Supervised, Heuristic, Entity-Recognising & Linking, Ontological Contradiction Killer (SHERLOCK)

Matthew Farrugia (694719), Sergio Rodriguez (842278)

Abstract

We develop an automated fact verification system for an in-class task derived from the FEVER challenge. Our system is a five-step pipeline comprising article retrieval, article selection, sentence selection, claim-evidence assessment, and label aggregation. We frame steps 2-5 as supervised classification tasks with features capturing entity linking, lexical ontology, co-reference resolution, and more. Our system achieves 77.4% document selection F_1, 61.9% sentence selection F_1 and 51.8% label accuracy on the class test set, *competitive results under all metrics.*

1 Introduction

Our class project is to develop an automated fact verification system. We are provided a dataset of 165.5k claims and a corpus of 5.1m Wikipedia article introductions, and are tasked with labelling each claim as "supported" (SUP), "refuted" (REF), or "not enough information" (NEI) with respect to the information in the corpus. If the claim is supported or refuted, we must also identify corpus sentences informing the judgement. This task and dataset are derived from the Fact Extraction and VERification (FEVER) task (Thorne et al., 2018a), a current benchmark for evaluating fact-verification systems.

In this report, we describe our approach to the project task. We follow a multi-step approach similar to the FEVER baseline, breaking down the task into a total of five steps. Inspired by top entrants to the FEVER shared task competition (Thorne et al., 2018b), and with original enhancements, our system overcomes two major weaknesses of the FEVER baseline system. First, we treat evidence retrieval as a sequence of task-specific classification problems, incorporating additional features beyond plain TF-IDF similarity scores. Second, we improve upon the baseline's *evidence-concatenating* claim-labelling step by instead evaluating all claim-evidence pairs and employing a dedicated claim-label aggregation classifier, leveraging knowledge from upstream.

2 System Description

Figure 1 illustrates our system. The pipeline divides into two high-level components: **(1) Evidence retrieval**; and **(2) Claim labelling**. In this section, we motivate and explain the design of each step.

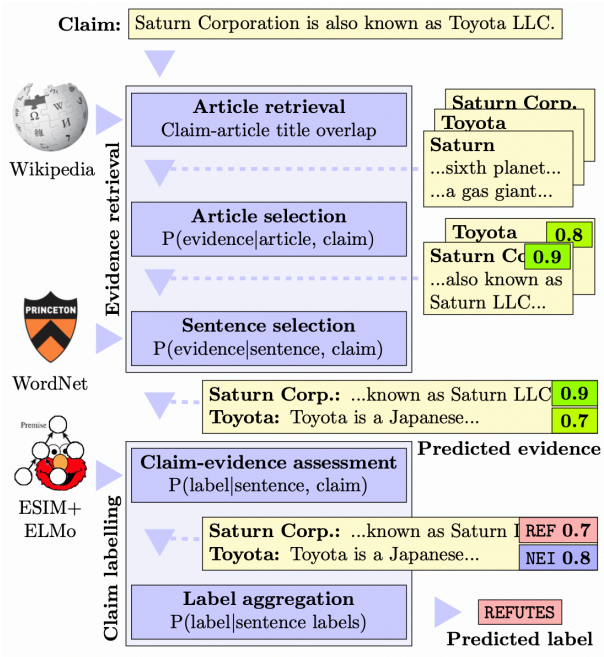

Figure 1: Illustration of our five-step pipeline system.

2.1 Article Retrieval

To enable a supervised classification approach to evidence selection we use Boolean retrieval as an efficient pre-filter, identifying a small selection of potentially relevant articles as classification inputs.

Based on the heuristics that: (1) Entity mentions in claims are often central to verification; and (2) Wikipedia articles contain most of the information available on their titular entity, we construct an inverted index over the capitalised terms in all 5.1 million Wikipedia article titles. We split on underscores and tokenise, with special handling for terminating periods (always denoting abbreviations in titles, e.g. "Tesla, Inc.", but separated by a tokeniser).

At prediction time, we use this index to retrieve all articles with titles appearing as a substring within the claim, excluding parenthesised text in the title. To speed up retrieval, we do not search the index for stopwords or lower-case terms from the claim, or for terms occurring in more than 15,000 titles—while these words may occur in matching titles, they are often accompanied by rarer words from the claim, or else the articles are unlikely to relate to the claim.

2.2 Article Selection

Given the set of retrieved articles for a claim, we train a classifier to predict the probability of each article containing evidence sentences.

