INFORMATION RETRIEVAL (CS60092)
TERM PROJECT

Query-by-Example for Scientific Article Retrieval

Team Members

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

Introduction

Query-by-Example (QBE) is a mode of information retrieval that has gained a lot of traction in recent years. This has to do with the fact that user requirements are often very specific and require a lot of background context to be understood properly. This poses a major challenge to traditional information retrieval systems, because the relatively short size of queries for which such systems can return a sufficient number of relevant results is too small for such surrounding contexts to be accommodated. While QBE allows a user more flexibility in terms of asking queries, it adds additional processing overhead on the IR systems because such query examples typically are quite complex and do not follow specific patterns, making them harder to analyse.

This project deals with the specific task of "**Faceted Scientific Paper Retrieval by Example**" - given a query scientific paper **Q** and a facet **f** in {background,method,result}, where **f** signifies the specific aspect of interest, the aim is to retrieve and rank relevant papers from a set of candidate papers **C** which are similar to **Q** with respect to **f**. Each candidate paper in **C** has been assigned a graded relevance rating as 0,1,2 or 3.

For each paper P in the CSF-Cube dataset (query or candidate), the title and the abstract are provided. Additionally, the abstract sentences have been assigned a label from the set {background, method, result,other} using the model provided by **Cohan et al [7]**.

Objective

Our benchmark dataset is **CSFCube [1]** - an expert-annotated test dataset containing rated rankings of relevant research papers for 50 query document-facet pairs. The aim is to explore new techniques for improving upon the state-of-the-art research work on this task. A key tenet behind retrieving documents relevant to a query is to **incorporate more effective measures of calculating relevance between the query and the candidate documents.** A widely-used approach is to represent each query and retrieval unit in terms of vector embeddings and rank the candidates based on appropriate similarity measures.

Background

The most basic baseline models used for representing queries and candidates in terms of embeddings are the **term-level baselines** like **tf-idf** and **cbow**, which create vector representations considering words independent of the order in which they are present. Since these approaches consider the sentences as a bag of words, they are unsuitable for representing the inherent word order-based context which is a key part of the context in sentences, and are therefore not considered in our study.

We mostly consider state-of-the-art sentence-level and abstract-level baselines in our study. **Sentence-level baselines** like **SentBERT [2]** and **SimCSE [3]** produce embeddings for each sentence in the query/candidate, and the relevance rating between query and candidate is found out by the **maximum pairwise sentence cosine similarity** between the sentences in both of them.

Abstract-level baselines like **SPECTER [4]** and **SciNCL [5]**, on the other hand, produce embeddings for the whole abstract. Such models typically take a transformer model like SciBERT which is trained on scientific literature and further fine-tune SciBERT on citation network data.

This allows the models to capture inter-document similarity, leading to more relevant retrievals for each query. Additionally we have also used **SciBERT [6]** as an abstract level baseline. All the three abstract level baseline essentially provide a fixed length embedding representation for the document using the abstract.

Approach

We have added the list of papers reviewed in the **References** section.

With regards to the approach, we have tested out the performance of the following baseline models on the CSFCube dataset :

Abstract level baselines :

1. SciBERT **[6]**
2. SPECTER **[4]**
3. SciNCL **[5]**

Sentence level baselines :

1. SentBERT **[2]**
2. SimCSE **[3]**

The model performances were analysed by evaluating the following metrics :

- **R-Precision** : Ratio between relevant documents retrieved until the rank that equals the number of relevant documents present in the collection in total, to the total number of relevant documents in collection.
- **Precision@20** : Ratio of relevant documents retrieved within top 20 candidates to total documents retrieved (20).
- **Recall@20**: Ratio of relevant documents retrieved within top 20 candidates to total relevant documents in dataset.
- **NDCG** : Measure of graded relevance by giving a discount to the rating, more rewards if the more relevant documents are ranked higher in the overall ranking.
- **NDCG%20 :** Evaluates NDCG, taking the top 20% entries of the pool size into account.

