Course: INTRODUCTION TO NLP

EXTRACTING KEYPHRASES AND RELATIONS FROM SCIENTIFIC PUBLICATIONS

May 5, 2023

Manish Kumar Singh (2020701024)
Nikhil (2021201013)

1 Introduction

A keyphrase refers to a word or group of words that depict the close connection between the content and the context within a document. These phrases can be simple nouns or noun phrases (NPs) that embody the fundamental concepts of the document, such as the topic. Although these tasks share similarities with named entity recognition, named entity classification, and relation extraction, identifying keyphrases presents a more significant challenge than identifying entities like personal names. This is because keyphrases can comprise numerous tokens without clear markers and contextual cues. Additionally, keyphrases often differ significantly between domains.

Scientific publishers are very interested in extracting key terms and identifying relationships between them. This helps recommend articles to readers, alert authors to missing citations, identify potential reviewers for submissions, and analyze research trends over time.

2 Problem Statement

The main focus of our project is to extract keyphrases and their relationships from published scientific articles, including research papers, using various techniques based on neural networks, such as RNN and pre-trained language models. The problem is subdivided into three subtasks to simplify the task, which are as follows:

- Identification of keyphrases at the mention-level
- Classification of keyphrases at the mention-level, with keyphrases types such as PROCESS, TASK and MATERIAL.
  PROCESS: includes keyphrases related to scientific models, algorithms, and processes.
  TASK: comprises keyphrases related to the application, end goal, problem, and task.
  MATERIAL: encompasses keyphrases that identify the resources used in the paper.
- Extraction of mention-level semantic relations between keyphrases of the same keyphrase types, utilizing relation types HYPONYM-OF and SYNONYM-OF.

3 Dataset

SemEval 2017 Task 10:[1] The dataset used in our project was constructed from open-access publications available on ScienceDirect. It includes 500 journal articles, with an even distribution among the domains of Computer Science, Material Sciences, and Physics. Each data instance is a single paragraph of text extracted from a scientific paper. The training

```
print(X_train[45])
```

```
['work', ',', 'light', 'propagation', 'in', 'a', 'scattering', 'medium', 'with', 'piece', '-', 'wise', 'constant', 'r
efractive', 'index', 'radiative', 'transport', 'equation', 'studied', '.', 'sub', '-', 'domain', 'rt', '##e', 'fres',
'##nel', 'reflection', 'and', 'transmission', 'phenomena', '##s', 'sub', '-', 'domains', '.', 'coupled', 'system', 'o
f', 'rt', '##es', 'fem', '.', 'simulations', 'solution', 'of', 'the', 'monte', 'carlo', 'method', '.', 'rt', '##e',
'monte', 'carlo', 'method', '.', 'addition', ',', 'light', '.', 'dot', 'reconstructions', 'refractive', 'index', 'cha
nges', 'image', 'reconstruction', 'procedure', '.']
```

```
print(Y_train[45])
```

```
['O', 'O', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'O', 'O', 'O', 'O', 'O',
'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'O', 'O', 'O', 'O', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I',
'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'O', 'O', 'O', 'O', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I', 'I',
'I']
```

Figure 1: Input and levels for Task-A.

data subset of the corpus comprises 350 documents, while 50 documents are reserved for development and another 100 for testing purposes.

Some noteworthy characteristics of the datasets include:-

- The dataset contains many long keyphrases. Approximately 22% of all keyphrases in the training set consist of five or more tokens, making the task of keyphrase identification challenging.
- However, 93% of these long keyphrases are noun phrases, which can be useful for simple heuristics to identify keyphrase candidates.
- Additionally, 31% of the keyphrases present in the training dataset occur only once, making it difficult for systems to generalize well to unseen keyphrases.

4 Methodology and Evaluation:

Our project involves predicting different types of tokens for each of the three subtasks as follows:-

Subtask A:taskA = O, I for tokens outside, or inside a keyphrase. It is a token classification Task 1. **Subtask B:** taskB = 'O', 'B-Material', 'B-Process', 'B-Task', 'I-Task', 'I-Material', 'I-Process' for tokens outside a keyphrase, or being part of a material, process, or task and so on. It is a token Classification task2. **Subtask C**: taskC = O, S, H for Synonym-of and Hyponym-of relations. For Subtasks A and B, we predict one output label for each input token, while for Subtask C, we predict a vector for each token that encodes

Models	EM
TextRank	0.28
TopicRank	0.26
YAKE	0.21
KeyBERT	0.18

Table 1: Performace of models (unsupervised) on SubTask-A.

the relationship between that token and every other token in the sequence for the first token in each keyphrase. Once the predictions for tokens are obtained, we convert them back to spans and relations between them in a post-processing step.

For the baseline, we implemented an LSTM model, and for the baseline+, we plan we implement a model based on BERT [3].