As a baseline, we implement a logistic regression classifier following UCL's HexaF system (Yoneda et al., 2018). Features include title position within claim, title stopwords and parenthesised text, and normalised token overlap between claim and article text (with special treatment for first sentences). Like UCL, we train on a balanced set of articles. We draw negative examples from upstream article retrieval.

We incorporate additional features geared towards entity disambiguation: Capitalisation of adjacent terms to the title within claim text; IDF-weighted token overlap between claim and article text; Token bigram overlap; and capitalised token overlap.

We further improve performance by creating an imbalanced training set (140k positive examples and 1015k negative examples) more closely reflecting the prediction-time distribution, and by employing a multi-layer perceptron (MLP) classifier with a single hidden layer of 30 units and ReLU nonlinearities (tuning architecture with the development data). We also experiment with a Random Forest classifier and with different training label distributions.

2.3 Sentence Selection

Given the sentences of selected articles for a claim, we train a classifier to predict the probability of each sentence being evidence (for or against) the claim. Chosen sentences become the pipeline's predicted evidence. Moreover, this step serves as a pre-filter increasing the quality of sentences passed on to claim labelling, which is sensitive to irrelevant inputs.

Sentence classification features include all token overlap features from article selection at the claim-sentence level, along with the article selection score. Furthermore, we POS-tag the text and lemmatise with WordNet's morphological lemmatiser (Miller, 1995), allowing us to capture semantic relatedness between the claim and sentence with additional features: Noun overlap; cardinal number overlap; cardinal number 'shape' overlap (based on number of digits); and synonym, antonym, meronym, holonym, hyponym, and hypernym overlap, reflecting some of the meaning-altering semantic mutations applied during FEVER claim generation.

We create an imbalanced training set (with 149k positive examples and 309k negative examples, both drawn from the sentences of selected articles) to train an MLP classifier (tuning with development data to one hidden layer, 60 units). We also try other training label distributions and a logistic regression classifier.

At prediction time, we truncate the predictions to the 2 most probable sentences, observing a small increase in sentence selection F_1 and downstream label accuracy on the development data.

2.4 Claim-Evidence Assessment

We next predict claim label probabilities (SUP, REF, NEI) based on individual selected sentences, creating inputs for the final *label aggregation* step.

We use an Enhanced Sequential Inference Model (ESIM) for Textual Entailment (TE) (Chen et al., 2017) pre-trained (AllenNLP, 2018) on the SNLI dataset (Bowman et al., 2015) using ELMo embeddings (Peters et al., 2018). We also experiment with Decomposable Attention (DA) (Parikh et al., 2016), an alternative TE model.

Many evidence sentences contain pronouns referencing the article's main entity, but such anaphors are opaque to a TE model. Hence, we see performance improvements by employing a simple co-reference resolution "trick" proposed for UCL's HexaF system: We prepend the article title's entity to each sentence, separated by a colon, so *"He is the ..."* becomes *"Sherlock Holmes: He is the ..."*.

2.5 Label Aggregation

Our final classifier aggregates per-sentence claim-label probabilities from the previous step into a final claim label, leveraging upstream knowledge (scores and features from the *sentence selection* step).

Our motivation for creating an aggregation step is twofold. Firstly, it allows us to fine-tune the label prediction with this task's data instead of relying entirely on pre-trained models. Secondly, it eliminates the assumption that sentences can be sensibly concatenated to be fed into an RTE model. We experiment with this assumption using a *concatenating evidence aggregator* built with an ESIM+ELMo TE model.

We concatenate the feature vectors and label probabilities for each of the *at-most two* predicted sentences, and synthesise training examples from upstream retrieved evidence to train an MLP classifier with two hidden layers of 15 and 2 units using ReLU activation (tuning architecture on the development data). Finally, we compare to a baseline *logical aggregation* strategy following the rules described by (Yoneda et al., 2018).

3 Experiments

In this section, we detail the experimental setup for evaluating and tuning each of our models and report the results motivating our design decisions.

3.1 Article Retrieval

The article retrieval task is driven by recall. Thus, we record the average article-level evidence recall over all claims in the development data to evaluate the

effect of ignoring terms occurring in many article titles. Figure 2 shows the diminishing returns present as we approach our setting of 15,000, indicating that we are already processing most terms important for title matching using this setting. Our final model achieves average article recall of 89.2% at this step.

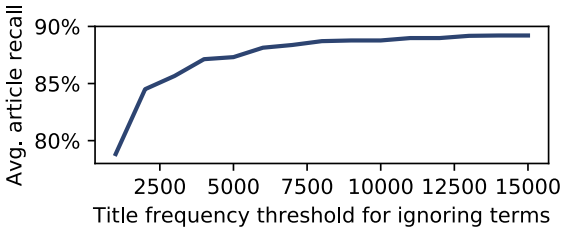

Figure 2: Average article-level evidence recall.

