CSE 635

Natural Language Processing and Text Mining

Project Proposal

Fact Extraction and Automated Claim Verification

Team: tmp

Members:

Anas, Anirudh, Snigdha

Overview

The increase in amounts of textual information available combined with the ease of sharing it through the web has increased the demand for verification, also known as *fact-checking*. An abundance of incorrect information can plant wrong beliefs in individual citizens and lead to a misinformed public, instigating chaos amongst the public, organizations, and countries. In this context, technology to automate fact-checking and source verification is of great interest to both media consumers and publishers. While it has received a lot of attention in the context of journalism, verification is important for other domains, e.g. information in scientific publications, product reviews, etc.

In order to advance research in this direction, a data set for claim verification - Fact Extraction and VERification (FEVER) was launched. This dataset contains 185,445 claims manually verified against the introductory sections of Wikipedia pages each of which comes with several evidence sets. An evidence set consists of facts, i.e. sentences from Wikipedia articles that jointly support or contradict the claim. On the basis of any one of its evidence sets, each claim is labeled as Supported, Refuted, or NotEnoughInfo if no decision about the veracity of the claim can be made.

Project Objectives

- Find and retrieve Wikipedia pages that are most relevant to the claim.
- Extract a set of sentences from the retrieved Wikipedia pages that support or refute the claims. The set of sentences form the evidence for the claim.
- Using this evidence, classify the claim as Supported and Refuted.
- If there isn't sufficient evidence to support or refute a particular claim, label the claim as NotEnoughInfo.
- Solve the three major sub-tasks:
    - Document Retrieval: Given a claim, find Wikipedia articles containing information about this claim.
    - Sentence Selection: From the retrieved articles, extract facts in the form of sentences that are relevant for the verification of the claim.
    - Recognizing Textual Entailment: On the basis of the collected sentences (facts), predict the labels for the claim.
- To evaluate the performance of the system, use the FEVER scoring program. Also, understand the evaluation metrics.

Literature Review

The FEVER system encompasses three different components: *Document Retrieval*, *Sentence (evidence) Selection*, and *Recognizing Textual Entailment*. The FEVER Shared Task required participants to develop systems to label claims with the correct class (Supported, Refutes or NotEnoughInfo) and also return the sentence(s) forming the necessary evidence for the assigned label (from Wikipedia). Out of 23 competing teams, 19 teams scored higher than the previously published baseline.

Numerous research papers of teams who overcame the baseline accuracy were read, discussed and evaluated. This report first briefly discusses the *baseline* approach and then summarizes the approach of some of the teams that participated in the FEVER Shared Task challenge.

Baseline:

Document Retrieval

Document retrieval is defined as the matching of some stated user query against a set of free-text records[1].

For fact-checking and verification, the project focuses on retrieving evidence related to the claim. As the baseline approach, DrQA's document retrieval component is used (PyTorch based implementation of DrQA, ACL 2017 Reading Wikipedia to Answer Open-Domain Questions).

1. Create a simple inverted index lookup
2. Perform term-vector model scoring
3. Compare the claim and evidence as TF-IDF weighted bag-of-words vectors
4. Improve by taking local word order into account with n-gram features

The FEVER baseline returns k-nearest (k=5) documents for a query using cosine similarity between binned unigram and bigram Term-Frequency Inverse Document Frequency (TF-IDF) vectors.

Sentence Selection

Sentence selection is the task of identifying sentences that contain the answer to a given question or claim[2].

In order to verify a claim, the project aims at selecting specific sentences from the retrieved documents to support or refute the claim. We extract the top l-most similar sentences from the k-most relevant documents using TF-IDF similarity.

1. Rank sentences based on TF-IDF similarity to the claim
2. Sort most similar sentences first and tune a cut-off using validation similarity on the development set
3. Evaluate both DrQA and a simple unigram TF-IDF implementation to rank for sentence selection

Recognizing Textual Entailment

Recognizing Textual Entailment or RTE is a generic task of capturing major semantic inference needs in Natural Language Processing applications. This means that given two fragments of text, recognize whether the meaning of one fragment of text is entailed (can be inferred) from another text.[3]

In order to perform RTE, the baseline approach is to use decomposable attention (DA). Decomposable attention simply aligns bits of local text substructure and then aggregates this information.

