Fact Extraction and Automated Claim Verification

|  |
| --- |
| **Elizabeth Soper, Rohit Lalchand Vishwakarma, Swapnil Kishore** |

Abstract

The objective of this task is to build an end-to-end system that can verify or refute a claim given to it as input. For the verification of the claim, the system uses evidence in the form of Wikipedia pages. The claim can be verified using sentences selected from these Wikipedia pages. The output of the system is one of the three labels for each claim. These target labels are: {SUPPORTES, REFUTES, NOTENOUGHINFO} representing that the claim has been verified, refuted and not enough information was available to verify the claim respectively.

1. Credits

This task is known as the FEVER task and is conducted annually as a competition which involves submissions from multiple teams. Several groups have performed ground-breaking research in the field of claim verification. Some of the inspirations for our work are mentioned here. The FEVER dataset and shared task were created by Thorne et al 2018, who also outline a baseline system. Soleimani et al 2019 apply BERT to the FEVER task, and achieve second place on the 2018 FEVER shared task. They use two fine-tuned BERT models: one for sentence selection and the other for claim classification. Yixin Nie et al 2018 have made a homogenous Neural Semantic Matching Network Model for the purpose of FEVER task. This homogeneous model had been used for all the three subtasks i.e. Document Retrieval from Wikipedia, Sentence Selection and Claim Verification.

1. Introduction

In today’s world, the circulation of news is instant and widespread. All it takes for information about an event at one end of the globe to reach the other end is a Twitter trend or a YouTube video. Due to this system information can pass through several intermediary sources before reaching an individual and naturally gets altered in the process. Facts are diluted or corrupted and rumors originate in this way. It can be difficult to identify the truth among the noise of false claims. In this project, we work on an end-to-end system that verifies claims by extracting evidence related to them from Wikipedia pages. Based on the collected evidence, the task then is to judge whether the claim can be verified or not. The FEVER (Fact Extraction and Verification) dataset was created to train and evaluate systems on this task. In the following proposal we first outline the task, including the FEVER dataset and evaluation method, give an overview of the current state-of-the-art, and finally propose a baseline and a novel system to tackle the important task of verifying claims.

1. Dataset Description

Manuscripts FEVER (Fact Extraction and VERification) consists of 185,445 claims manually classified as ‘SUPPORTED’, ‘REFUTED’, or ‘NOTENOUGHINFO’. The purpose of this dataset is to evaluate an end-to-end system that performs the task of evidence extraction and claim verification. The dataset is divided into training data and testing data. The training dataset consists of the following fields:

1. Id: The id of the claim
2. label: One of {SUPPORTED, REFUTED, NOTENOUGHINFO}
3. claim: The text of the claim
4. evidence: A list of evidence sets extracted as being relevant to the claim.

The test dataset comprises of 20000 test claims. Along with this the FEVER task also provides a corpus of over 5 million pre-processed Wikipedia pages which is called the Pre-processed Wikipedia pages (June 2017 dump).

1. Task Description

The task is broadly divided into 3 different subtasks. These are evidence extraction, sentence selection and claim labelling. The overall pipeline for the end-to-end system that comprises of these tasks is given below:

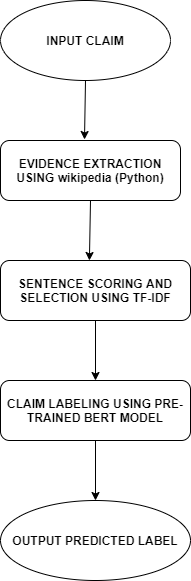


Figure 1. End-to-end system pipeline.

* 1. Evidence Extraction

The goal of this subtask is to get the Wikipedia pages most relevant to the claim. This is the first subtask that is performed. The strategies we tried for this step are:

1. **Baseline Strategy:** In the baseline model that we built the strategy for this subtask was using the ‘wikipedia’ library available in Python which wraps around the MediaWiki Api. As input to the command for retrieving pages, we are passing the noun phrases of each claim. The noun phrases have been extracted from each claim by using the noun\_phrases component of TextBlob library available for python. As part of the baseline system, we are retrieving only the top three wikipedia pages which are relevant to the claim. Later on in this report we present comparisons of results obtained when passing the full text of the claim as input against only passing the phrases obtained using nltk.
2. **Sequence Matcher:** Another strategy that we were able to implement for this subtask was to use Python’s inbuilt sequence matcher to find the most relevant Wikipedia pages for the claim. The title of each Wikipedia page was compared to the text of the claim and a score for each Wikipedia page is generated. Then the three highest scoring pages are selected. Since for this purpose the sequence matcher needs to generate a score for each of the over 5 million documents available, this method is extremely time-consuming and requires extensive resources.
   1. Sentence Selection