3.2 Article Selection

We evaluate article selection using average article-level precision, recall, and F_1 calculated over claims in the development data with inputs drawn from our best article retrieval model.

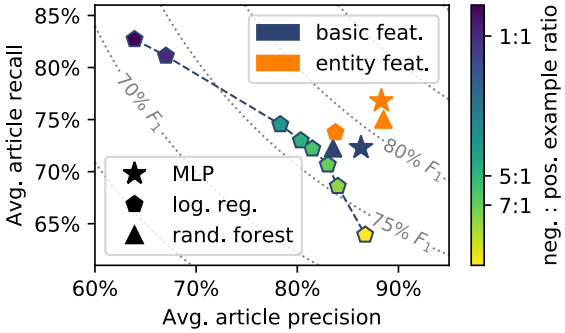

Figure 3: Article selection average precision, recall, F_1.

Notably, Figure 3 shows the significant impact of training label distribution on the precision-recall trade-off for the basic logistic regression model. The highest F_1 (76.6%) comes from training with 7 times more negative examples (cf. a balanced training set; 72.2%). We fix this distribution and explore the effect of adding entity disambiguation features on each classifier's performance. Our best model (MLP, additional features) scores 88.3% average precision and 76.8% average recall, yielding 82.1% F_1.

3.3 Sentence Selection

We evaluate sentence selection in pipeline context with input sentences from upstream selected articles, recording average sentence-level precision, recall, and F_1 over all claims in the development data.

Figure 4 highlights the effect of training label distribution on the precision-recall trade-off for each classifier. For both classifiers, we observe the best F_1 score with 2 negative examples per positive example. Regardless of training label distribution, the MLP consistently outperforms logistic regression. Our best model scores 65.6% average precision and 56.9% average recall, yielding 61.0% F_1 (before applying truncation).

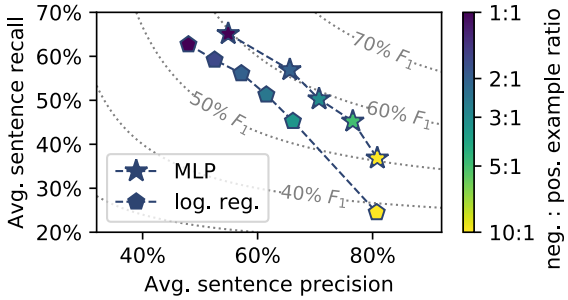

Figure 4: Sentence selection avg. precision, recall, F_1.

3.4 Claim-Evidence Assessment

We perform oracle isolation testing using the *devset* with *label accuracy* as our performance indicator. We use gold evidence input for SUP and REF claims and upstream retrieved sentences for NEI claims. We employ the baseline *logical aggregation* model described above to decouple this step from the next while still producing a label during isolation testing.

Co-ref / TE model	DA	ESIM
No co-reference resolution	55.7%	58.1%
Co-reference resolution "trick"	58.2%	**61.5%**

Table 1: Label accuracy for TE oracle experiments.

Table 1 shows results from four experiments on the impact of TE model selection and co-reference resolution. We observe ESIM outperforming DA and co-reference resolution improving accuracy; hence, we select ESIM + co-reference resolution as our claim-evidence assessment model for the pipeline.

3.5 Label Aggregation

To experiment with *label aggregation* models, we fix upstream models to their best performing parameters and conduct full-pipeline tests on the *devset* using *label accuracy* as the main performance metric.

Aggregator	Concatenating	Logical	MLP
Label Acc.	48.7%	48.7%	**52.1%**

Table 2: Results for label aggregation experiments.

Table 2 compares label accuracy performance of our best *concatenation model,* the baseline *logical aggregator* and our final *MLP aggregator.* We observe that the former strategies achieve similar performance and the MLP outperforms both. This empirical result motivates our introduction of a fifth 'label aggregation' step as discussed in section 2.5.

4 Final Approach Performance

Table 3 shows our final pipeline model's performance on the provided development and test sets. The

performance on both sets is similar, indicating we are unlikely to have overfit our models while tuning and experimenting with the development data.

	Doc. F1	Sent. F1	Lab. Accu.
Devset	77.5%	61.5%	52.1%
Testset	77.4%	61.9%	51.8%

Table 3: Final approach performance.