1. For each word that is represented by an embedding vector, create a soft alignment matrix using neural attention
2. Use (soft) alignment to decompose the task into sub-problems that are solved separately
3. Merge the results of sub-problems to produce the final classification

The above modules can be graphically represented as:

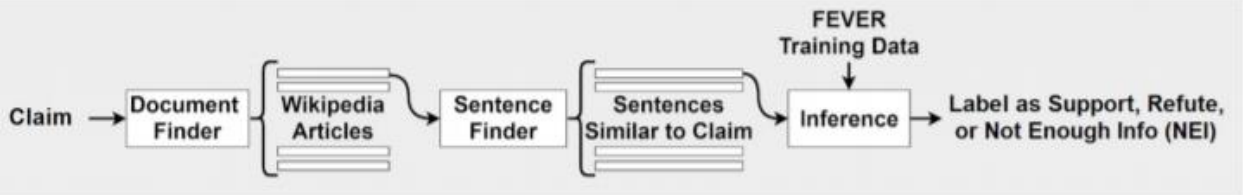

Figure 1 Visual representation of Fact Checking and Verification System

Claim: The Rodney King riots took place in the most populous county in the USA.

[wiki/Los_Angeles_Riots]

The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.

[wiki/Los_Angeles_County]

Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.

Verdict: Supported

Figure 2 Example claim from the FEVER shared task:
a manually verified claim that requires evidence from multiple Wikipedia pages

Team Specific System Description

1. **UNC-NLP** [4]

   This team set out to improve upon the previous state of the art which used an SNLI (Stanford Natural Language Inference) model. Instead of using the NLI model, they used three Neural Semantic Matching Networks (NSMN).

   Document Retrieval: Select documents based on matched keywords between the title and claim. Rank the selected documents giving dis-ambiguous documents a higher score. Pass the ranked documents and compare a combination of a title and the first line with the claim using the NSMN. Additionally, the external page view count was used to further rank the articles.

   Sentence Selection: Passing the sentences of the selected documents through an NSMN model to classify if the statements are relevant and ignoring sentences that score below a certain threshold.

   Recognizing Textual Entailment: Creating evidence set by concatenating all the sentences and comparing them to the claim using the NSMN network with a combination of a 30 dimension WordNet, 5 dimensions embedding to create unique IDs, and the scores produced by previous layers.

2. **Athene UKP TU Darmstadt** [5]

   Document Retrieval: Extract noun phrases using AllenNLP parser and fetch documents using the MediaWiki API. Discard documents whose titles don't match with the claim after stemming to reduce noise.

   Sentence Selection: An LSTM model is trained using Hinge loss and negative sampling to select sentences. For training, negative samples are generated by randomly selecting sentences from documents.

   Recognizing Textual Entailment: Combine the sentences and claim generated and then apply enhanced LSTM to each pair and then used average and max pooling and then feed the result to a multi-layer perceptron for classification.

3. **Papelo** [6]

   This team's novelty lies in the fact that they've used a high precision entailment classifier based using transformer networks pre-trained by OpenAI. One of the caveats of this implementation is to ignore instances that required multiple statements to resolve a claim.

   Document Retrieval: A combination of TF-IDF, named entities in titles, disambiguation, and capitalized expressions occurring in the first fifty lines of the document to get the most relevant articles.

   Sentence Selection: Top five relevant sentences from the articles are selected using TF-IDF.

   Recognizing Textual Entailment: Used a custom version of OpenAI's pre-trained transformer network which was trained on the Fever Dataset.

4. **Ohio State University** [7]

   Document Retrieval: Gather documents based on keywords/phrases in the claim that appear in the wiki dump.

   Sentence Selection: Break the claim into named entities + nouns and find any sentence(s) that match those entities.

   Recognizing Textual Entailment: Using decomposition attention for natural language inference and adding additional vectors based on entities, classify the claims into one of the three labels. Not Enough

Info (NEI) sentences are discarded and the highest scored label of the remaining sentences is assigned to the claim.

