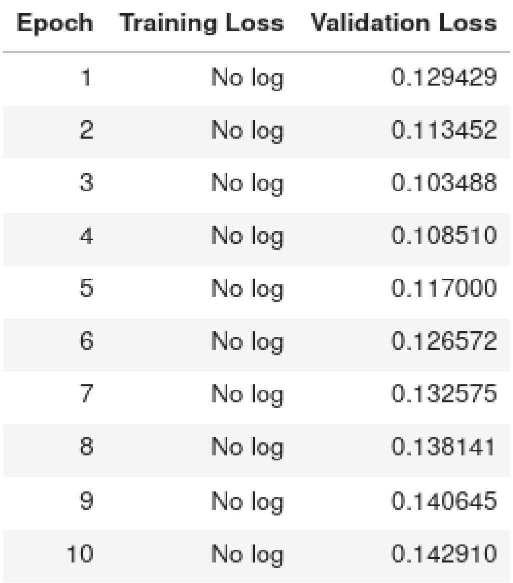
2. Keywords and Keyphrases extraction and Classification -

We have used the SciBERT pretrained model. It is a variation of BERT which is pretrained on scientific data. We have finetuned the model on the SemEval17 Task 10 dataset which has 350 paragraphs. These text data is converted to word tokens, and we have treated this task similar to the Named Entity Recognition (NER) task. So, the task now boils down to the Token Classification task.

We have taken the tokenizer of SciBERT from the AutoTokenizer package from transformers and to load the pretrained SciBERT model we have used the AutoModelForTokenClassification package. We have finetuned the model on 10 epochs for which we got validation accuracy as shown in figure below. We have uploaded the model on huggingface using the push_to_hub parameter of Trainer Argument Class. We have used the learning rate = 0.00002 and the number of epochs is 10. The model which is giving the least loss is saved and used for final predication and evaluation.

To finetune we first tried to finetune on Kaggle GPU itself but due to memory constraints we could not run the training. So, then we decided to use WanDB’s gpu to finetune the model.

Also, we have tried using KeyBERT library for unsupervised Keyphrases extraction but didn’t get the desired results.



Epoch	Training Loss	Validation Loss
1	No log	0.129429
2	No log	0.113452
3	No log	0.103488
4	No log	0.108510
5	No log	0.117000
6	No log	0.126572
7	No log	0.132575
8	No log	0.138141
9	No log	0.140645
10	No log	0.142910

3. Relationship Identification -

We have used the sentence-transformers to find the embedding of each phrase. Embedding for both the entities are calculated using model.encode() function. Each embedding vector is of size 384. Embeddings of both the entities are concatenated to get a vector double the size of original vector (I.e., 768). Now, each pair of entities one of the labels assigned to them out of the 3 labels.

Now, this problems boils down to a supervised classification problem & now we can use any one of the matter we have used SVM Classifiers to label the output as 0, 1 or 2. We have used **MiniLM Sentence Transformer** as our text encoder, where in we have fed both the entities to the model encode method which fetches us the encoding & we have merged both these encodings against their relation & prepared a data frame for train & test.

Next up we trained the SVM classifier on training data & analyzed the predicted results using relevant metrics.

Evaluation & Analysis -

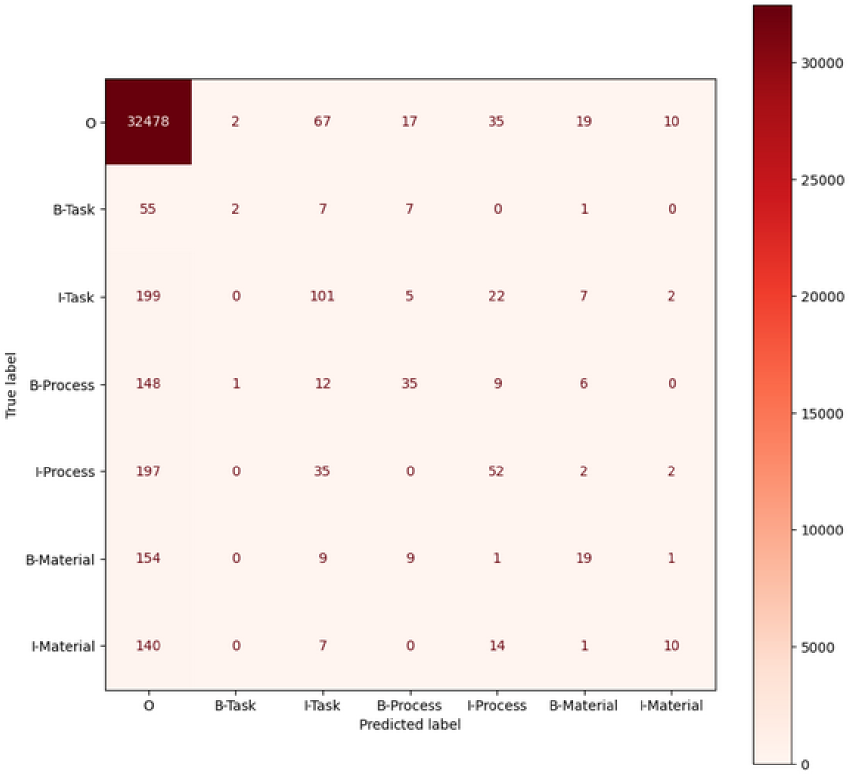
We have predicted the labels on the test set. Below are the results of the evaluation matrices-

