UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF)

Takuma Yoneda
Computational Intelligence Lab.
Toyota Technological Institute
sd18438@toyota-ti.ac.jp

Jeff Mitchell, Johannes Welbl,
Pontus Stenetorp, Sebastian Riedel
Dept. of Computer Science
University College London
{j.mitchell, j.welbl,
p.stenetorp, s.riedel}@cs.ucl.ac.uk

Abstract

In this paper we describe our 2^{nd} place FEVER shared-task system that achieved a FEVER score of 62.52% on the provisional test set (without additional human evaluation), and 65.41% on the development set. Our system is a four stage model consisting of document retrieval, sentence retrieval, natural language inference and aggregation. Retrieval is performed leveraging task-specific features, and then a natural language inference model takes each of the retrieved sentences paired with the claimed fact. The resulting predictions are aggregated across retrieved sentences with a Multi-Layer Perceptron, and re-ranked corresponding to the final prediction.

1 Introduction

We often hear the word "Fake News" these days. Recently, Russian meddling, for example, has been blamed for the prevalence of inaccurate news stories on social media,[1] but even the reporting on this topic often turns out to be fake news (Uberti, 2016). An abundance of incorrect information can plant wrong beliefs in individual citizens and lead to a misinformed public, undermining the democratic process. In this context, technology to automate fact-checking and source verification (Vlachos and Riedel, 2014) is of great interest to both media consumers and publishers.

The Fact Extraction and Verification (FEVER) shared task provides a benchmark for such tools, testing the ability to assess textual claims against a corpus of around 5.4M Wikipedia articles. Each claim is labeled as `SUPPORTS`, `REFUTES` or `NOT ENOUGH INFO`, depending on whether relevant evidence from the corpus can support/refute it. Systems are evaluated on the proportion of claims for which both the predicted label is correct *and* a complete set of relevant evidence sentences has been identified.

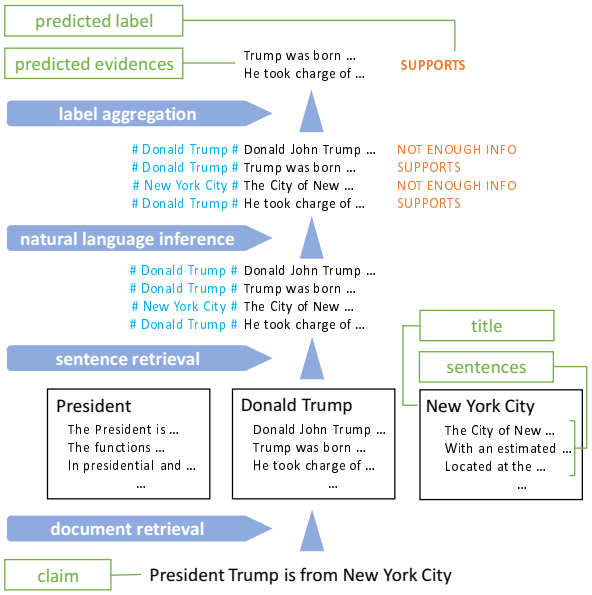

Figure 1: Illustration of the model pipeline for a claim.

The original dataset description paper (Thorne et al., 2018) evaluates a simple baseline system that achieves a score of ∼33% on this metric, using tf-idf based retrieval to find the relevant evidence and a natural language inference (NLI) model to classify the relation between the returned evidence and the claim. Our system attempts to improve on this baseline by addressing two major weaknesses. Firstly, the original retrieval component only finds a full evidence set for 55% of claims. While tf-idf is an effective task agnostic approach to information retrieval, we find that a simple linear model using task-specific features is able to achieve much stronger performance. Secondly, the NLI component uses an overly simplistic strategy for aggregating retrieved evidence, by simply concatenating all the sentences into a single paragraph. Instead, we employ

[1] https://www.bbc.co.uk/news/world-us-canada-41821359

Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 97–102
Brussels, Belgium, November 1, 2018.

an explicit aggregation step to combine the knowledge gained from each evidence sentence. These improvements allow us to achieve a FEVER score of 65.41% on the development set, and 62.52% on the test set.