5. **GESIS Cologne** [8]

   Document Retrieval and Sentence Selection: Index every sentence including information about the article where the sentence is from on Solr and create queries based on the name entities and noun chunks of the claim.

   Recognizing Textual Entailment: Use decomposition attention (as in baseline model), however instead of comparing claim with all top 5 sentences at once, treat every sentence separately. Join the results of the top 5 sentences with an ensemble learner including the rank of the sentence retriever of the Wikipedia sentences.

6. **Directed Acyclic Graph** [8]

   Document Retrieval: Use similar TF-IDF vectors of Wikipedia documents and claims, and documents whose names are similar to named entities mentioned in the claim to retrieve documents.

   Sentence Selection: Supply the sentences in the retrieved documents to a decomposition attention-based textual entailment recognition module. This module calculates the probability of each sentence supporting the claim, contradicting the claim and not providing any relevant information.

   Recognizing Textual Entailment: Features are compared using the probabilities and used by a Random Forest Classifier to determine the overall truthfulness of the claim. The sentences which support this classification are returned as evidence.

	Evidence (%)			**Label**	**Fever**
Team	**Precision**	**Recall**	**F1**	**Accuracy(%)**	**Score(%)**
UNC-NLP	42.27	70.91	52.96	**68.21**	**64.21**
Athene UKP TU Darmstadt	23.61	**85.19**	36.97	65.46	61.58
Papelo	**92.18**	50.02	**64.85**	61.08	57.36
Ohio State University	77.23	47.12	58.53	50.12	43.42
GESIS Cologne	12.09	51.69	19.60	54.15	40.77
Directed Acyclic Graph	51.91	36.36	42.77	51.36	38.33

Table 1 Result comparison of teams that participated in FEVER Shared Task challenge

Data Set

The dataset built for this challenge consists of claims that have already been labeled by human annotators. The dataset is split into ~*79%* training data, ~*5%* development data, ~*5%* test data and ~*11%* held out data. The table below, which was taken from the FEVER challenge paper[5], gives the exact figures. Each sample in the dataset contains a claim with the following fields:

- Claim: the text of the actual claim
- Verifiable: indicates whether the claim is labeled as *"Supported"* or *"Refuted"*
- Label: the label of the claim (one of *"Supported"*, *"Refuted" or "NotEnoughInfo"*)

- Evidence: this field is left blank in the case where the claim's label is *"NotEnoughInfo"*. Otherwise, it is a set of evidence groups containing the necessary evidence to support or refute the claim. Each evidence group contains the following fields:
    - Page name: the name of the article that contains the accompanying evidence sentence
    - Page number: a number indicating the sentence number in the article
    - Two additional fields representing the annotation job and id which are used internally by the FEVER scoring system and are ignored by the participant

Split	SUPPORTED	REFUTED	NEI
Training	80,035	29,775	35,639
Dev	3,333	3,333	3,333
Test	3,333	3,333	3,333
Reserved	6,666	6,666	6,666

Figure 3 Dataset split sizes for SUPPORTED, REFUTED and NOTENOUGHINFO (NEI) classes

Evaluation Methodology

The measurement of the quality of predictions takes into account how well each of the three major components in the system performs. For each claim, there is a set of evidence groups. Each evidence group contains ground truth evidence sentences. The scoring mechanism discussed below considers each evidence group separately until either a matching group is found, or the prediction fails to provide a complete group of evidence sentences in any of the ground truth sets.

Document retrieval and sentence selection are evaluated jointly by calculating the macro precision, macro recall and F1 score of the retrieved sentences that are predicted as evidence in the case where the claim is predicted to be either *"Supported"* or *"Refuted"*. In the case where the claim cannot be supported or refuted (i.e. when the ground-truth label of the claim is *"Not Enough Info"),* the claim does not factor into the recall and precision metrics, since there is no evidence to check against. For both recall and precision, a maximum of five sentences is considered from the set of predicted evidence sentences. Once the precision and recall are calculated for each claim as per the formulae below, the total precision and recall are then considered for all instances of claims where the ground truth label is either *"Supported"* or *"Refuted".*

