Strong and Light Baseline Models for Fact-Checking Joint Inference

Kateryna Tymoshenko
DISI, University of Trento
38123 Povo (TN), Italy
kateryna.tymoshenko@unitn.it

Alessandro Moschitti*
Amazon
Manhattan Beach, CA 90266, USA
amosch@amazon.com

Abstract

How to combine several pieces of evidence to verify a claim is an interesting semantic task. Very complex methods have been proposed, combining different evidence vectors using an evidence interaction graph. In this paper, we show that in case of inference based on transformer models, two effective approaches use either (i) a simple application of max pooling over the Transformer evidence vectors; or (ii) computing a weighted sum of the evidence vectors. Our experiments on the FEVER claim verification task show that the methods above achieve the state of the art, constituting strong baseline for much more computationally complex methods.

1 Introduction

Automatic Fact Checking is quickly gaining attention of the NLP and AI communities. The FEVER.ai Fact Extraction and Verification Shared Task (Thorne et al., 2018) provides a benchmark for evaluating fact-checking systems. In FEVER, given a claim, C, and a collection of approximately five million Wikipedia pages, W, the task is to predict whether C is *supported* (SUP) or *refuted* (REF) by W, or whether there is *not enough information* (NEI) in W to support or refute C. If C is classified as SUP or REF, the respective evidence should be provided. Tab. 1 shows a FEVER claim and the gold-standard evidence refuting it.

The overall task is complex, as one needs to retrieve the documents that contain the evidence (*document retrieval*, *DocIR*), select relevant evidence (*evidence selection*, *ES*) and label the claim given the evidence (*evidence reasoning*, *ER*), which is the focus of our work. Formally, given a claim, C, and a list top K evidence sentences, $(E_1, ..., E_K)$, retrieved with *DocIR* and selected by *ES* respectively, the *ER* task is to predict the claim label (SUP/REF/NEI).

Claim	Evidence
Coeliac disease is not treated by maintaining a gluten-free diet. (REF)	[(Coeliac_disease, "The only known effective treatment is a strict lifelong gluten-free diet....")]

Table 1: FEVER data examples

There can be multiple inter-dependent evidence sentences per claim, thus joint modeling them allows for taking multiple clues into account, thus intuitively improving system accuracy. Indeed, individual sentences may not constitute standalone evidence, but they can contain several clues, which, together, can support or refute the claim. For example, Sentence 8 of the Gluten-free_diet Wikipedia page, *"..gluten-free diet is demonstrated as an effective treatment, but several studies show that about 79 % ... an incomplete recovery of the small bowel..."*, which is not listed as ground truth evidence for the claim, still supports the REF signal.

Given the above intuition, recent state-of-the-art (SOTA) approaches (Zhou et al., 2019; Ye et al., 2020; Liu et al., 2020; Zhong et al., 2020; Zhao et al., 2020) combine different pieces of evidence with graph networks, also increasing computational and space complexity. In this paper, we show that simple joint transformer-based methods achieve better performance than the best complex systems. Specifically, we (i) text-*concatenate* evidence sentences, (ii) apply *max pooling* to their individual embedding representation, or (iii) compute their *weighted sum*. Since June 1^{st} 2021, our baseline is sixth in terms of Label Accuracy (LA) and seventh in terms of FEVER score on the official task leaderboard[1], where the absolute difference from the fourth top LA is 0.2%.

We believe our results are important to enable

*Professor at the University of Trento.

[1] https://competitions.codalab.org/competitions/18814#results

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4824–4830

researchers to select the right scientific challenge, providing the appropriate baselines. For example, proposing complex models that are less accurate than our baselines can most likely mislead the research community, thus knowing our baselines can help to lead research in this area in the right directions. Additionally, our baselines are strong, simple to use, and easily reproducible, enabling fast comparison with innovative inference models.

2 Related work

SOTA approaches. Most recent approaches encode claim and evidence texts using Transformer-based language representation models (LRM), such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and others. **GEAR** (Zhou et al., 2019) and **KGAT** (Liu et al., 2020) construct graphs with evidences as nodes and use deep graph neural ER networks to propagate knowledge; **DREAM** (Zhong et al., 2020) reasons on a graph built on top of a semantic role labeler output; **Transformer-XH** (Zhao et al., 2020) propagates knowledge between `[CLS]` tokens of different evidence pieces; **CorefBERT** (Ye et al., 2020) trains a BERT-based LRM, which employs an additional objective modeling coreference knowledge, and use it within KGAT architecture. The winners of the original FEVER competition (Nie et al., 2019) used older LRMs and a modified enhanced sequential inference model (ESIM) (Chen et al., 2017) to do ER.