1. **For Subtask1 and Subtask2 -**

For subtask1 and subtask2 extract keywords and classify them into one of the 7 labels.

a. Confusion Matrix -

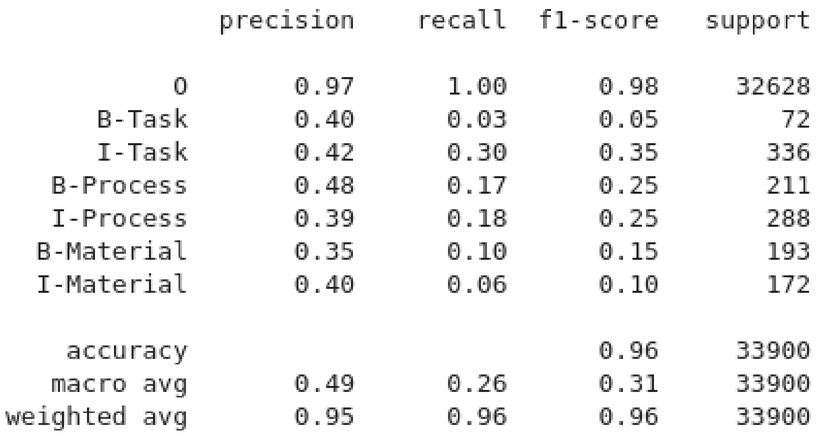
From the confusion matrix, we can see that the keywords are extracted with approx. 50% accuracy and out of this 50% the classification seems fine. The classification is analyzed in the next section.



True label \ Predicted label	O	B-Task	I-Task	B-Process	I-Process	B-Material	I-Material
O	32478	2	67	17	35	19	10
B-Task	55	2	7	7	0	1	0
I-Task	199	0	101	5	22	7	2
B-Process	148	1	12	35	9	6	0
I-Process	197	0	35	0	52	2	2
B-Material	154	0	9	9	1	19	1
I-Material	140	0	7	0	14	1	10

b. **Classification Report -**

The 'O' Class is classified with very high accuracy, all other classes are classified. All other classes are classified with accuracy of 5% to 35% accuracy. The macro f1-score is 0.31 when the tasks are done together.

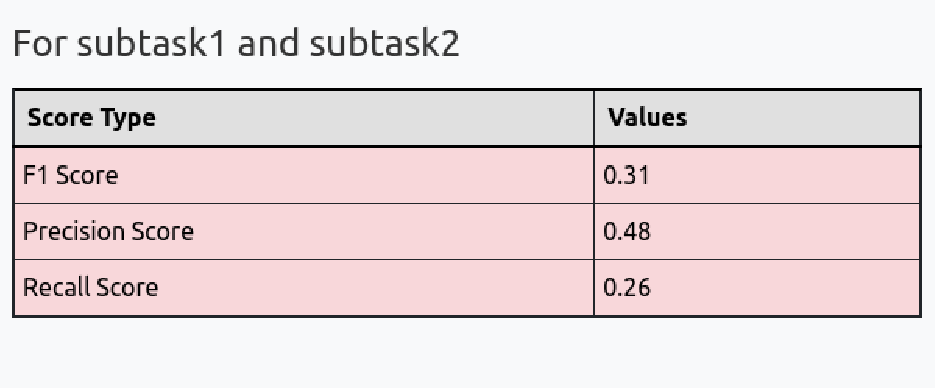


```
              precision    recall  f1-score   support

           O       0.97      1.00      0.98     32628
      B-Task       0.40      0.03      0.05        72
      I-Task       0.42      0.30      0.35       336
   B-Process       0.48      0.17      0.25       211
   I-Process       0.39      0.18      0.25       288
  B-Material       0.35      0.10      0.15       193
  I-Material       0.40      0.06      0.10       172

    accuracy                           0.96     33900
   macro avg       0.49      0.26      0.31     33900
weighted avg       0.95      0.96      0.96     33900
```