2 System Description

Our system is a four stage model consisting of document retrieval, sentence retrieval, NLI and aggregation. Document retrieval attempts to find the name of a Wikipedia article in the claim, and then ranks each article based on capitalisation, sentence position and token match features. A set of sentences are then retrieved from the top ranked articles, based on token matches with the claim and position in the article. The NLI model is subsequently applied to each of the retrieved sentences paired with the claim, giving a prediction for each potential evidence sentence. The respective predictions are aggregated using a Multi-Layer Perceptron (MLP), and the sentences are finally reranked so that the evidence which is consistent with the final prediction are placed at the top.

2.1 Document Retrieval

Our method begins by building a dictionary of article titles, based on the observation that the FEVER claims frequently include the title of a Wikipedia article containing the required evidence. These titles are first normalised by lowercasing, converting underscores to spaces and truncating to the first parenthesis if present. An initial list of potential articles is then constructed by detecting any such title in the claim. For each article, the probability of containing the gold evidence is predicted by a logistic regression model, using as features the position and capitalisation within the claim, presence of stop words, and token match counts between the first sentence of the article and the claim. Likewise we include the same counts also for the rest of the article as features, alongside whether the name was truncated, and whether the excised words are mentioned in the claim (e.g., "*Watchmen*" *vs* "*Watchmen (film)*").

The model is trained on a balanced set of positive and negative examples drawn from the training set, and the top-ranked articles are then passed on to the sentence retrieval component.

This process is related to, but goes substantially beyond entity recognition and linking (Mendes et al., 2017). These processes attempt to identify

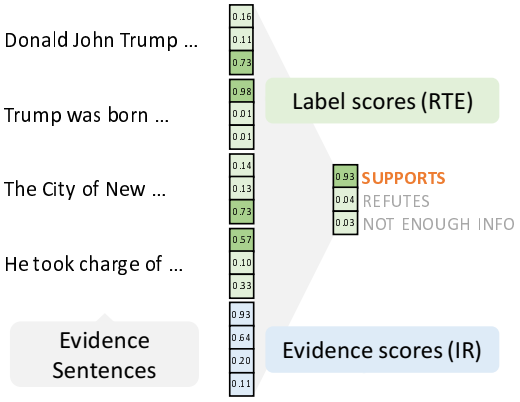

Figure 2: Overview: Aggregation Network

mentions of names from a limited class of entities (e.g. people, places, organisations). In our case, the mentions cover a much wider range of lexical items, including not only names but also common nouns, verbs or adjectives. Nonetheless, both types of model share the objective of finding mentions and linking them to a reference set.

2.2 Sentence Retrieval

We observed that many evidence sentences appear at the beginning of an article, and they often mention the article title. We thus train a logistic regression model, using as features the position of the sentence within the article, its length, whether the article name is present, token matching between the sentence and the claim, and the document retrieval score. The top-ranked sentences from this model are then passed to the subsequent NLI stage.

2.3 Natural Language Inference (NLI)

In this component, an NLI model predicts a label for each pair of claim and retrieved evidence sentence. We adopted the Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017) as NLI model. ESIM employs a bidirectional LSTM (BiLSTM) to encode premise and hypothesis, and also encodes local inference information so that the model can effectively exploit sequential information. We also experimented with the Decomposable Attention Model (DAM) (Parikh et al., 2016) — as used in the baseline model, however ESIM consistently performed better. The Jack the Reader (Weissenborn et al., 2018) framework was used for both DAM and ESIM.

We first pre-trained the ESIM model on the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), and then fine-tuned

on the FEVER dataset. We used 300-dimensional pre-trained GloVe word embeddings (Pennington et al., 2014). As training input, we used gold evidence sentences for SUPPORTS and REFUTES samples, and retrieved evidence sentences for NOT ENOUGH INFO.

It is worth noting that there are two kinds of evidences in this task. The first is a complete set of evidence, which can support/refute a claim, and can consist of multiple sentences. The second is incomplete evidence, which can support or refute the claim only when paired with other evidence.

The baseline model (Thorne et al., 2018) simply concatenates all evidence sentences and feeds them into the NLI model, regardless of their evidence type. In contrast, we generate NLI predictions individually for each predicted evidence, thus processing them in parallel.

Furthermore, we observed that evidence sentences often include a pronoun referring to the subject of the article without explicitly mentioning it. This co-reference is opaque to the NLI model without further information. To resolve this, we prepend the corresponding title of the article to the sentence, along with a separator as described in Figure 1. We also experimented with adding line numbers to represent sentence position within the article, which did not, however, improve the label accuracy.