The test set results also show that our system surpasses the *competitive threshold* established for the in-class competition for all performance metrics.

5 Error Analysis

The confusion matrix in Figure 5 illustrates the two main classes of claim labelling errors our pipeline system makes, framing our error analysis.

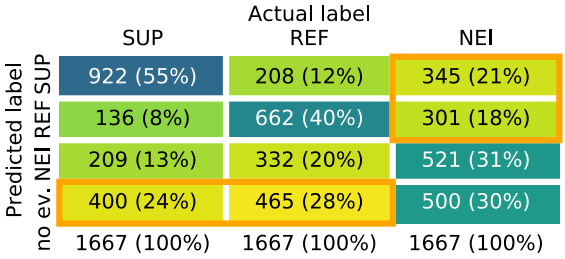

Figure 5: Confusion matrix over development data.

5.1 Evidence Retrieval

The *no ev.* row represent cases where the *evidence retrieval* steps find no evidence for a claim, in which case we cannot proceed with claim labelling, and predict NEI. Such cases represent most SUP and REF claim misclassifications: Evidence retrieval's *recall errors* impact label accuracy most severely.

Evidence sentences are lost variously throughout steps 1-3. In particular: (1) Exact title-matching misses many evidence-containing articles, mostly when titles are not directly mentioned in a claim (e.g. article *"Peru"* supports the claim *"Chile is in Asia"*). However, we retain at least one evidence-containing article (e.g. *"Chile"*) in all but 147 cases. (2) Article selection filters out all remaining evidence for 420 claims. Our model is over-sensitive to whether the title is at the beginning of the claim (e.g. accepting the article *"Daag (1973 film)"* for *"Daag is a film"* but not *"a Daag is a film"*). Moreover, token overlap features disadvantage articles with refuting evidence (e.g. *"Daag is a painting"* barely overlaps with the refuting article). (3) Sentence selection rejects *all* remaining evidence sentences for a further 588 claims and *some* remaining sentences for 319 other claims. In many cases, the rejected sentences just don't have enough overlap with the claim text (they evidence the claim only partially or indirectly). Often rejected refuting sentences differ from the claim in ways not captured by our ontological features, especially when the contradictory terms are *names* (e.g. sentence *"Angelsberg [is in] Luxembourg"* v.s claim *"Angelsberg is in Canada"*).

5.2 Claim Labelling

The high confusion in the NEI column highlights the trouble our *claim labelling* models have realising that the given evidence does not contain enough information to make a judgement, resulting in significant mislabellings for NEI claims.

Further isolated oracle tests of the *claim-evidence assessment* module reveal that the ESIM+ELMo pre-trained model misclassifies 63% of NEI claims, suggesting that the pre-trained model is the root cause of the problem. This is likely due to the lack of custom training, compounded with the strong assumption that SNLI's *Entails, Contradicts* and *Neutral* labels map to SUP, REF and NEI; e.g. the NEI claim *"Yandex operates in Luxembourg"* is labelled as REF given the evidence *"Yandex operates the largest search engine in Russia"* because it would be a contradiction under SNLI evidence standards.

6 Conclusions and Future Work

We designed and evaluated a 5-step fact-verification system, with class-competitive results. Nevertheless, Section 5 highlights clear shortcomings in each step.

Directions for improvement include: (1) Indexing articles beyond titles to find evidence-containing articles not mentioned in claims; (2) Enriching our representation of claims and sentences to capture the kind of indirect semantic relatedness that constitutes evidence, for *or* against a claim (a more nuanced relation than IR's traditional notion of "similarity") —We could improve on our token-level ontological relationship extraction e.g. by using full sentence embeddings or parsing-based structural embeddings; (3) Training the TE model on the significant FEVER dataset, sensitising it to task-specific standards of evidence and the nature of Wikipedia as a corpus, and possibly extending the model to explicitly cater to the small proportion of FEVER claims requiring composition of multiple evidence sentences (which we have not begun to properly address); and (4) Exploiting Wikipedia's networked structure for both entity disambiguation and evidence "expansion" (i.e. by augmenting retrieved sentences with additional information about each entity mentioned, or "linked to", in the sentence)—While this structure may not exist beyond Wikipedia, recent advances in Entity Linking may allow us to infer structure over other corpora (Raiman and Raiman, 2018), enabling fact verification in a more general setting.

References

AllenNLP. 2018. AllenNLP Models ESIM.

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, Lisbon, Portugal, September. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. *arXiv:1609.06038 [cs]*, September. arXiv: 1609.06038.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium, November. Association for Computational Linguistics.