Experiments

Abstract level baselines :

1. **SciBERT-uncased-scivocab**: The entire abstract is utilised for both query and candidate papers. The abstract is fed to the model and the CLS embedding is used as the document representation.
2. **SPECTER**: This model takes as input Title + [SEP] + Abstract, where [SEP] is the separator token of the tokenizer. For both query and candidate we feed input in above fashion and then take [CLS] embedding as the document representation.

3. **SciNCL**: Same input-output semantics as that of SPECTER.

For each of the above 3 abstract-level baselines we have used 3 approaches for using the abstract. Say we have the query paper **Q**, the facet **f** and the candidate papers' set **C**. For each sentence **s** in the abstract of a paper, let its label(background, method, result, other) be **s_label.**

1. **QA-CA (Query Abstract + Candidate abstract)**: Irrespective of facet f, we use all the abstract sentences as input irrespective of whether that sentence has **s_label** same as **f** or not.
2. **QF-CA (Query facet + Candidate abstract)**: In this case the candidate abstract is used in the same fashion as above. For the query paper, we consider only those sentences **s** from its abstract for which **s_label** is the same as **f**. All such sentences are concatenated to form the abstract for the particular facet **f**. Let this facet **f** specific abstract be called **abstract_f**.
3. **QF-CF (Query facet + Candidate facet)**: For both the query and the document pair, we use abstract_f which is constructed as stated in point 2.

For ranking the candidate documents for a query-facet pair, **L2 norm** is used as the measure of distance between query and candidate embeddings (higher distance means lower similarity). The candidates are **ranked in increasing order of their L2 distance** from the query embedding.

Sentence level baselines :

1. **Sentence-BERT:** This model takes a sentence and outputs its dense vector representation(embedding). There are two variations of this family of models - **SentBERT Paraphrased** and **SentBERT NLI** and both have the same input and output format.
2. **SimCSE:** There are two variations of this model namely Unsupervised SimCSE(**UnSimCSE**) and Supervised SimCSE(**SuSimCSE**). Both of these models take a sentence as input and output its dense vector representation(embedding).

For each of the above 3 sentence-level baseline, we have used 3 approaches for using the sentences. Say we have the query paper **Q**, the facet **f** and the candidates papers set **C**. For each sentence s in the abstract, let its label(background, method, result, other) be **s_label**.

1. **QA-CA (Query Abstract + Candidate abstract)**: Irrespective of facet we calculate cosine similarities of a sentence of Q and sentence of a candidate paper, and take maximum value of cosine similarity obtained. We then compare this cosine similarity for different candidate papers and rank the results.
2. **QF-CA (Query facet + Candidate abstract)**: Facets sentences from Query abstract are taken and all abstract sentences are taken from Candidate paper. Maximum cosine similarity is found and the documents are ranked in decreasing order of cosine similarity.
3. **QF-CF (Query facet + Candidate facet)**: For both Candidate and Query we have taken facet sentences, then calculated the maximum cosine similarity. Then the candidate documents are ranked according to these maximum cosine values.

For ranking, we saw that we have used cosine similarities and the candidate paper are **ranked in decreasing order of their maximum cosine similarities**.

Analysis

The CSFCube dataset was used as a benchmark to test various state-of-the-art baseline models based on multiple metrics (explained in **Approach** section) and a comparative summary of the overall performance of various models is as follows :

- **SPECTER vs SciBERT : (SPECTER > SciBERT)**
  SPECTER makes use of SciBERT under the hood, but it adds citation network information which provides a way to incorporate inter-document similarity. This is necessary for finding relevance between query and candidate documents, hence SPECTER overperforms SciBERT on all metrics under the QA-CA, QF-CA and QF-CF

categories, in which the whole abstract is encoded into a representation. SciNCL has performance slightly better than SPECTER in general.