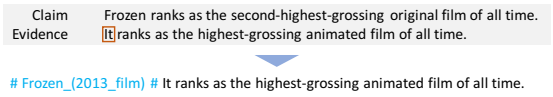

Figure 3: Illustration for the co-reference problem with individual sentences: What 'it' refers to is not obvious for a NLI model.

2.4 Aggregation

In the aggregation stage, the model aggregates the predicted NLI labels for each claim-evidence pair and outputs the final prediction.

The NLI model outputs three prediction scores per pair of sentences, one for each label. In our aggregation model, these scores are all fed into an MLP, alongside the evidence confidence scores for each of the (ranked) evidence sentences. Since the label balance in the training set is significantly biased, we give the samples training weights which are inversely proportional to the size of their respective class. We also experimented with drawing samples according to the size of each class, but using the full training data with class weights performed better. The final MLP model contains 2 hidden layers with 100 hidden units each and Rectified Linear Unit (ReLU) nonlinearities (Nair and Hinton, 2010). We observed only minor performance differences when modifying the size and number of layers of the MLP.

Aside from this neural aggregation module, we also tested *logical aggregation*, *majority-vote* and *top-1 sentence*. In *logical aggregation*, our module takes the NLI predictions for all evidence sentences, and outputs either SUPPORTS or REFUTES if at least one of them has such a label, and NOT ENOUGH INFO if all predictions have that label. In cases where both SUPPORTS and REFUTES appear among the predictions, we take one from the highest ranked evidence. *Majority-vote* counts the frequency of labels among prediction and outputs the most frequent label. 15 predicted evidence sentences are used in each aggregation method.

3 Results

3.1 Aggregation Results

Table 1 shows the development set results of our model under the different aggregation settings. Note that the Evidence Recall and F1 metrics are calculated based on the top 5 predicted evidences. We observe that the *Majority-vote* aggregation method only reaches 43.94% of FEVER Score and 45.36% of label accuracy, either of which are much lower than other methods. Since there are only a few gold evidence sets for most claims, the majority of NLI predictions tend to be NOT ENOUGH INFO, rendering a majority aggregation method impractical.

Conversely, the *top-1 sentence* aggregation only uses the top-ranked sentence alone to form a label prediction. In this scenario a failure of the retrieval component is critical, nevertheless the system can achieve a FEVER score of 63.36%, leaving a large gap to the baseline model (Thorne et al., 2018). The *logical aggregation* improves slightly over omitting aggregation entirely (*top-1 sentence*). However, the neural aggregation module produces the best overall results, both in terms of FEVER score and label accuracy. This demonstrates the advantage of using a neural aggregation model operating on individual NLI confidence scores, compared to the more rigid use of only the predicted labels in *logical aggregation*.

Aggregation Method	FEVER Score	Label Accuracy	Evidence Recall	Evidence F1
Majority-vote	43.94	45.36	83.91	35.36
Top-1 sentence	63.36	66.30	84.62	35.72
Logical	64.29	68.26	85.03	36.02
MLP	65.41	69.66	84.54	35.84

Table 1: Development set scores for different aggregation methods, all numbers in percent.

Gold \ Prediction	*supports*	*refutes*	*not enough info*	**Total**
supports	5,345	336	985	6,666
refutes	827	4,196	1,643	6,666
not enough info	1,288	989	4,389	6,666
Total	7,460	5,521	7,017	19,998

Table 2: Confusion matrix on the development set.

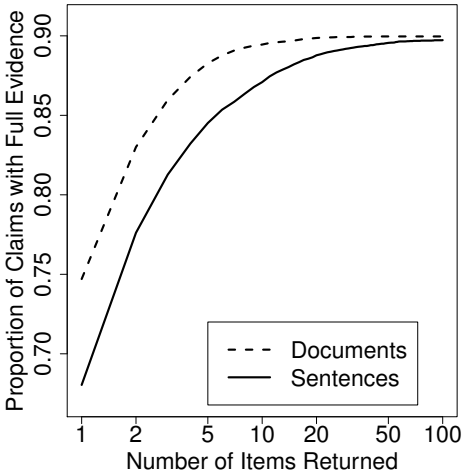

Figure 4: Performance of retrieval models on the development set.

