SciDocFind: Faceted Ranked Retrieval of Scientific Research Papers

IR Course Project: Group 11

Group members

- Soni Aditya Bharatbhai (20CS10060)
- Likhith Reddy Morreddigari (20CS10037)
- Shreyas Jena (20CS30049)
- Rishi Raj (20CS30040)

Problem Statement and Motivation

- **Input:** A query paper **Q**, a facet **f** and a set of candidate papers **C**.
- **Output/Task:** Ranked retrieval of the candidate papers based on similarity with **Q** with respect to the facet **f**.
- Finer grained control on literature search.
- Explore techniques to improve upon state-of-the-art works.
- Find effective ways to find relevance between query and candidates.
- Essentially we require better document-level embeddings.
- The representation must also consider the search-facet.

Dataset Description

- **Dataset used for evaluation:** CSF-Cube
- The dataset has 50 query-facet pairs each with a set of candidate papers.
- Title and abstract provided for each query/candidate paper.
- Each candidate paper is assigned a relevance score from {0,1,2,3}
- Three facets used for retrieval: background, method and result
- Each sentence in the abstract is assigned a label from { background, method, result, other} depending on which facet the sentence describes

Baselines

- **Abstract level baselines:**
  - SciBERT
  - SPECTER
  - SciNCL
- **Sentence level baselines:**
  - SentBERT-Paraphrased
  - SentBERT-NLI
  - Supervised SimCSE
  - Unsupervised SimCSE

Abstract level baselines

- **SciBERT:**
  - Model architecture same as that of BERT but uses a different vocabulary (SciVocab)
  - Trained from scratch using the S2ORC dataset
  - Input: Abstract of query/candidate paper
  - Output: 768 dimensional embedding for each token
- **SPECTER and SciNCL:**
  - Both leverage citation data to fine-tune SciBERT using different approaches
  - Input: Title + [SEP token] + Abstract
  - Output: 768-dimensional embedding for each token
- CLS embedding used as a dense vector representation of the paper.
- L2 distance between query and candidate embedding used during ranking.
- Candidates ranked in increasing order of L2 distance.

Sentence level baselines

- **SentBERT and SimCSE:**
  - Used to obtain a dense vector representation for a sentence
  - **Input:** Sentences from the abstract of the query/candidate paper
  - **Output:** An embedding corresponding to each input sentence
  - Each sentence of the abstract is encoded separately using the model
  - Two variants for each model used as baselines
- Each sentence in the abstract of the query and candidate paper encoded separately
- Maximum cosine similarity computed between query and candidate sentences
- Ranking done in decreasing order of above similarity score

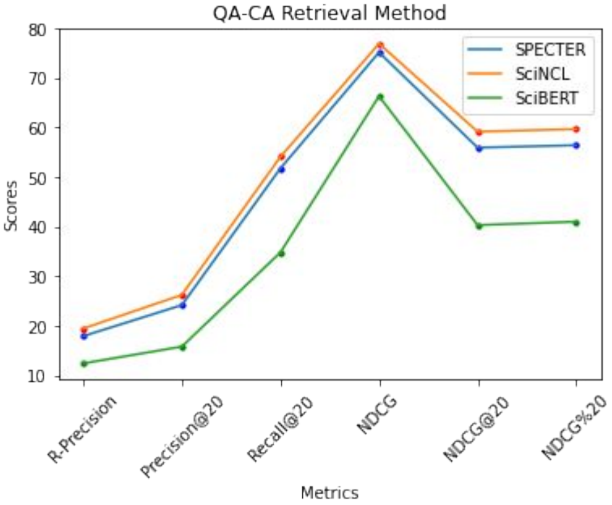
Approaches used

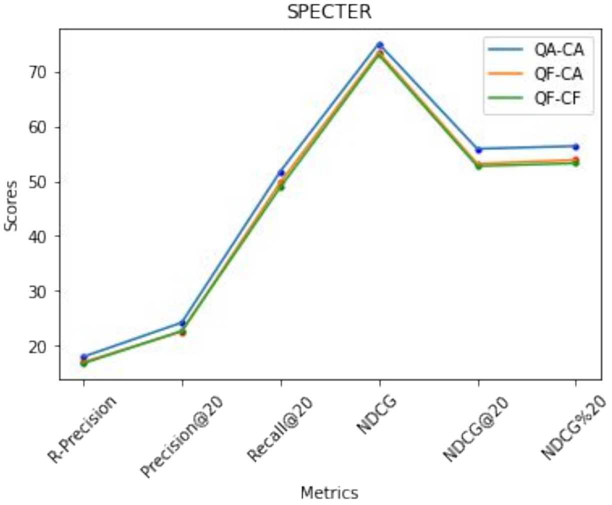
Three approaches used for construction of abstract:

