UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification

Andreas Hanselowski[†*]**, Hao Zhang**[*]**, Zile Li**[*]**, Daniil Sorokin**[†*]**,**
Benjamin Schiller[*]**, Claudia Schulz**[†*]**, Iryna Gurevych**[†*]

[†]Research Training Group AIPHES
Computer Science Department, Technische Universität Darmstadt
https://www.aiphes.tu-darmstadt.de

[*]Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Computer Science Department, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de/

Abstract

The Fact Extraction and VERification (FEVER) shared task was launched to support the development of systems able to verify claims by extracting supporting or refuting facts from raw text. The shared task organizers provide a large-scale dataset for the consecutive steps involved in claim verification, in particular, document retrieval, fact extraction, and claim classification. In this paper, we present our claim verification pipeline approach, which, according to the preliminary results, scored third in the shared task, out of 23 competing systems. For the document retrieval, we implemented a new entity linking approach. In order to be able to rank candidate facts and classify a claim on the basis of several selected facts, we introduce two extensions to the Enhanced LSTM (ESIM).

1 Introduction

In the past years, the amount of false or misleading content on the Internet has significantly increased. As a result, information evaluation in terms of fact-checking has become increasingly important as it allows to verify controversial claims stated on the web. However, due to the large number of fake news and hyperpartisan articles published online every day, manual fact-checking is no longer feasible. Thus, researchers as well as corporations are exploring different techniques to automate the fact-checking process[1].

In order to advance research in this direction, the Fact Extraction and VERification (FEVER) shared task[2] was launched. The organizers of the FEVER shared task constructed a large-scale dataset (Thorne et al., 2018) based on Wikipedia. This dataset contains 185,445 claims, 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. Supported by the structure of the dataset, the FEVER shared task encompasses three sub-tasks that need to be solved.

Document retrieval: Given a claim, find (English) 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 one of three labels for the claim: *Supported*, *Refuted*, or *NotEnoughInfo*. To evaluate the performance of the competing systems, an evaluation metric was devised by the FEVER organizers: a claim is considered as correctly verified if, in addition to predicting the correct label, a correct evidence set was retrieved.

In this paper, we describe the pipeline system that we developed to address the FEVER task. For document retrieval, we implemented an entity linking approach based on constituency parsing and handcrafted rules. For sentence selection, we developed a sentence ranking model based on the *Enhanced Sequential Inference Model* (ESIM) (Chen et al., 2016). We furthermore extended the ESIM for recognizing textual entailment between multiple input sentences and the claim using an attention mechanism.

According to the preliminary results of the

FEVER shared task, our systems came third out of 23 competing teams.

2 Background

In this section, we present underlying methods that we adopted for the development of our system.

2.1 Entity linking

The document retrieval step requires matching a given claim with the content of a Wikipedia article. A claim frequently features one or multiple *entities* that form the main content of the claim.

Furthermore, Wikipedia can be viewed as a knowledge base, where each article describes a particular *entity*, denoted by the article title. Thus, the document retrieval step can be framed as an entity linking problem (Cucerzan, 2007). That is, identifying entity mentions in the claim and linking them to the Wikipedia articles of this entity. The linked Wikipedia articles can then be used as the set of the retrieved documents for the subsequent steps.

2.2 Enhanced Sequential Inference Model

Originally developed for the SNLI task (Bowman et al., 2015) of determining entailment between two statements, the ESIM (Enhanced Sequential Inference Model) (Chen et al., 2016) creates a rich representation of statement-pairs. Since the FEVER task requires the handling of claim-sentence pairs, we use the ESIM as the basis for both sentence selection and textual entailment. The ESIM solves the entailment problem in three consecutive steps, taking two statements as input.

Input encoding: Using a bidirectional LSTM (BiLSTM) (Graves and Schmidhuber, 2005), representations of the individual tokens of the two input statements are computed.

Local inference modeling: Each token of one statement is used to compute attention weights with respect to each token in the other statement, giving rise to an attention weight matrix. Then, each token representation is multiplied by all of its attention weights and weighted pooling is applied to compute a single representation for each token. This operation gives rise to two new representations of the two statements.

Inference composition: These two statement representations are then fed into two BiLSTMs, which again compute sequences of representations for each statement. Maximum and average pool-

ing is applied to the two sequences to derive two representations, which are then concatenated (last hidden state of the ESIM) and fed into a multilayer perceptron for the classification of the entailment relation.