RTE is measured through the accuracy of the label classifications for a given set of claims. In addition to the above metrics, the FEVER score was introduced, which aims to provide a measurement that takes into account both recall and accuracy as a single comprehensive unit. The evaluation metrics are given by the formulae below.

$$Macro\ Precision\ (for\ each\ pred.) = \frac{\#\ Sentences\ correctly\ predicted\ as\ evidence}{\#\ Sentences\ correctly\ predicted\ as\ evidence\ +\ \#\ Sentences\ falsely\ predicted\ as\ evidence}$$

$$Macro\ Recall\ (for\ each\ prediction) = 1\ if\ true\ evidence \subseteq predicted\ evidence; 0\ otherwise$$

$$Total\ Precision\ = [\ \sum_{prediction \in V} Macro\ Precision(prediction)\ ] \div |V|$$

$$Total\ Recall\ = [\ \sum_{prediction \in V} Macro\ Recall(prediction)\ ] \div |V|$$

$$In\ the\ above\ equations, V = set\ of\ claims\ for\ which\ label\ \neq NotEnoughInfo$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad Accuracy = \frac{\#\ Correctly\ predicted\ claims}{Total\ \#\ of\ claims}$$

$$FEVER\ Score = \frac{\#\ Claims\ with\ correct\ label\ and\ correct\ evidence}{Total\ \#\ of\ claims}$$

Proposed System

The FEVER challenge was undertaken by several teams, and as the results show, even the top participants were not able to achieve very high scores. This proves that the tasks are very challenging, even given advanced high-end models and sophisticated representation techniques. The top participant in the challenge achieved a FEVER score of *64.21%*, and for the purposes of this project, this score will be considered state-of-the-art.

The proposed system will be built in an attempt to improve upon the state-of-the-art, or at least advance the results beyond the FEVER baseline score of *27.45%*. The components proposed in this system have been chosen after consideration of the performance of the individual components of the systems of the previous participants. Each stage of the system pipeline is described below in order:

- Document retrieval: given a claim, document retrieval takes place over the following stages:
    - Named Entity Recognition as well as Noun Phrase Extraction are performed on the claim, resulting in a set of phrases
    - Extracted entities and phrases are searched on Wikipedia, and the titles of the matches are retained in a set of candidate document titles
    - TF-IDF is applied between each title in the retrieved set of titles and the claim itself, and top-ranking documents according to this score are selected for the next stage
- Sentence selection: given the retrieved set of documents, sentence selection takes the following course:
    - The first *k* sentences in each document are selected, where *k* is a hyperparameter that is subject to change based on development experiments
    - Features are constructed that consider each sentence in relation to the claim and the document in which it was found, such as its position within the article, its length, and whether the document title appears in it
    - The sentences are ranked by feeding their features to a Multi-layer Perceptron
    - The top 5 highest-ranking sentences are chosen for the next stage
- Recognizing textual entailment: once the candidate sentences are selected, they go through the final stages of the pipeline:
    - The sentences are represented through a GloVe embedding [9]
    - The embeddings are fed to a stacked biLSTM network, which assigns a label to each sentence in reference to the claim
    - The scores are fed to a Multi-layer Perceptron, which finally produces the scores for each of the three labels