c. **Precision, Recall and F1 scores**



For subtask1 and subtask2

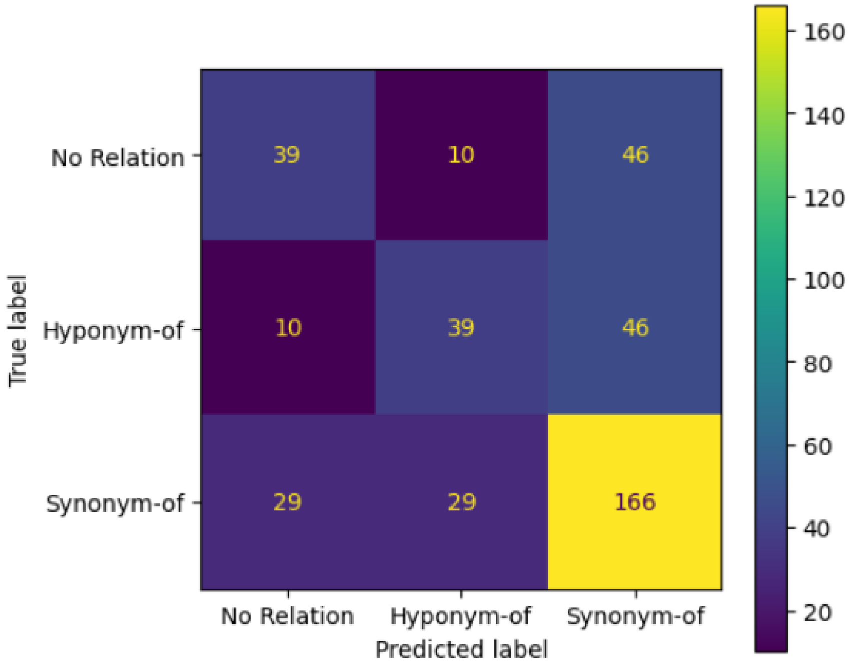
Score Type	Values
F1 Score	0.31
Precision Score	0.48
Recall Score	0.26

2. **For Subtask3 -**

For subtask3, the relationship identification task, the result of the test set is as follows.

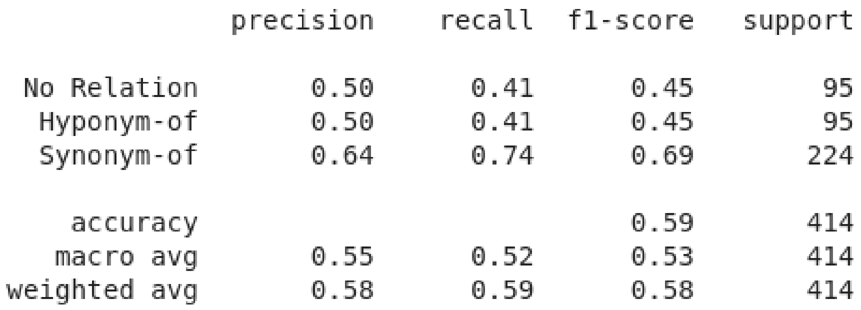
a. **Confusion Matrix -**

From the confusion matrix, we can see that the Synonyms-of class is classified with high accuracy, and the Hyponym-of class and the No-relation class with good accuracy.



b. **Classification Report -**

The macro f1-score is 0.53, the f1-score for Synonym-of is high, so it implies that the classifier is able to classify the synonym-of with accuracy.

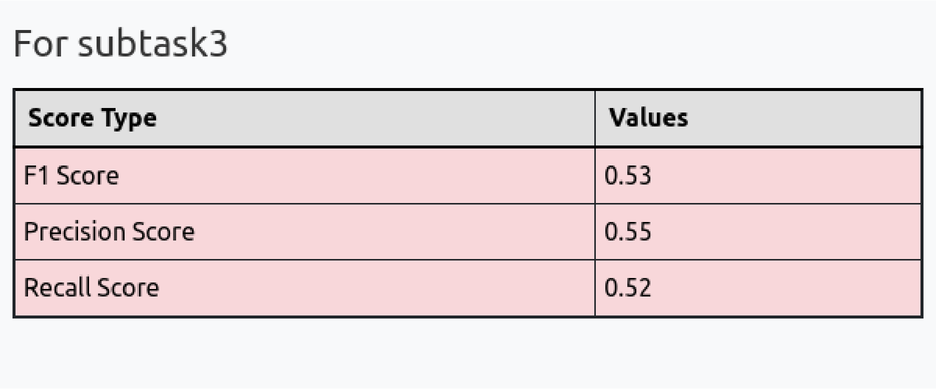


```
              precision    recall  f1-score   support

 No Relation       0.50      0.41      0.45        95
  Hyponym-of       0.50      0.41      0.45        95
  Synonym-of       0.64      0.74      0.69       224

    accuracy                           0.59       414
   macro avg       0.55      0.52      0.53       414
weighted avg       0.58      0.59      0.58       414
```

c. **Precision, Recall and F1 scores -**



For subtask3

Score Type	Values
F1 Score	0.53
Precision Score	0.55
Recall Score	0.52

Conclusion -

The Scientific domain has very few annotated datasets available, SemEval 2017 task 10 was a sweet and short attempt to make dataset available for research purposes in scientific research domain. All the submissions of the task are based on RNNs and LSTMs, so we tried to solve the problem using transformers. For the same purpose we have used the pretrained SciBERT which is a scientific domain variation of BERT to solve the first 2 subtasks and also used ever reliable SVM to solve the 3rd subtask. It is clearly seen that the transformers-based models outperform the traditional RNN methods and SVM too gave decent results.