The top-scoring *published* approach, **DOMLIN++** (Stammbach and Ash, 2020), simply text-concatenates evidence pieces and uses a RoBERTa-based classifier, thus supporting our thesis that simple models can be very effective. On the other hand, they use additional DocIR components and data (MultiNLI (Williams et al., 2018) corpus) for fine-tuning. To the best of our knowledge, their code/output are not available online yet[2], so we cannot compare to them directly at the moment.

Baselines. The baselines in the above works, apart from the previous SOTA systems, consist in applying a transformer-based classifier to (i) *concatenation* of C and all E_i, $i = 1..K$ (Zhou et al., 2019; Zhong et al., 2020; Zhao et al., 2020); or (ii) *separate* (C, E_i) pairs, $i = 1..K$, and aggregating the results heuristically (Zhou et al., 2019). The latter also considered *max-pooling* and *weighted-sum* baselines, but used them only on subsets of the development set with multiple gold evidence pieces per claim. In this work, we use them in the full-scale setting.

3 Strong baseline models

BERT for classification. BERT LRM and its version with the improved training procedure, RoBERTa, have obtained outstanding results on a number of NLP tasks. When using BERT-based architectures for classification, a special `[CLS]`[3] token is prepended to an input text sequence. Its embedding from the last layer of the transformer, $h^{[CLS]} \in \mathbb{R}^{h_{dim}}$ is a vector representation of the sequence. h_{dim} is the hidden dimension size. The final prediction is $p = softmax(L)$, where $L = Wh^{[CLS]} \in \mathbb{R}^N$, $W \in \mathbb{R}^{N \times h_{dim}}$, and N is the number of classes[4].

Baseline approaches. We investigate four simple Transformer-based baseline approaches: **Local**, **Concat**, **MaxPool**, **WgtSum**.

The input to the task are a claim, C, and a list of top K evidence sentences selected by an ES component, $E = \{E_i\}$, $i = 1, .., K$. Tab. 2 describes the input format. Following Liu et al. (2020), we incorporate E_i source page name into the input. We use cross-entropy loss to train all the models.

Local: for each E_i, we (i) use the standard 3-way classification Transformer-based model, T_{class}, to get an evidence-level label prediction, P_i, along with its corresponding $l_i = max(L_i)$ score, where $L_i \in \mathbb{R}^N$ is the logits vector produced by T_{class} for E_i; (ii) sort the predictions list, $P = [(P_i, l_i)]$, on l in the reverse order; (iii) create P', a sublist of P, where P_i is not `NEI` and $l_i > 0$. If P' is not empty, P'_1 is the claim label, otherwise it is `NEI`. We introduce P', because we want to capture the `SUP/REF` signal even if it is weaker compared to that of `NEI`.

Concat: T_{class} run on the input described in Tab. 2.

Local and **Concat** are similar to the Bert-pair and Bert-Concat baselines, respectively introduced in (Zhou et al., 2019).

MaxPool: encodes each (C, E_i) pair with a transformer model, concatenates the resulting $h_i^{[CLS]}$ into a matrix $H^{[CLS]} \in \mathbb{R}^{h_{dim} \times K}$ and max-pools it, column-wise, into $h_{mp}^{[CLS]} \in \mathbb{R}^{h_{dim}}$. The output

[2] We could try to re-implement their pipeline following the high-level descriptions in their paper, however, our re-implementation still will not be able to re-produce their ER input due to the inevitable implementation differences

[3] This is standard for BERT, other language models can use a different token in a different position

[4] This strategy is employed by BERT. Practical implementations of the other models can also apply more transformations to $h^{[CLS]}$ to obtain L.

Baseline	Input type example
Local, **MaxPool**, **WgtSum**	[CLS] C [SEP] E_i^{page} <psep> E_i [SEP]
Concat	[CLS] C [SEP] E_1^{page} <psep> E_1 [SEP] ... [SEP] E_K^{page} <psep> E_K [SEP]

Table 2: Input data for the baselines. E_i^{page} is the name of the E_i source Wikipedia page. [SEP] and [CLS] are the standard "separator" and "classification" tokens used in BERT-like models. <psep> is delimiter separating page name from the evidence text. We use ". ", while Liu et al. (2020) use [SEP]

is $p = softmax(W_{mp}h_{mp}^{[CLS]})$, $W_{mp} \in \mathbb{R}^{3\times h_{dim}}$. It is inspired by the max pooling evidence aggregation procedure employed by (Hanselowski et al., 2018; Zhou et al., 2019).