4.1 Evaluation:

We will automatically evaluate our model's accuracy of keyphrases, keyphrase types, and relations using exact match (EM) and F1 scores.

- **Exact Match(EM):** measures the percentage of documents where the predicted answer is identical to the correct answer. if the characters of the answer predicted by the model exactly match the characters of any of the ground truths, EM = 1, otherwise EM = 0.
- **F1:** captures the precision and recall that the words selected as part of the answer are actually part of the ground truths.

5 Experiment And Results:-

5.1 For SubTask-A:

We tried two different Techniques, Unsupervised and Supervised.

For **unsupervised** Keyphrases extraction from the annotated text data, we used TextRank, TopicRank, YAKE and KeyBERT

- **TextRank:** The TextRank keyword extraction algorithm extracts keywords using a part-of-speech tag-based approach to identify candidate keywords and scores them using word co-occurrences determined by a sliding window.
- **TopicRank:** is a graph-based keyphrase extraction algorithm that uses the longest noun phrases in documents to group them into topics. It builds a graph using topics

```
print(X_train[45])
```

```
['work', ',', 'light', 'propagation', 'in', 'a', 'scattering', 'medium', 'with', 'piece', '-', 'wise', 'constant', 'r
efractive', 'index', 'radiative', 'transport', 'equation', 'studied', '.', 'sub', '-', 'domain', 'rt', '##e', 'fres',
'##nel', 'reflection', 'and', 'transmission', 'phenomena', '##s', 'sub', '-', 'domains', '.', 'coupled', 'system', 'o
f', 'rt', '##es', 'fem', '.', 'simulations', 'solution', 'of', 'the', 'monte', 'carlo', 'method', '.', 'rt', '##e',
'monte', 'carlo', 'method', '.', 'addition', ',', 'light', '.', 'dot', 'reconstructions', 'refractive', 'index', 'cha
nges', 'image', 'reconstruction', 'procedure', '.']
```

```
print(Y_train[45])
```

```
['O', 'O', 'B-Task', 'I-Task', 'I-Task', 'I-Task', 'I-Task', 'I-Task', 'I-Task', 'I-Task', 'I-Task', 'I-Task', 'I-Tas
k', 'I-Task', 'I-Task', 'B-Process', 'I-Process', 'I-Process', 'O', 'O', 'O', 'O', 'O', 'B-Process', 'I-Process', 'B-
Process', 'I-Process', 'I-Process', 'I-Process', 'I-Process', 'I-Process', 'I-Process', 'O', 'O', 'O', 'O', 'B-Proces
s', 'I-Process', 'I-Process', 'I-Process', 'I-Process', 'B-Process', 'I-Process', 'B-Process', 'B-Material', 'I-Mater
ial', 'I-Material', 'I-Material', 'I-Material', 'I-Material', 'I-Material', 'B-Process', 'I-Process', 'B-Material',
'I-Material', 'I-Material', 'I-Material', 'O', 'O', 'O', 'O', 'B-Process', 'I-Process', 'B-Process', 'I-Process', 'I-
Process', 'B-Process', 'I-Process', 'I-Process', 'I-Process']
```

Figure 2: Input and levels for Task-B.

as vertices and adds edges between each vertex, using a weight proportional to how close the two topics appear in the document. The algorithm eventually outputs one representative key phrase from each of the selected topics.

- **YAKE:** is a statistical keyphrase extraction model that tokenizes documents into word tokens and computes statistics for each. These statistics are combined into a final score, which returns the words with the highest score and combines adjacent words with high scores into key phrases.

- **KeyBERT:** creates BERT embeddings of document texts with predefined lengths, then calculates cosine similarities between document and keyphrase embeddings to extract keyphrases that best describe the entire document.

The results are shown in Table1.

For Supervised training, we used LSTM for baseline and BERT for baseline++. In Baseline++, we first experimented with the BERT-base model and then with the sci-BERT-base[2] model, which is trained on scientific papers. The results are shown in Table2

5.2 For SubTask-B:

Used LSTM for baseline and BERT for baseline++. In Baseline++, we first experimented with the BERT-base model and then with the sci-BERT-base[2] model, which is trained on scientific papers. The results are shown in Table3.

Models	Precision	Recall	F1 score
LSTM-model	0.41	0.36	0.39
BERT-base-model	0.37	0.56	0.45
Sci-BERT-base-model	0.53	0.59	0.56

Table 2: Performace of models on SubTask-A.

Models	Precision	Recall	F1 score
LSTM-model	0.23	0.20	0.21
BERT-base-model	0.30	0.45	0.34
Sci-BERT-base-model	0.38	0.49	0.42

Table 3: Performace of models on SubTask-B.

5.3 For Task-C:

We have used an unsupervised method to find the HYPERNYMS and SYNONYMS pairs in the input sentences using WordNet. WordNet is a lexical database for the English language that is included in the Natural Language Toolkit (NLTK). It provides access to many English words and their semantic relationships, such as synonyms, antonyms, hypernyms and meronyms. Fig˜3 shows an example text from train data and extracted pairs of synonyms and hypernyms.