3 Our system for fact extraction and claim verification

In this section, we describe the models that we developed for the three FEVER sub-tasks.

3.1 Document retrieval

As explained in Section 2.1, we propose an *entity linking* approach to the document retrieval sub-task. That is, we find entities in the claims that match the titles of Wikipedia articles (documents). Following the typical entity linking pipeline, we develop a document retrieval component that has three main steps.

Mention extraction: Named entity recognition tools focus only on the main types of entities (Location, Organization, Person). In order to find entities of different categories, such as movie titles, that are numerous in the shared task data set, we employ the constituency parser from AllenNLP (Gardner et al., 2017). After parsing the claim, we consider every *noun phrase* as a potential entity mention. However, a movie or a song title may be an adjective or any other type of syntactic phrase. To account for such cases, we use a heuristic that adds all words in the claim before the main verb as well as the whole claim itself as potential entity mentions. For example, a claim "*Down With Love is a 2003 comedy film.*" contains the noun phrases '*a 2003 comedy film*' and '*Love*'. Neither of the noun phrases constitutes an entity mention, but the tokens before the main verb, '*Down With Love*', form an entity.

Candidate article search: We use the MediaWiki API[3] to search through the titles of all Wikipedia articles for matches with the potential entity mentions found in the claim. The MediaWiki API uses the Wikipedia search engine to find matching articles. The top match is the article whose title has the largest overlap with the query. For each entity mention, we store the seven highest-ranked Wikipedia article matches.

The MediaWiki API uses the online version of Wikipedia and since there are some discrepancies

---

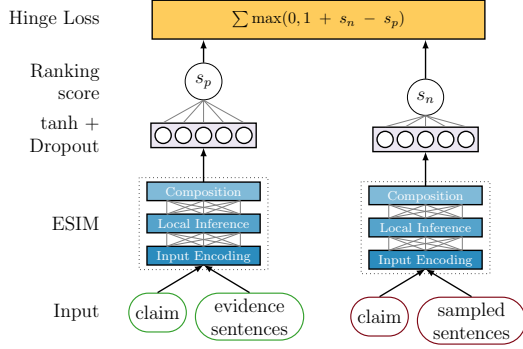[3] https://www.mediawiki.org/wiki/API:Main_page

Figure 1: Sentence selection model

between the 2017 dump used in the shared task and the latest version, we also perform an exact search over all Wikipedia article titles in the dump. We add these results to the set of the retrieved articles.

Candidate filtering: The MediaWiki API retrieves articles whose title overlaps with the query. Thus, the results may contain articles with a title longer or shorter than the entity mention used in the query. Similarly to previous work on entity linking (Sorokin and Gurevych, 2018), we remove results that are longer than the entity mention and do not overlap with the rest of the claim. To check this overlap, we first remove the content in parentheses from the Wikipedia article titles and stem the remaining words in the titles and the claim. Then, we discard a Wikipedia article if its stemmed article title is not completely included in the stemmed claim.

We collect all retrieved Wikipedia articles for all identified entity mentions in the claim after filtering and supply them to the next step in the pipeline. The evaluation of the document retrieval system on the development data shows the effectiveness of our ad-hoc entity linking approach (see Section 4).

3.2 Sentence selection

In this step, we select candidate sentences as a potential evidence set for a claim from the Wikipedia articles retrieved in the previous step. This is achieved by extending the ESIM to generate a ranking score on the basis of two input statements, instead of predicting the entailment relation between these two statements.

Architecture: The modified ESIM takes as input a claim and a sentence. To generate the ranking score, the last hidden state of the ESIM (see Section 2.2) is fed into a hidden layer which is con-

nected to a single neuron for the prediction of the ranking score. As a result, we are able to rank all sentences of the retrieved documents according to the computed ranking scores. In order to find a potential evidence set, we select the five highest-ranked sentences.

Training: Our adaptation of the ESIM is illustrated in Fig. 1. In the training mode, the ESIM takes as input a claim and the concatenated sentences of an evidence set. As a loss function, we use a modified hinge loss with negative sampling: $\sum max(0, 1 + s_n - s_p)$, where s_p indicates the positive ranking score and s_n the negative ranking score for a given claim-sentence pair. To get s_p, we feed the network a claim and the concatenated sentences of one of its ground truth evidence sets. To get s_n, we take all Wikipedia articles from which the ground truth evidence sets of the claim originate, randomly sample five sentences (not including the sentences of the ground truth evidence sets for the claim), and feed the concatenation of these sentences into the same ESIM. With our modified hinge loss function, we then try to maximize the margin between positive and negative samples.