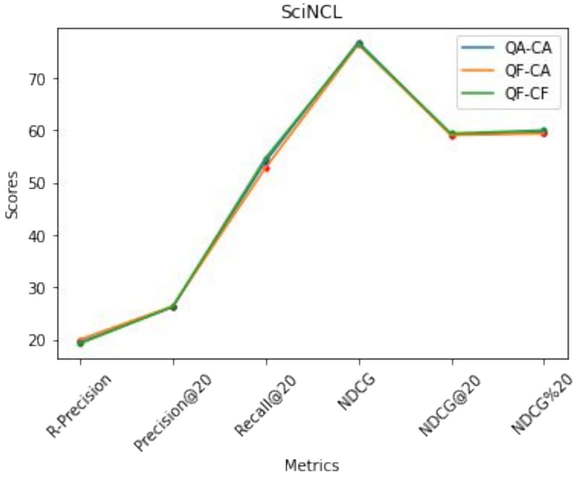
1. **Query Abstract - Candidate abstract (QA-CA):** For both query and candidate, entire abstract is considered.
2. **Query Facet - Candidate abstract (QF-CA):** For query only those sentences of abstract considered which have same label as the facet. Candidate same as in 1.
3. **Query facet - Candidate facet (QF-CF):** For both query and candidate, only those sentences of abstract considered which have same label as the facet.