6 Analysis and Observations:

Sci-Bert pre-trained on scientific dataset outperforms Bert and LSTM model in both the experiments by a huge margin. This shows Sci-Bert is able to transfer what it learned in pre-training to keyphrases extraction and classification tasks as the pre-training data is similar to SemEval 2017 Task-10 data. However, still, the scores are low. One of the main reasons is the less amount of training data. It has only 350 paragraphs for training. Another reason which makes this task challenging is that many annotated Key phrases contain punctuation and special characters. Notably, the dataset contains many long keyphrases. 22% of all keyphrases in the training set consists of words of 5 or more tokens. This contributes to making the task of keyphrase identification very challenging. However, 93% of those keyphrases are noun phrases, which is valuable information for simple heuristics to identify keyphrase candidates. For Task-C we manually analysed 80 samples out of 350 training samples. We found no annotation of the relationship between two keyphrases as SYNONYM labels for HYPERNYM is much less to train a model.

```
print(text)
```

```
Longitudinal beam and target single-spin asymmetries have been at the center of the attention lately, since they have been measured by th
e HERMES and CLAS experimental Collaborations [1–4] and more measurements are planned. They were originally believed to be signals of the
so-called T-odd fragmentation functions [5], in particular, of the Collins function [6–12]. However, both types of asymmetry can receive
contributions also from T-odd distribution functions [13–16], a fact that has often been neglected in analyses. An exhaustive treatment o
f the contributions of T-odd distribution functions has not been carried out completely so far, especially up to subleading order in an e
xpansion in 1/Q, Q2 being the virtuality of the incident photon and the only hard scale of the process, and including quark mass correcti
ons. It is the purpose of the present work to describe the longitudinal beam and target spin asymmetries in a complete way in terms of le
ading and subleading twist distribution and fragmentation functions. We consider both single-particle inclusive DIS, e+p→e′+h+X, and sing
le-jet inclusive DIS, e+p→e′+jet+X. We assume factorization holds for these processes, even though at present there is no factorization p
roof for observables containing subleading-twist transverse-momentum dependent functions (only recently proofs for the leading-twist case
have been presented in Refs. [17,18]).
```

```
synonym_pairs = find_synonym_pairs(preprocessed_text)
print(synonym_pairs)
```

```
[('longitudinal', 'longitudinal'), ('beam', 'beam'), ('target', 'target'), ('single', 'single'), ('spin', 'spin'), ('spin', 'twist'), ('a
symmetries', 'asymmetry'), ('lately', 'recently'), ('believed', 'consider'), ('odd', 'odd'), ('fragmentation', 'fragmentation'), ('functi
ons', 'function'), ('functions', 'purpose'), ('functions', 'work'), ('function', 'functions'), ('function', 'purpose'), ('function', 'wor
k'), ('types', 'case'), ('asymmetry', 'asymmetries'), ('distribution', 'distribution'), ('q', 'q'), ('process', 'work'), ('process', 'pro
cesses'), ('purpose', 'functions'), ('present', 'present'), ('present', 'presented'), ('work', 'functions'), ('work', 'processes'), ('lea
ding', 'leading'), ('twist', 'twist'), ('inclusive', 'inclusive'), ('e', 'e'), ('pe', 'pe'), ('x', 'x'), ('jet', 'jet'), ('factorizatio
n', 'factorization'), ('proof', 'proofs')]
```

```
hypernym_pairs = find_hypernym_pairs(preprocessed_text)
print(hypernym_pairs)
```

```
[('spin', 'present'), ('spin', 'twist'), ('attention', 'treatment'), ('attention', 'work'), ('called', 'order'), ('called', 'consider'),
('particular', 'fact'), ('function', 'expansion'), ('receive', 'consider'), ('receive', 'assume'), ('fact', 'case'), ('treatment', 'expan
sion'), ('carried', 'work'), ('including', 'consider'), ('present', 'spin'), ('present', 'present'), ('present', 'presented'), ('work',
'proof'), ('work', 'proofs'), ('consider', 'holds')]
```

Figure 3: Synonyms and Hypernyms pair from a text of train data.

7 Conclusion:

In this work, we have tried to address the problem of keyphrases extraction and relations from a scientific publication. First, we used pre-trained model to find keyphrases in input sentences. We observe that the EM scores for those models are very low. Then, We train LSTM and Bert-based models for Subtask A and B. We observed that Bert pre-trained on scientific data outperformed Bert and LSTM model on Task A and B. In Task-C, we used WordNet to find synonyms and hypernyms pairs in a sentence as the annotation for the relation between keyphrases is very less to train a model.

References

[1] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*, 2017.

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *EMNLP*, 2019.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.