Testing: At testing time, we calculate the score between a claim and each sentence in the retrieved documents. For this purpose, we deploy an ensemble of ten models with different random seeds. Then, the mean score of a claim-sentence pair over all ten models of the ensemble is calculated and the scores for all pairs are ranked. Finally, the five sentences of the five highest-ranked pairs are taken as an output of the model.

3.3 Recognizing textual entailment

In order to classify the claim as $Supported$, $Refuted$ or $NotEnoughInfo$, we use the five sentences retrieved by our sentence selection model described in the previous section. For the classification, we propose another extension to the ESIM, which can predict the entailment relation between *multiple* input sentences and the claim. Fig. 2 gives an overview of our extended ESIM for the FEVER textual entailment task.

As word representation for both claim and sentences, we concatenate the Glove (Pennington et al., 2014) and FastText (Bojanowski et al., 2016) embeddings for each word. Since both types of embeddings are pretrained on Wikipedia, they are particularly suitable for our problem setting.
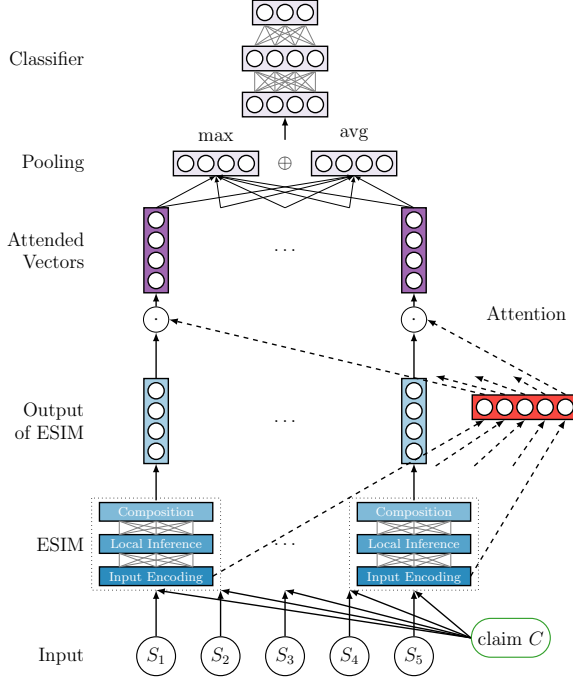
105

Figure 2: Extended ESIM for recognizing textual entailment

To process the five input sentences using the ESIM, we combine the claim with each sentence and feed the resulting pairs into the ESIM. The last hidden states of the five individual claim-sentence runs of the ESIM are compressed into one vector using attention and pooling operations.

The attention is based on a representation of the claim that is independent of the five sentences. This representation is obtained by summing up the *input encodings* of the claim in the five ESIM runs. In the same way, we derive five sentence representations, one from each of the five runs of the ESIM, which are independent of the claim. For each claim-sentence pair, the single sentence representation and the claim representation are then individually fed through a single layer perceptron. The cosine similarity of these two vectors is then used as an attention weight. The five output vectors of all ESIMs are multiplied with their respective attention weights and we apply average and max pooling on these vectors in order to reduce them to two representations. Finally, the two representations are concatenated and fed through a 3-layer perceptron to predict one of the three classes *Supported*, *Refuted* or *NotEnoughInfo*. The idea behind the described attention mechanism is to allow the model to extract information from the five sentences that is most relevant for the classification of the claim.

4 Results

Table 1 shows the performance of our document retrieval and sentence selection system when retrieving different numbers of the highest-ranked Wikipedia articles. In contrast to the results reported in Table 3, here we consider a single model instead of an ensemble. The results show that both systems benefit from a larger number of retrieved articles.

#search results	doc. accuracy	sent. recall
3	92.60	85.37
5	93.30	86.02
7	**93.55**	**86.24**

Table 1: Performance of the retrieval systems using different numbers of MediaWiki search results

For the subtask of recognizing textual entailment, we also experiment with different numbers of selected sentences. The results in Table 2 demonstrate that our model performs best with all five selected sentences.