WgtSum: encodes each (C, E_i) pair with a transformer, computes the weighted sum $h_{ws}^{[CLS]} = \sum_{i=1}^{K} \alpha_i h_i^{[CLS]} \in \mathbb{R}^{h_{dim}}$, $\alpha_i = softmax_i(W_{ws}h_i^{[CLS]})$, $W_{ws} \in \mathbb{R}^{1\times h_{dim}}$. The weight α_i is intended to reflect the relative importance of E_i. The output is $p = softmax(Wh_{ws}^{[CLS]})$, $W \in \mathbb{R}^{3\times h_{dim}}$. **WgtSum** is similar to the Zhou et al. (2019)'s attention baseline in the sense that we aggregate pieces of evidence representations via a weighted summation. However, differently from us, they obtain the weights by computing attention between the claim and the evidence hidden states. We refer to **Concat**, **MaxPool** and **WgtSum** as *global* systems.

4 Experiments

Implementation. Our system is an AllenNLP pipeline (Gardner et al., 2017). Our code is available at https://github.com/iKernels/reasoning-baselines. We use the pre-trained BERT and RoBERTa LRMs from the *transformers*[5] library, namely *bert-base-cased*, *roberta-base* and *roberta-large*.

Training setup. We train for three epochs, with an evaluation checkpoint every 500 and 2500 training steps for global and local models correspondingly, thus having 14 checkpoints in total. We use $K = 5$ evidence pieces per claim. For all the models the batch size/number of gradient accumulation steps are 8/8 and 2/32 with base and large LRMs, respectively. We use Adam optimizer with slanted triangular learning rate (Howard and Ruder, 2018), $cut\_frac = 0.1$, ratio of 32[6].

When experimenting with *roberta-base* we tried learning rates [1e-5; 5e-5] with the step of 1e-5 and observed no noticeable difference between the rates in the range [2e-5; 5e-5]. Additional details are available in the appendix.

[5] https://github.com/huggingface/transformers

[6] Standard values suggested in (Howard and Ruder, 2018)

Split	#SUP	#REF	#NEI
TRAIN	80,035	29,775	35,639
DEV	6,666	6,666	6,666
TEST	6,666	6,666	6,666

Table 3: FEVER dataset statistics. # denotes the number of claims in a given class.

FEVER metrics. The primary shared task metric is FEVER[7]. It takes the correctness of the evidence set provided with the claim label[8] into account. The evidence set must contain all sentences belonging to at least one evidence[9] associated with a claim. No evidence is needed for NEI claims. *Label Accuracy (LA)* is another standard metric. *Oracle FEVER (OFEVER)* is the FEVER metric computed using oracle downstream components after DocIR or ES, i.e., it estimates downstream component's upper bound performance.

4.1 The dataset

We conduct our experiments on the official FEVER 1.0 Shared Task dataset[10]. Tab. 3 reports the FEVER 1.0 statistics. Verifiable (SUP or REF) claims are associated with at least one evidence. 35.23% of verifiable claims in DEV are associated with multiple evidence sentences, independent or inter-dependent.

Evidence Reasoning (ER) dataset. We run the ER experiments on the evidence sentences retrieved by Liu et al. (2020), published on their github[11] with ES OFEVER score of 96.25. Their *DocIR* module retrieves documents for a given claim via entity linking following (Hanselowski et al., 2018), and the ES module selects relevant evidence (*ES*) via BERT-based system with pairwise loss.

Following (Liu et al., 2020), when training and selecting the best checkpoint we use gold evidence completed with the non-gold evidence pieces retrieved by ES, so that the total amount of evidence pieces per claim is K. When evaluating on DEV we simply use top 5 evidence pieces retrieved by ES, i.e. the results in Sec. 4.2 are obtained on the

[7] Scorer: https://github.com/sheffieldnlp/fever-scorer

[8] At least one of top 5 predicted evidences must be correct.

[9] An evidence consists of one or more sentences. One claim can have multiple evidences.