Architecture

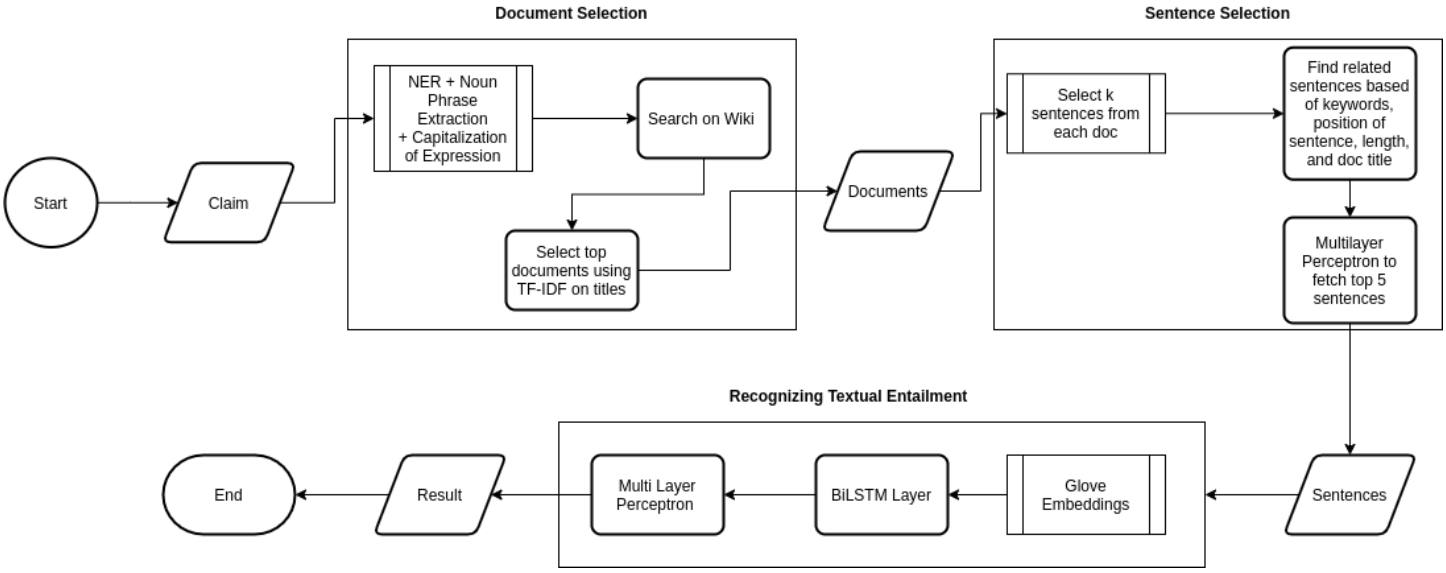

Figure 4 Architecture

Project Plan

The Gantt Chart below summarizes the project plan.