#sentence(s)	label accuracy	FEVER score
1	67.94	63.64
2	68.33	64.30
3	67.82	63.72
4	67.61	63.59
5	**68.49**	**64.74**

Table 2: Performance of the textual entailment model using different numbers of sentences

In Table 3, we compare the performance of our three systems as well as the full pipeline to the baseline systems and pipeline implemented by the shared task organizers (Thorne et al., 2018) on the development set. As the results demonstrate, we were able to significantly improve upon the baseline on each sub-task. The performance gains over the whole pipeline add up to an improvement of about 100% with respect to the baseline pipeline.

5 Error analysis

In this section, we present the error analysis for each of the three sub-tasks, which can serve as a basis for further improvements of the system.

5.1 Document retrieval

The typical errors encountered for the document retrieval system can be divided into three classes.

Task (metric)	system	score (%)
Document retrieval (accuracy)	baseline	70.20
	our system	93.55
Sentence selection (recall)	baseline	44.22
	our system	87.10
Textual entailment (label accuracy)	baseline	52.09
	our system	68.49
Full pipeline (FEVER score)	baseline	**32.27**
	our system	**64.74**

Table 3: Performance comparison of our system and the baseline system on the development set

Spelling errors: A word in the claim or in the article title is misspelled. E.g. *"Homer Hickman wrote some historical fiction novels."* vs. *"Homer Hickam"*. In this case, our document retrieval system discards the article during the filtering phase.

Missing entity mentions: The entity mention represented by the title of the article, which needs to be retrieved, is not related to any entity mention in the claim. E.g. Article title: *"Blue Jasmine"* Claim: *"Cate Blanchett ignored the offer to act in Cate Blanchett."*.

Search failures: Some article titles contain a category name in parentheses for the disambiguation of the entity. This makes it difficult to retrieve the exact article title using the MediaWiki API. E.g. the claim *"Alex Jones is apolitical."* requires the article *"Alex Jones (radio host)"*, but it is not contained in the MediaWiki search results.

5.2 Sentence selection

The most frequent case, in which the sentence selection model fails to retrieve a correct evidence set, is that the entity mention in the claim does not occur in the annotated evidence set. E.g. the only evidence set for the claim *"Daggering is nontraditional."* consists of the single sentence *"This dance is not a traditional dance."*. Here, *"this dance"* refers to *"daggering"* and cannot be resolved by our model, since the information that *"daggering"* is a dance is not mentioned in the evidence sentence or in the claim. Some evidence sets contain two sentences one of which is less related to the claim. E.g. the claim *"Herry II of France has three cars."* has an evidence set that contains the two sentences *"Henry II died in 1559."* and *"1886 is regarded as the birth year of the modern car."*. The second sentence is not directly related to the

claim, thus, it is ranked very low by our model.

5.3 Recognizing textual entailment

A large number of claims are misclassified due to the model's disability to interpret numerical values. For instance, the claim *"The heart beats at a resting rate close to 22 beats per minute."* is not classified as *refuted* on the basis of the evidence sentence *"The heart beats at a resting rate close to 72 beats per minute."*. The only information refuting the claim is the number, but neither GloVe nor FastText embeddings can embed numbers distinctly enough. Another problem are challenging *NotEnoughInfo* cases. For instance, the claim *"Terry Crews played on the Los Angeles Chargers."* (annotated as *NotEnoughInfo*) is classified as *refuted*, given the sentence *"In football, Crews played ... for the Los Angeles Rams, San Diego Chargers and Washington Redskins, ..."*. The sentence is related to the claim but does not exclude it, which makes this case difficult.

6 Conclusion

In this paper, we presented the system for fact extraction and verification, which we developed in the course of the FEVER shared task. According to the preliminary results, our system scored third out of 23 competing teams. The shared task was divided into three parts: (i) Given a claim, retrieve Wikipedia documents that contain facts about the claim. (ii) Extract these facts from the document. (iii) Verify the claim on the basis of the extracted facts. To address the problem, we developed models for the three sub-tasks. We framed document retrieval as entity linking by identifying entities in the claim and linking them to Wikipedia articles. To extract facts in the articles, we developed a sentence ranking model by extending the ESIM. For claim verification we proposed another extension to the ESIM, whereby we were able to classify the claim on the basis of multiple facts using attention. Each of our three models, as well as the combined pipeline, significantly outperforms the baseline on the development set.

7 Acknowledgements

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced LSTM for natural language inference. *arXiv preprint arXiv:1609.06038*.

Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Daniil Sorokin and Iryna Gurevych. 2018. Mixing Context Granularities for Improved Entity Linking on Question Answering Data across Entity Categories. In *Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM 2018)*, pages 65–75. Association for Computational Linguistics.

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