Analysis of baseline models

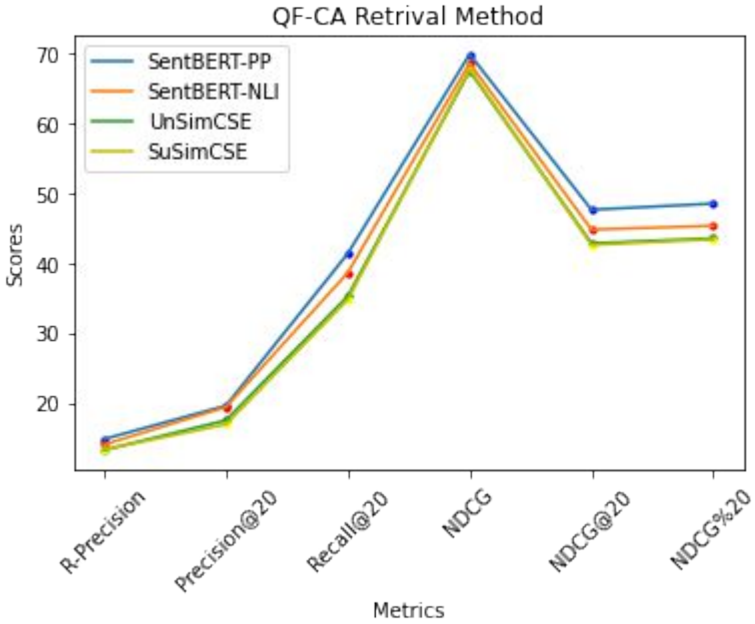
- **Abstract level models:**

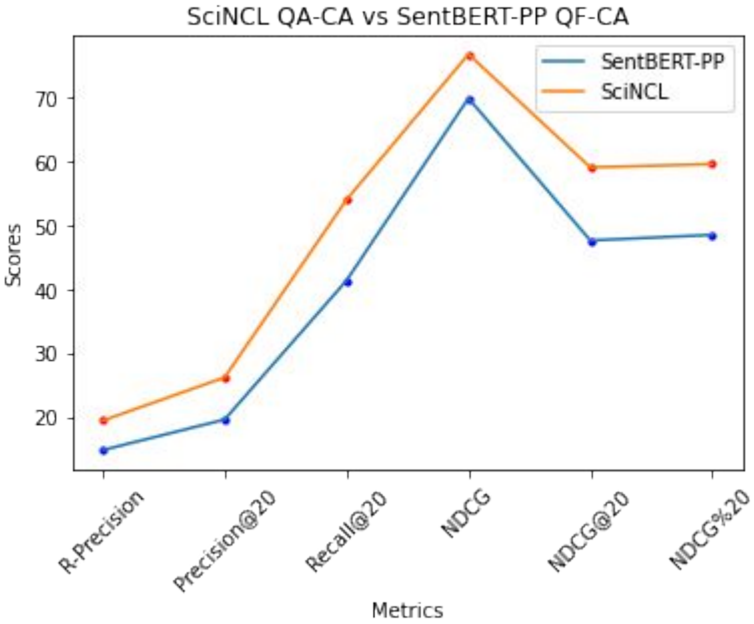

QA-CA Retrieval Method


SPECTER


SciNCL

- **Sentence level baselines:**





Fine-tuning approach

SPECTER and SciNCL:

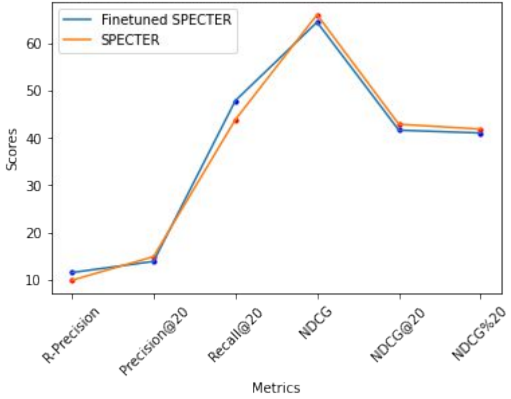
- These models do not consider the number of times paper P1 cites paper P2
- **Dataset used to fine-tune:** Highly influential citations(HIC) dataset from the SciRepEval Benchmark
- **HIC dataset:** Tuples of the form: ⟨q, ⟨c_1 , s_1 ⟩, ⟨c_2 , s_2 ⟩, . . . , ⟨c_n , s_n ⟩⟩, where q is the query paper, c_i is the candidate paper and s_i is the score
- s_i=1 if c_i is cited 4 or more times by q, else s_i=0
- To filter those query papers from HIC dataset which belong to the CS domain, we train a multi-label classifier.
- Fine-tuned SciBERT model having linear classification layer on top, using the Field of Study dataset from SciRepEval. Loss function = Cross-Entropy

- Filtered CS domain papers from HIC dataset and constructed triples of the form <query, positive paper, negative paper>
- Fine-tuned SPECTER and SciNCL using above dataset
- **Loss function:** Triplet Loss
- L(q, p, n) = max(0, D(q, p) — D(q, n) + margin)

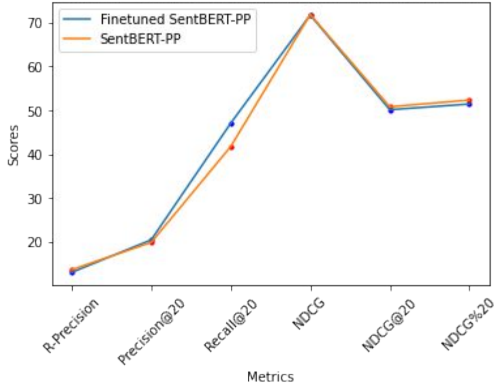
SentBERT-PP:

- The model captures sentence level similarity for general purpose but not specifically trained for sentences in CS domain
- **Dataset used:** PARADE dataset, consists of pairs of sentences from CS domain with binary scores
- **Loss function:** Contrastive loss function
- $L = (1-Y) * ||x_i - x_j||^2 + Y * max(0, m - ||x_i - x_j||^2)$

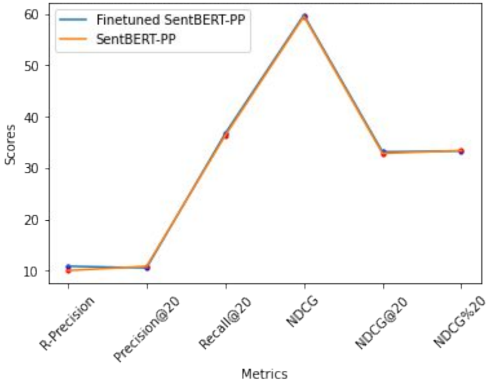
Analysis of fine-tuned models



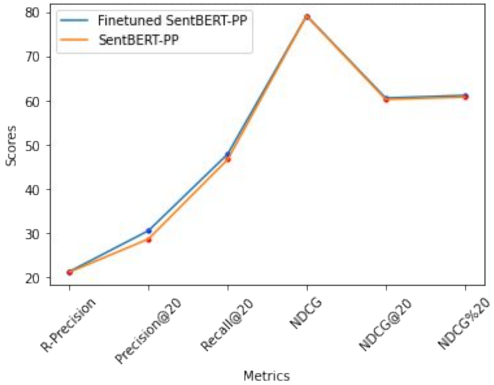
SPECTER vs Finetuned SPECTER QA-CA, Facet = Method



