Finding Related ArXiv Papers with BERT

CS224U Project - Experiment Protocol

Junshen Kevin Chen
Stanford University
jkc1@stanford.edu

1 Introduction

In this project, I propose to design a system that evaluates the level of relation between two academic papers (classification), and suggest possible related work given a text abstract as query (retrieval) using semantic information, in hopes that it would augment the current keyword-matching-based academic search engines such as Google Scholar.

I propose to use a pre-trained BERT [todo](and possibly other pre-trained deep transfomer models if time permits) for this task, fine-tuning and building a model that encodes the text of each paper in the corpus into a vector representation. Then, given an unseen query, use the same model to encode the query text, and retrieve document(s) related to the query based on vector distances.

2 Hypothesis

I hypothesize that a pre-trained BERT model can be fine-tuned to encode "paper relation" using a corpus of ArXiv paper abstracts, and this model is able to retrieve related articles in the corpus to some accuracy. I further hypothesize the resulting system would have a run time complexity in the seconds, such that it has practical uses in augmenting existing keyword-matching-based academic search engines.

3 Data

3.1 Paper abstracts and citation graph

I use Semantic Scholar's Open Research Corpus (s2-corpus) [todo] as my main source of data. This corpus encompasses numerous popular sources of academic research, including ArXiv, IEEE, NeurIPS, etc. It also contains paper from various regions and in various languages. There are a totoal of [todo] items in this corpus.

Each entry in this data set is an academic paper uniquely identified by an ID, and includes metadata information such as the title, fields of study, authors, year of publication, source conference, etc.

The s2-corpus includes the citation graph in the form of the ID's of in-citations and out-citations for each paper. From my investigation, this citation graph is significantly more complete in comparison to the ArXiv citation graph produced by [todo].

I use the paper abstract text from this corpus, which is the main input of the BERT-based model.

3.2 Paper full text

Because Semantic Scholar is a search engine that only points to contents in other hosts and conferences, the s2-corpus provides no full-text of papers. To augment the models with full texts, I use the ArXiv dataset by [todo].

3.3 Preprocessing

Before using the aforementioned two corpora, I first run them through a data pipeline to perform the following preprocessing steps:

1. Filtering out articles not in English (based on abstract and title), since BERT is pre-trained in English.
2. Filtering out articles not in ArXiv, since we do not have the full text otherwise.
3. Optionally keeping only articles in Computer Science. I will train a smaller model with CS articles only for a proof-of-concept, then use the full ArXiv corpus to train a bigger model.
4. Removing in-citation or out-citation pointers of articles that are not kept after the previous steps
5. Tokenizing and encoding each abstract text with pre-trained BERT tokenizer, prepending

the `[CLS]` token, then trimming to the max BERT length of 512

3.4 Train-dev-test split

After the aforementioned filtering steps, the resultant CS-only articles corpus has [todo] entries. I randomly split this corpus to train, dev, test sets in the proportion of $90 : 5 : 5$.

3.5 Sampling

For the specific model I propose, I need training and testing samples in the form of triplets:

$(anchor, positive\_sample, negative\_sample)$

Where a positive sample is an article that either cites or is cited by the anchor, and a negative sample is an article neither cites nor cited by the anchor.

I employ algorithm 1 to sample the triplets:

Algorithm 1: Sample Triplets

Input : S, set of articles
Output : D, set of article triplets
$D \leftarrow \{\}$
for $a \in S$ **do**
  for $p \in Citations(a)$ **do**
    repeat
      $n \leftarrow RandomChoice(S)$
    until $n \notin Citations(a)$;
    $D \leftarrow D \cup (a, p, n)$
  end
end

The algorithm is performed on each of train, dev, test set.

4 Metrics

5 Models and General Reasoning

As the problem is defined such that the distances between BERT-encoded vector representation of related articles are small, and that of unrelated articles are large, I propose to use

6 Summary of Progress