- **SPECTER vs SentBERT/SimCSE : (SPECTER > SentBERT/SimCSE for QA-CA)**
  SentBERT produces sentence-level embeddings, but each such embedding doesn't take the context of the surrounding sentences into account, unlike SPECTER, which uses whole abstract-level embeddings. This gives SPECTER an advantage in terms of performance especially in whole abstract-type categories like QA-CA, as compared to faceted categories like QF-CF or QF-CA.

- Abstract-level baselines show better performance on whole abstract-level categories like QA-CA, while sentence-level baselines perform better for faceted query categories like QF-CA or QF-CF.

- A large drop in accuracy is observed across faceted and non-faceted query document pairs for SciBERT as its training on the whole abstract will naturally help it perform better on QA-CA. Such a drastic drop isn't observed for SciNCL and SPECTER on going from QA-CA to QF-CA/QF-CF as their training also includes citation network information. Hence, SciNCL and SPECTER perform consistently well on various generalised faceted and non faceted queries. **This can be used as the justification to use one of these models for future research in this domain.**

- The models are usually observed to perform well for the "background" facet, but they perform poorly on the "method"and "result" facets. This can be attributed to the fact that the "background" facet usually contains generic sentences describing a situation, which can be easily matched using known methods. On the other hand, "method" and "result" typically involve complex jargon and hence the similarity between two sentences may not be found, even if both sentences refer to the same thing.

Further Work

- We aim to use the SPECTER and SciNCL architecture and add a few extra feedforward layers on top. These layer weights will then be finetuned using **co-citation data from the SciDocs [4]** dataset. The SciDocs dataset currently contains 30k total papers with annotated co-citations, and since co-citations can be considered as a stronger measure of similarity between two papers than direct citations, we aim to leverage the co-citations dataset for finetuning the SPECTER and SciNCL models.
- **PARADE dataset** consists of data for paraphrase identification pairs where the sentences are from the computer science domain. Since the sentence level baseline modes are general purpose, they have not been trained specifically to capture sentence similarity in the computer science domain. We aim to use this dataset to obtain improvement on sentence level baselines.
- **Sequential Sentence Classification (SSC Model)**: We have set up the model of Cohan et.al [7] which can classify each sentence of abstract into one of these classes - { background, objective, method, result, other }. We have set it up so that we can possibly use it for sentence level baseline models given a dataset which only has abstract.

Work Distribution

1) **Morredigari Likhith Reddy:** Literature survey, Sentence level baselines coding, CSF Cube Dataset preparation, set up SSC model
2) **Soni Aditya Bharatbhai:** Literature survey, Abstract level baselines coding, CSF Cube Dataset preparation, set-up SSC model
3) **Rishi Raj:** Literature survey, Prepared results spreadsheet and computed evaluation metrics for abstract level baselines, Analysis of results.
4) **Shreyas Jena:** Literature survey, Prepared results spreadsheet and computed evaluation metrics for abstract level baselines, Analysis of results

Links

1) Code : https://shorturl.at/fuLR8
2) Baselines results sheet : https://shorturl.at/cmpBE

References

[1] Sheshera Mysore, Tim O'Gorman, Andrew McCallum and Hamed Zamani (2021). "CSFCube – A Test Collection of Computer Science Research Articles for Faceted Query by Example".
[2] Nils Reimers and Iryna Gurevych (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks".
[3] Tianyu Gao, Xingcheng Yao, Danqi Chen (2022). "SimCSE: Simple Contrastive Learning of Sentence Embeddings".
[4] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, Daniel S. Weld (2020). "SPECTER: Document-level Representation Learning using Citation-informed Transformers".
[5] Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, Georg Rehm (2022). "Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings".
[6] Iz Beltagy, Kyle Lo, Arman Cohan (2019). "SciBERT: A Pretrained Language Model for Scientific Text".
[7] Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, Daniel S. Weld (2019). "Pretrained Language Models for Sequential Sentence Classification".