The second subtask involves selecting the sentences that are most suitable as evidence from the Wikipedia pages that were the output of the first subtask. The important thing to note here is that the text for each claim can also be a concatenation of sentences from the same source or multiple sources. The strategies implemented for this subtask were:

1. **Raw TF-IDF Score:** Here we have searched for the sources obtained in the wikipedia corpus provided by the FEVER task. The wikipedia corpus is a dump of pre-processed wikipedia pages from June 2017. Here “pre-processed” means that a set of sentences have automatically been selected from each of the pages. The chosen evidence has to be a subset of these sentences for each source. We have ranked the sentences present for each source according to TF-IDF score. Then the top scoring sentence from each source is obtained and the evidence for the respective claim is presented as a concatenation of these sentences. For this purpose a simple TF-IDF script in python was used which calculates the scores for each sentence and returns the top scoring sentence.
2. **Combined Similarity Score:** The follow-up strategy we implemented was to calculate an overall score for each sentence by adding the raw tf-idf score for each sentence to the cosine-similarity score for that sentence and the claim. To generate the cosine-similarity score the tfidf vectorizer from sklearn and the sparse function from scipy in python are used. After adding the two scores, the overall score for each sentence is generated which is then used to rank the sentences.
   1. Claim Labelling: This is the final subtask which involves classifying the claim into one of the three target labels on the basis of the evidence collected. The evidence consists of the sentences extracted from the Wikipedia pages. The strategy here is to use the pre-trained BERT base model to create representations for each sentence. BERT is a language representation model based on a transformer architecture and has achieved state-of-the-art performance on numerous NLP tasks. After creating these BERT representations, we concatenate the representations for claim and evidence sentences and use this to train a logistic regression classifier.
3. Results

In this section, results of the evaluation of the system are presented. The metrics that are being used to evaluate the system are label accuracy, precision, recall and f1 score.

* 1. LABEL ACCURACY:

|  |  |  |
| --- | --- | --- |
|  | TEXT | PHRASES |
| SIMILARITY SCORE | **45.46** | **41.38** |
| COMBINED SCORE | **44.56** | **42.78** |

As we can see from the results given above, the highest accuracy we achieve is when evaluating on the similarity score only while using the full text of the claim.

* 1. PRECISION:

|  |  |  |
| --- | --- | --- |
|  | **TEXT** | **PHRASES** |
| **SIMILARITY SCORE** | **53.3** | **19.6** |
| **COMBINED SCORE** | **55.5** | **20.2** |

* 1. RECALL:

|  |  |  |
| --- | --- | --- |
|  | **TEXT** | **PHRASES** |
| **SIMILARITY SCORE** | **13.2** | **11.2** |
| **COMBINED SCORE** | **13.5** | **13.5** |

* 1. F1:

|  |  |  |
| --- | --- | --- |
|  | **TEXT** | **PHRASES** |
| **SIMILARITY SCORE** | **21.1** | **14.254** |
| **COMBINED SCORE** | **21.7** | **16.2** |

1. Conclusion

In conclusion, this work done on this project managed to achieve a peak label accuracy of 45.46 and a peak evidence f1 score of 21.7. A lot of work has been done in this area before and the best results were achieved by the UNC-NLP group. This task is very important in today’s world as fake news is widespread. Thus, an end-to-end system which, given a claim as input, is able to accurately judge whether the claim is verifiable or not is practically invaluable.

Acknowledgements

We would like to thank our course instructor Professor Rohini Srihari for assigning us this task so we could work on it and learn new and different things along the way. We would also like to thank the teaching assistant for this course Archita Pathak for providing detailed instructions and plenty of resources to take help from so that we could work on this project as smoothly as possible.

Reference

Thorne et al 2018 (baseline FEVER system):[**https://arxiv.org/abs/1803.05355**/](https://arxiv.org/abs/1803.05355/)

Devlin et al 2018 (BERT architecture): [**https://arxiv.org/pdf/1810.04805.pdf**](https://arxiv.org/pdf/1810.04805.pdf)

Soleimani et al 2019 (BERT for FEVER): [**https://arxiv.org/abs/1910.02655**](https://arxiv.org/abs/1910.02655)

Hanselowski et al 2018 (MediaWiki for FEVER): [**https://www.aclweb.org/anthology/W18-5516.pdf**](https://www.aclweb.org/anthology/W18-5516.pdf)

Yixin Nie et al 2018 (NSMN for FEVER): [**https://arxiv.org/abs/1811.07039**](https://arxiv.org/abs/1811.07039)

Fever Shared Task 2018: <https://fever.ai/2018/task.html>