Project proposal

Automated Quality Assurance for Literature Citations

Jeffrey Dick

2024-11-26

1. Statement of the problem; what makes it an interesting problem

Literature citations are an important part of scientific articles. Proper citations relate the work to past research and justify the methods and claims made in an article. However, all too often, citations suffer from poor quotation accuracy – they do not reflect the content of the cited source or do not support the authors' claims. Studies of quotation accuracy report that 10-20% or more of citations in various disciplines are inaccurate. A tool that automatically detects inaccurate citations would boost confidence in the scientific literature.

The problem affects a wide cross section of stakeholders, from authors to editors and their institutions. The conventional advice for improving quotation accuracy is “read before you cite”. While human judgment remains central to production of quality scientific content, an automated system to check quotation accuracy would be a valuable addition. Recent advances in natural language processing (NLP) and availability of datasets to train the models make this a feasible goal.

2. Data needed to solve the problem; data sources

The project will use data from two studies of quotation accuracy, specifically SciFact and CitationIntegrity, published in 2020 and 2024. In terms of number of citation instances, these are not large datasets (1409 and 3063, respectively), but data preprocessing requires extraction of sentences relevant to the claim (rationale sentences) from source documents, greatly multiplying the number of sentences that need to be considered.

Quotation accuracy in scientific articles (where a source is cited) is related to the problem of claim verification (where a source is not cited). Therefore, claim verification datasets are useful for pre-training the models. There are various publicly available claim verification corpora, and I have provisionally selected Fever, which consists of 185,445 claims synthesized from Wikipedia.

The data will be downloaded from the following sources:

SciFact: https://github.com/allenai/scifact linking to Amazon S3
CitationIntegrity: https://github.com/ScienceNLP-Lab/Citation-Integrity/ linking to Google Drive
Fever: https://fever.ai/dataset/fever.html

3. Approach to solving the problem

This is a supervised classification problem. The system will predict a label for each claim (Support, Refute, or Not Enough Information). The predictors are the text of a claim and the rationale sentences from a cited source. The system will use deep learning, e.g. Bidirectional Encoder Representations from Transformers (BERT).

Working plan:

- Phase 1: Use the available preprocessed data in the SciFact and CitationIntegrity datasets, which provide rationale sentences extracted from articles or abstracts. The first step will replicate previous work done by NLP researchers. Then, by combining the datasets, this project will build new capabilities.

- Phase 2: Use entire PDFs rather than provided sentences in order to optimize a preprocessing pipeline. This will involve more complex NLP engineering and possibly large language models (LLMs). An additional goal is to develop claim retrieval or generation capability. However, claim retrieval is a relatively young field, so this may not be feasible within the time frame of this project.
- Phase 3: Deployment to the cloud.
- Phase 4: Production stage and analysis of quotation accuracy in large datasets, such as bioRxiv.

4. Final deliverable

A web app will be built for users to submit texts for citation checking. This will be a convenient tool for authors and editors, as they can simply upload a claim and the cited PDF to check quotation accuracy.

5. Minimal computational resources

A sufficiently powerful GPU is needed to train the models. According to the makers of the SciFact dataset (https://arxiv.org/abs/2004.14974), training the RationaleSelection and LabelPrediction modules takes about 700 and 640 minutes, respectively, using a single Nvidia P100 GPU on Google Colab Pro.