Finally, after obtaining the aggregated label with the MLP, the model re-sorts its evidence predictions in such a way that those evidences with the same predicted label as the final prediction are ranked above those with a different label (see upper part of Figure 1). We observed that this re-ranking increased the evidence recall by 0.18 points (when used with MLP aggregation).

The overall FEVER score is the proportion of claims for which both the correct evidence is returned and a correct label prediction is made. We first describe the performance of the retrieval components, and then discuss the results for NLI.

3.2 Retrieval Results

On the development set, the initial step of identifying Wikipedia article titles within the text of the claim returns on average 62 articles per claim. These articles cover the full evidence set in 90.8% of cases and no relevant evidence is returned for only 2.9% of claims. Ranking these articles, using the model described above, achieves 81.4% HITS@1, and this single top-ranked article contains the full evidence in 74.7% of instances. Taking the text of the 15 best articles and ranking the sentences achieves 73.7% HITS@1, which is equivalent to returning the full evidence for 68% of claims. Figure 4 illustrates the performance of the IR components as the number of returned items increases.

4 Error Analysis

Table 2 shows the confusion matrix for the development set predictions. We observe that the system finds it easiest to classify instances labelled as `SUPPORTS`, whereas using the `NOT ENOUGH INFO` label correctly is most difficult.

We next describe some frequent failure cases of our model in the description below.

Limitations of word embeddings. Numerical expressions like years (1980s *vs* 80s) or months (January *vs* October) tend to have similar word embeddings, rendering it is difficult for a NLI model to distinguish them and correctly predict `REFUTES` cases. This was the most frequent error type encountered in the development set.

Confusing sentence. An NLI model aligns two sentences and predicts their relationship. For example, when two sentences are "Bob is in his house and he cannot sleep" and "Bob is awake", a

model can conclude that the second sentence follows from the first one by simply aligning *Bob* with *Bob* and *cannot sleep* with *awake*. However, it sometimes fails to capture a correct alignment, which results in a fail prediction. For example, "Andrea Pirlo is an *American* professional footballer" *vs* "Andrea Pirlo is an *Italian* professional footballer who plays for an *American* club."

Sentence Complexity. In some cases, just taking an alignment is not enough to predict the correct label. In these cases, the model needs to capture the relationship between multiple words. For example, "Virginia keeps all computer chips manufactured within the state for use in Virginian electronics." *vs.* "Virginia's computer chips became the state's leading export by monetary value."

5 Future Work

For the model to read sentences that includes numerical expressions correctly, it could be helpful to explicitly encode the numerical expression and obtain a representation that captures the numerical features (Spithourakis and Riedel, 2018). Leveraging context-dependent pre-trained word embeddings such as ELMo (Peters et al., 2018) could help dealing better with more complex sentences.

6 Conclusion

In this paper, we described our FEVER shared-task system. We employed a four stage framework, composed of document retrieval, sentence retrieval, natural language inference, and aggregation. By applying task specific features for a retrieval model, and connecting an aggregation network on top of the NLI model, our model achieves a score of 65.41% on the development set and 62.52% on the provisional test set.

Acknowledgments

This work has been supported by the European Union H2020 project SUMMA (grant No. 688139), by an Allen Distinguished Investigator Award, and by an Engineering and Physical Sciences Research Council scholarship.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. Association for Computational Linguistics.

Afonso Mendes, David Nogueira, Samuel Broscheit, Filipe Aleixo, Pedro Balage, Rui Martins, Sebastiao Miranda, and Mariana S. C. Almeida. 2017. Summa at tac knowledge base population task 2017. In *Proceedings of the Text Analysis Conference (TAC) KBP 2017*, Gaithersburg, Maryland USA. National Institute of Standards and Technology.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA. Omnipress.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Georgios P. Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. *CoRR*, abs/1805.08154.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*.

David Uberti. 2016. Washington post fake news story blurs the definition of fake news. *Columbia Journalism Review*.

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.

Dirk Weissenborn, Pasquale Minervini, Tim Dettmers, Isabelle Augenstein, Johannes Welbl, Tim Rocktaschel, Matko Bosnjak, Jeff Mitchell, Thomas Demeester, Pontus Stenetorp, and Sebastian Riedel.

2018. Jack the Reader A Machine Reading Framework. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL) System Demonstrations*.