Christopher Malon. 2019. Team Papelo: Transformer Networks at FEVER. *arXiv:1901.02534 [cs]*, January. arXiv: 1901.02534.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2018. Combining Fact Extraction and Verification with Neural Semantic Matching Networks. *arXiv:1811.07039 [cs]*, November. arXiv: 1811.07039.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. *arXiv:1606.01933 [cs]*, June. arXiv: 1606.01933.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv:1802.05365 [cs]*, February. arXiv: 1802.05365.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for Fact Extraction and VERification. *arXiv:1803.05355 [cs]*, March. arXiv: 1803.05355.

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, November. Association for Computational Linguistics.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102, Brussels, Belgium, November. Association for Computational Linguistics.

Supplemental Material

A.1 Related Works

The closest related works are the systems created for the 2018 FEVER shared task competition (Thorne et al., 2018b). Our task differs from the FEVER task in the conditions used to evaluate the supplied evidence: We use a relaxed criterion where, in the small proportion of cases where multiple sentences are required to form a judgement, each of these sentences is considered sufficient evidence (the FEVER task requires that a "complete" set of evidence sentences be retrieved). Nevertheless, we expect the knowledge and techniques developed for either task to be broadly transferable.

The FEVER task was originally framed as comprising three steps: (1) **Document selection**: Retrieve Wikipedia articles that contain relevant information for assessing a claim; (2) **Sentence selection**: Extract from these articles any sentences necessary for making a judgement; and (3) **Natural language inference (NLI)**: Determine whether the evidence supports or refutes the claim, or that there is not enough information in the corpus to decide.

In this section, we summarize the most interesting patterns from the shared task participants, following the task breakdown proposed by the FEVER authors.

A.1.1 Document Selection

While the FEVER baseline cast article selection as a traditional Information Retrieval (IR) problem (treating the claim as a "query" and using TF-IDF-based similarity metrics to find "relevant" articles) we, and many shared task entrants, observed that the task is better framed as end-to-end Named Entity Recognition and Disambiguation, also known as Entity Linking (EL) or, aptly, "Wikification" (many EL datasets are derived from Wikipedia). State-of-the-art EL systems use deep learning backed by an inferred entity type system (Raiman and Raiman, 2018).

In contrast, many shared task entrants used heuristic approaches scanning claims for entity mentions using, variously, Named Entity Recognition (e.g Nie et al., 2018), Constituency Parsing for extracting Noun Phrases (e.g Hanselowski et al., 2018), capitalized expression detection (e.g. Malon, 2019), or simple substring matching (e.g. Yoneda et al., 2018), retrieving Wikipedia articles whose titles match the mentioned entities. This simple approach to EL exploits the standardised structure of Wikipedia articles and their titles and performs accurately on much of the FEVER data. We adopt a similar approach.

A.1.2 Sentence Selection

The shared task saw two prominent approaches to sentence selection. Some participants preserved the separation between the sentence selection and NLI steps from the baseline (e.g. Yoneda et al., 2018; Hanselowski et al., 2018). These systems successfully employed claim-sentence similarity-based ranking or supervised classification approaches.

Other entrants (e.g. Nie et al., 2018) merged sentence selection and NLI into a single step, typically using a neural network. These coupled systems incorporated features related to the semantic overlap between sentence and claim, such as by using word embeddings or ontological relationships between words.

We adopt the former approach (in particular, supervised classification), but enrich our classifier with additional semantic and ontological features.

A.1.3 Natural Language Inference (NLI)

Following the baseline, all participants framed the NLI task as a supervised classification task similar to the Stanford Natural Language Inference task (SNLI) (Bowman et al., 2015).

There are some important differences between the FEVER data and SNLI data. In particular, the SNLI examples contain shorter sentences and use a simpler vocabulary than the FEVER examples (Malon, 2019). In response, many participants employed successful models built for SNLI after re-training them on FEVER data. Some popular pre-trained models were the Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017), and Decomposable Attention (DA) (Parikh et al., 2016).

16.82% of FEVER claims require more than one sentence to form appropriate evidence for deducing the appropriate claim label. More generally, the document and sentence selection steps may identify more than one sentence in relation to a claim. A key challenge in porting SNLI models to the FEVER setting is thus the handling of multi-sentence premises. The two main approaches were (1) pre-concatenation of all identified evidence to form a single premise, and (2) individual classification of all claim-evidence pairs followed by some form of label aggregation. Aggregation strategies ranged from rule-based techniques (Malon, 2019) to supervised classification including with attention-based neural models (Yoneda et al., 2018; Hanselowski et al., 2018).