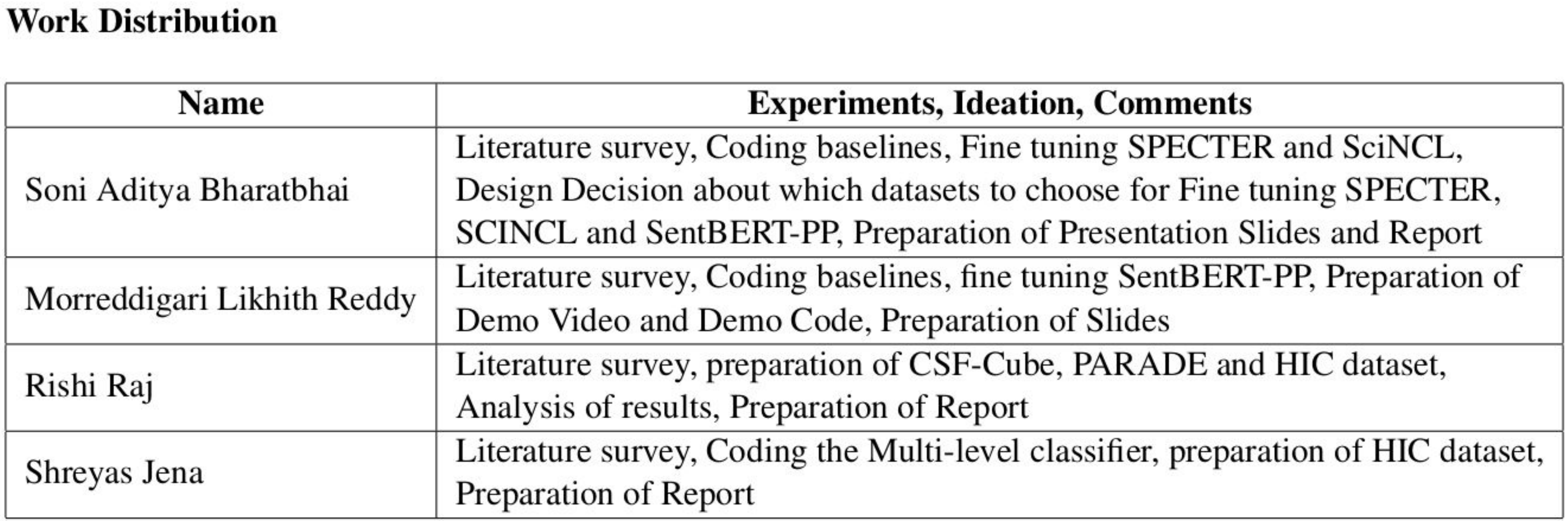
SentBERT-PP vs Finetuned SentBERT-PP QF-CA, Facet = Result



SentBERT-PP vs Finetuned SentBERT-PP QF-CA, Facet = Method



SentBERT-PP vs Finetuned SentBERT-PP QF-CA, Facet = Background



Work Distribution

Name	Experiments, Ideation, Comments
Soni Aditya Bharatbhai	Literature survey, Coding baselines, Fine tuning SPECTER and SciNCL, Design Decision about which datasets to choose for Fine tuning SPECTER, SCINCL and SentBERT-PP, Preparation of Presentation Slides and Report
Morreddigari Likhith Reddy	Literature survey, Coding baselines, fine tuning SentBERT-PP, Preparation of Demo Video and Demo Code, Preparation of Slides
Rishi Raj	Literature survey, preparation of CSF-Cube, PARADE and HIC dataset, Analysis of results, Preparation of Report
Shreyas Jena	Literature survey, Coding the Multi-level classifier, preparation of HIC dataset, Preparation of Report

THANK YOU!