|corpus| = 5416536

|vocab| after stopword removal = 4474099

|vocab| = 4474252

|corpus| = 5,396,041

|vocab| = 3,922,275

non-zero count for sparse matrix: 211,785,807

Course Project: Automated Fact Checking

Qiang Zhang, Emine Yilmaz

February 8, 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 is divided into a labeled training subset, a labeled development subset and a reserved testing subset. Details about this dataset can be found in [1]. A demo for reading this dataset is on the website [2].

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 training document collection (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 vector representation of claims and documents respectively (5 marks). Given a claim in the training subset, compute its cosine similarity with each document and return the five most similar documents for that claim (5 marks). (10 marks in total)

3. Probabilistic Document Retrieval. Establish a unigram language model based on the training subset (5 marks). Respectively apply Laplace Smoothing, Jelinek-Mercer Smoothing and Dirichlet Smoothing to construct a query likelihood language model and return the five most similar documents for each claim in the training subset (5 marks). (10 marks in total)

4. Sentence Relevance. For each 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 five documents based on a word embedding method, (such as Word2Vec, GloVe, FastText or ELMo) (5 marks). Implement a logistic regression model to classify sentences in these five documents based on their relevance. Use your models to identify five relevant documents to the claims in the labelled development data and select the five most relevant sentences for a given claim within these documents (5 marks). Report the performance of your system in this dataset using an evaluation metric you think would fit to this task. Analyze the effect of learning rate on the model loss (5 marks). Instead of using

[1] http://fever.ai/resources.html

[2] https://github.com/QiangAIResearcher/Fact-Extraction-and-Verification

Python sklearn or other packages, the implementations of the logistic regression algorithm should be your own. (15 marks in total)

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. Truthfullness of Claims. Using the selected five sentences in Subtask 4 as your training data and using their corresponding truthfullness labels, build a neural network based model to assess the truthfulness of each claim in the training subset. You may use existing packages like Tensorflow or PyTorch in this subtask. You are expected to propose your own architecture for the neural network. Report the performance of your system in the labelled development dataset using evaluation metrics you have implemented. 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)

7. Literature Review. Do a literature review regarding fact checking and misinformation detection, briefly summarize and compare existing models. 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, 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 format of the jsonl file should follow the Answer Submission Instructions at the bottom of the Sumission Instructions page on FEVER website [3]. 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 [4] (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 [5]. 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 1st.

[3] http://fever.ai/2018/task.html#Answer_Submission_Instructions_101/

[4] https://www.acm.org/publications/proceedings-template

[5] https://competitions.codalab.org/competitions/18814/

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.