Course Project: Automated Fact Checking

Qiang Zhang, Emine Yilmaz

March 26, 2019

1 Task Definition

An increasing amount of online misinformation has motivated research in automated fact checking. Your task in this project is to develop information retrieval and data mining methods to assess the veracity of a claim. Specifically, the automated fact checking project consists of three steps: relevant document retrieval, evidence sentence selection, and claim veracity prediction.

2 Dataset

We will be using the publicly available Fact Extraction and Verification (FEVER) dataset [1]. It consists of 185,445 claims manually verified against Wikipedia pages and classified as *Supported*, *Refuted* and *NotEnoughInfo*. For the first two classes, there is a combination of sentences forming the necessary evidence supporting or refuting the claim. This dataset consists of a collection of documents (wiki-pages), a labeled training subset (train.jsonl), a labeled development subset (dev.jsonl) and a reserved testing subset (test.jsonl). For a claim in the train.jsonl file, the value of the "evidence" field is a list of relevant sentences in the format of [_, _, wiki document ID, sentence ID]. More details about this dataset can be found in [1] and this website [2]. A demo for reading this dataset is on the website [3].

2.1 Claims for IR Tasks

To reduce computational time, for the subtask 2 and 3, you are supposed to return retrieval results only for the first 10 verifiable claims in the train.jsonl file. The list of the claim IDs are [75397, 150448, 214861, 156709, 129629, 33078, 6744, 226034, 40190, 76253]. For the subtask 4 and 6, to train the models, you are free to use any amount of training data from the training subset but keep in mind the number of used training examples impact your model's performance.

c value:
For the entire corpus the average value of c is: 0.00791
For the top 50 terms in the corpus, the average value of c is 0.08991

For the top 10 terms in the corpus, the average value of c is 0.13374
The corpus also contains non-english words

3 Involved Subtasks

The course project involves several subtasks that are required to be solved. This is a research oriented project so you are expected to be creative and coming up with your own solutions is strongly encouraged for any part of the project.

1. Text Statistics. Count frequencies of every term in the collection of documents (5 marks), plot the curve of term frequencies and verify Zip's Law (5) marks. Report the values of the parameters for Zipf's law for this corpus. (10 marks in total)

2. Vector Space Document retrieval. Extract TF-IDF representations of the 10 claims and all the documents respectively based on the document collection. The goal of this representation to later compute the cosine similarity between the document and the claims. Hence, for computational efficiency, you are allowed to represent the documents *only* based on the words that would have an effect on the cosine similarity computation (5 marks). Given a claim, compute its cosine similarity with each document and return the document ID (the

[1] http://fever.ai/resources.html
[2] http://fever.ai/2018/task.html
[3] https://github.com/QiangAIResearcher/Fact-Extraction-and-Verification

"id" field in the wiki-page) of the five most similar documents for that claim (5 marks). (10 marks in total)

3. Probabilistic Document Retrieval. Establish a query-likelihood unigram language model based on the document collection, and return the five most similar documents for each one of the 10 claims (5 marks). Implement and apply Laplace Smoothing, Jelinek-Mercer Smoothing and Dirichlet Smoothing to the query-likelihood language model, return the five most similar documents for the 10 claims (5 marks). (10 marks in total)

4. Sentence Relevance. For a claim in the training subset and the retrieved five documents for this claim (either based on cosine similarity or the query likelihood model), represent the claim and sentences in these documents based on a word embedding method, (such as Word2Vec, GloVe, FastText or ELMo) (5 marks). With these embeddings as input, implement a logistic regression model trained on the training subset (5 marks). Use the first 10 verifiable claims in the development dataset, i.e., claim ID [137334, 111897, 89891, 181634, 219028, 108281, 204361, 54168, 105095, 18708], to evaluate the performance of the logistic regression model. Note that you need to first retrieve five relevant documents for each of the 10 claims based on Q2 or Q3. Report the performance of your system in this dataset using an evaluation metric you think would fit to this task. Analyze the effect of the learning rate on the model training loss (5 marks). Instead of using Python sklearn or other packages, the implementations of the logistic regression algorithm should be your own. (15 marks in total)

Google pretrained embeddings used

Q4 Available at: https://www.kaggle.com/rajaupadhyay/test-src/edit

5. Relevance Evaluation. Implement methods to compute recall, precision and F1 metrics (5 marks). Analyze the sentence retrieval performance of your model using the labelled data in the development subset (5 marks). (10 marks in total)

6. Truthfulness of Claims. Filter out the 'NOT ENOUGH INFO' claims and only keep the 'SUPPORTS' or 'REFUTES' claims in the train.jsonl and dev.jsonl datasets. Using the relevant sentences specified in the 'evidence' field as your training data and using their corresponding truthfulness labels in the train.jsonl file, build a neural network based model to assess the truthfulness of a claim in the training subset. No need to retrieve documents and select sentences for this part, just use the sentences specified in the 'evidence' field in the train.jsonl and dev.jsonl.You may use existing packages like Tensorflow or PyTorch in this subtask. You are expected to propose your own network architecture for the neural network. Report the performance of your system in the labelled development subset using evaluation metrics you have implemented. Furthermore, describe the motivation behind your proposed neural architecture. The marks you get will be based on the quality and novelty of your proposed neural network architecture, as well as its performance. (15 marks in total)

81.53% dev set accuracy

Bi-LSTM available at: https://www.kaggle.com/rajaupadhyay/q6-nli-implementation/edit

CNN model Available at: https://www.kaggle.com/rajaupadhyay/irdm-dl/edit

7. Literature Review. Do a literature review regarding fact checking and misinformation detection, identify pros and cons of existing models/methods and provide critical analysis. Explain what you think the drawback of each of these models are (if any). (10 marks in total)

8. Propose ways to improve the machine learning models you have implemented. You can either propose new machine learning models, new ways of sampling/using the training data, or propose new neural architectures. You are allowed to use existing libraries/packages for this part. Explain how your proposed method(s) differ from existing work in the literature. The marks you get will be based on the quality and novelty of your proposed methods, as well as the performance of your final model. (20 marks in total)

4 Summary of the Marks

Subtask	1	2	3	4	5	6	7	8
Marks	10	10	10	15	10	15	10	20

5 What to submit

You are expected to submit all the code you have written, two csv files containing the retrieval results of subtask 2 and 3 respectively, a jsonl file containing your final model's prediction (the results of the model you have obtained in the last step) on the test subset, and a written report up

to 8 pages (including references). The example format of the csv file can be found in this website [4]. The format of the jsonl file should follow the Answer Submission Instructions at the bottom of the Sumission Instructions page on FEVER website [5]. Your report should also include your findings on each of the steps.

Unless otherwise stated above, all the code should be your own and you are not allowed to reuse any code that is available online. You are allowed to use both Python and Java as the programming language.

Your report should describe the work you have done for each of the aforementioned steps. You are required to use the SIGIR 2019 style template for your report. You can either use LaTeX or Word available from the ACM Website [6] (use the sigconf proceedings template). Do not change the template (e.g. reducing or increasing the font size, margins, etc.).

For model performance comparison, students are encouraged to submit their test subset predictions to the FEVER Challenge CodaLab and report their results as well [7]. Marks for the last subtask will be based on your model's performance on the test subset, as well as the quality and the novelty of your proposed method.

6 Deadline

The deadline for submitting your project is midnight on April 10th.

References

[1] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[4] https://goo.gl/uXpWGU
[5] http://fever.ai/2018/task.html#Answer_Submission_Instructions_101/
[6] https://www.acm.org/publications/proceedings-template
[7] https://competitions.codalab.org/competitions/18814/