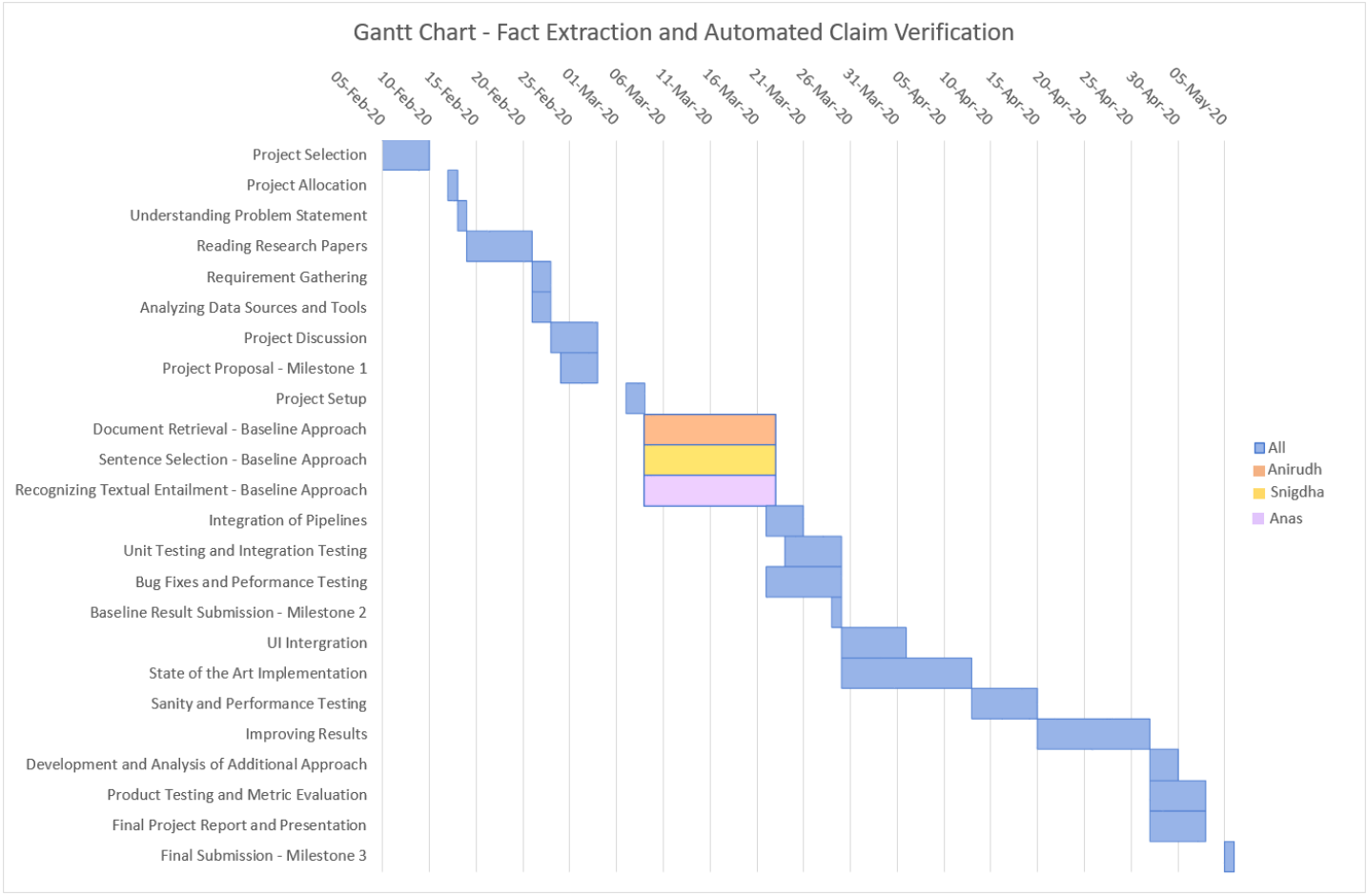

Figure 5 Gantt Chart

Conclusion

In this proposal, the FEVER challenge was introduced as a task that is meant to address the recently growing number of fake news outlets. The FEVER dataset, which contains *~185* thousand samples, was introduced, and the evaluation method used by the FEVER challenge judges was explained. The approaches taken by previous participants to tackle the challenge were discussed, with the highest-ranking team achieving a score that improved significantly upon the baseline.

A new approach that combines some of the best methods used by the previous participants was devised in an attempt to advance the state-of-the-art or provide a robust baseline system that outperforms the one proposed by the makers of the challenge. The team plans to produce a fully functioning system following the timeline outlined above. The team plans to adhere to the proposed system insofar as the experiments produce the anticipated results. Some components might be tweaked if experiments reveal that other alternatives might prove more promising, but no major changes to the overall system will be made.

References

[1] https://en.wikipedia.org/wiki/Document_retrieval

[2] Deep Learning for Answer Sentence Selection

[3] https://aclweb.org/aclwiki/Recognizing_Textual_Entailment

[4] Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In Association for the Advancement of Artificial Intelligence (AAAI).

[5] Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

[6] Christopher Malon. 2018. Team Papelo: Transformer networks at FEVER. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 109–113, Brussels, Belgium. Association for Computational Linguistics.

[7] Luken, Jackson. "QED: A Fact Verification and Evidence Support System." Electronic Thesis or Dissertation. Ohio State University, 2019. OhioLINK Electronic Theses and Dissertations Center. 28 Feb 2020.

[8] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The Fact Extraction and VERification (FEVER) Shared Task. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

[9] Pennington J., Socher R. Manning C.. 2014. *GloVe: Global Vectors for Word Representation.* https://nlp.stanford.edu/pubs/glove.pdf

[10] DrQA/README.md at master · facebookresearch/DrQA

[11] Thorne J., Vlachos A., Christodoulopoulos C., Mittal A. 2018. *FEVER: a large-scale dataset for Fact Extraction and VERification*. arXiv:1803.05355 [cs.CL] (Dec. 2018). https://arxiv.org/abs/1803.05355

[12] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. *Reading wikipedia to answer open-domain questions*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, pages 1870–1879. https://doi.org/10.18653/v1/P17-1171.