Assignment 2 report: Automated Fact Checking

Campardo Giorgia, Chinellato Diego, Fanti Pietro, Longhi Carlo
Alma Mater Studiorum - Master's Degree in AI, Natural Language Processing course

Abstract

In this report we show an application of neural architectures to the task of fact checking. Our model, trained on the FEVER dataset (James Thorne (2018)), determines whether a given statement conveys a trustworthy information or not. We show that Recurrent Neural Networks can be employed to achieve a performance of 0.79 in terms of F1-score on this dataset.

1. Introduction

Fact checking is the task of determining whether claims made in written or spoken language are true, using some evidence. In this assignment we tackle this job using neural architectures.

Given a corpus of claims, generated by altering sentences extracted from Wikipedia, and their associated evidences, the task consists of predicting if the latter support or refutes the former. Claims and Evidences in the dataset are pre-processed by removing all the special characters, including those representing the pronunciation of entities, that in the corpus are present inside square brackets. The evidences are also stripped of the tag-like repetitions present after the last phrase.

2. Architecture

Our architecture is composed of four different stages: word embedding, sentence encoding, merging and classifier. Claims and evidences are given in input to the first section that replaces every word with its corresponding GloVe embedding. For out-of-vocabulary words, we have tried both random embedding and mean of the context embedding obtaining similar results, but with the first method way faster. Then, they are encoded separately and merged; finally, the resulting encoding is passed on to the classifier. The sentence encoding is obtained using one of the following approaches:

- RNN_last: last state of a Bidirectional RNN, GRU or LSTM network of one or more layers
- RNN_avg: average of the states of a Bidirectional RNN, GRU or LSTM network of one or more layers
- Bag of Vectors: the mean value of the token embeddings
- MLP: the output of a simple MLP layer

The sentence encodings of the claim and that of the evidence can be merged using different strategies: concatenation, sum or mean. The last stage of our architecture is the classifier, composed of a stack of fully connected layers.

To speed-up the training process, we implemented also early stopping.

3. Experiments

To select the best architecture, we first tested different sentence encoders using the same configuration:

- *learning rate*: 0.0001;

- *max number of epochs*: 50;
- *batch size*: 512;
- *optimizer*: RMSprop;
- for RNN encoders: *hidden size*: 64; *recurrent size*: 2.

We selected the best one by comparing their F1 scores. Then we carried out experiments to determine the best merging strategy for the encoding of the claim and evidence.

To further estimate the performance of our models, we implemented the claim verification evaluation and computed the same metrics.

Encoder	Valid Accuracy	Valid F1 Score	Valid Claim F1
GRU_LAST	0.757	0.793	0.790
GRU_AVG	0.754	0.788	0.786
LSTM_LAST	0.749	0.787	0.785
ELMAN_LAST	0.739	0.780	0.777
LSTM_AVG	0.746	0.778	0.774
ELMAN_AVG	0.736	0.775	0.772
MLP	0.689	0.736	0.735
BOV	0.617	0.681	0.677

Table 1: Results of the experiments to choose the encoder. The data refers to experiments carried out using the concatenation merging strategy. Results are ordered from the best to the worst.

Merger	Valid Accuracy	Valid F1 Score	Valid Claim F1
concatenation	0.757	0.793	0.790
sum	0.745	0.782	0.779
mean	0.742	0.782	0.778

Table 2: Results of the experiments to choose the merger. The data refers to experiments carried out using the GRU_LAST encoder.

At the end, we have fixed also the best merger and we have used a random search to tune the other hyperparameters, including also the cosine similarity. However, no runs in the random search have performed better than the best run of the merger tuning.

4. Results

Our experiments show that the best architecture for the sentence encoder is the Gated Recurrent Unit (Kyunghyun Cho (2014)), with the best results obtained by taking the last state of the network. Our architecture works best when concatenating the encodings of the claim and its evidence.

The results obtained with the claim verification evaluation are coherent with the ones obtained with the standard metrics.

Bibliography

Christos Christodoulopoulos Arpit Mittal James Thorne, Andreas Vlachos. Fever: a large-scale dataset for fact extraction and verification. 2018.

Caglar Gulcehre Dzmitry Bahdanau Fethi Bougares Holger Schwenk Yoshua Bengio Kyunghyun Cho, Bart van Merrienboer. Learning phrase representations using rnn encoder–decoderfor statistical machine translation. 2014.