[10] FEVER 1.0 Shared Task at https://fever.ai/resources.html

[11] https://github.com/thunlp/KernelGAT/tree/master/data

		FEVER	LA	LRM
	KGAT SOTA baseline, lr=5e-05			
1:	KGAT$_{pub}$	75.88	78.02	bert-base-cased
2:	(Liu et al., 2020)	76.11	78.29	roberta-large
	Reproducing KGAT results, lr=5e-05			
3:	KGAT	75.64	77.80	bert-base-cased
4:		77.21	79.52	roberta-base
	Other learning rates (lr) for KGAT			
5:	KGAT, lr=2e-5	74.87	77.15	bert-base-cased
6:		77.66	79.98	roberta-base
7:		78.66	80.77	roberta-large
8:	KGAT, lr=3e-5	75.28	77.48	bert-base-cased
9:		77.75	80.06	roberta-base
	Local models, lr=2e-05			
10:	Aggr. heuristic 1	73.05	75.11	bert-base-cased
11:		75.62	77.85	roberta-base
12:	Aggr. heuristic 2	71.79	73.66	bert-base-cased
13:		73.98	75.96	roberta-base
	Global baselines, lr=2e-05			
14:	Concat	74.23	76.51	bert-base-cased
15:		77.09	79.25	roberta-base
16:		78.27	80.31	roberta-large
17:	MaxPool	74.72	76.99	bert-base-cased
18:		77.48	79.82	roberta-base
19:		78.85	81.16	roberta-large
20:	WgtSum	74.48	76.85	bert-base-cased
21:		77.62	80.01	roberta-base
22:		**79.02**	**81.30**	roberta-large

Table 4: Results on the official DEV set. $_{pub}$ is the result officially published in the reference paper; **lr** is learning rate. *Aggr.* is a shorthand for *Aggregation*.

real-life gold standard-agnostic output of the *DocIR* and *ES* modules.

Note that by construction, we generate one train/test instance per each (C, E_i) pair when training/testing **Local** models and then aggregate the labels predicted for different evidence pieces, i.e the total amount of instances is around number of claims times K[12]. For example, we train **Local** on 722K examples, split into 548K `NEI`, 127K `SUP`, 48K `REF`. When training/testing **Concat**, **WgtSum** and **MaxPool**, we generate one instance per each claim.

4.2 Results

Tab. 4 reports the performance of the systems described in Sec. 3 on the official DEV set.

In previous work, systems employing KGAT ER architecture (Liu et al., 2020; Ye et al., 2020) achieve top performance in terms of FEVER. KGAT ER input data are publicly available enabling us to conduct fair comparison. Lines 1 and 2 report KGAT performance as in (Liu et al., 2020). We integrated their implementation of the KGAT ER component into our pipeline and obtained performance numbers comparable to those published (lines 3, 1). Interestingly, our runs with *roberta-base* outperform the published results of KGAT runs with *roberta-large* (lines 4, 6, 9). We also include its best result (that is an upperbound of KGAT) with *roberta-base* that we obtained with the learning rate of 3e-5. KGAT with *roberta-large* and learning rate of 2e-5 further pushes the performance 1 point up, while the training with the learning rates of 3e-5 and 5e-5 did not converge.

Local models. Lines 10-13 report performance of our local models with two evidence label aggregation heuristics. Heuristic 1 consists in applying the procedure described in Sec. 3 to the labels assigned to all evidence pieces by **Local**. Heuristic 2 is to simply pick the label assigned to the evidence sentence *top-ranked* by ES as in (Zhou et al., 2019). The aggregation heuristic 1 is more competitive.

Global models. Lines 14-22 report performance of the **Concat**, **WgtSum**, **MaxPool** global systems, which all clearly outperform **Local**. Note, that in the **Concat** setting C and E_i, $i = 1, .., K$, are concatenated, thus it is sensitive to the relative E_i order. Overall, all three models perform comparably between each other and to **KGAT** (lines 14-22 vs 5-7). **MaxPool** and **WgtSum** marginally outperform **Concat** with *roberta-large*.

We also trained **Concat** with *roberta-large* setting *K=1* both for training and predicting, i.e., using only top evidence piece retrieved by ES. The resulting LA of 79.57 is only approximately one point behind that of **Concat** (Line 16) and **KGAT** (Line 7). This suggests that good performance can be obtained on the FEVER dataset even without joint reasoning over multiple E_i-s, and that there is still room for further improvement for the systems able to reason upon multiple evidence pieces. Also this could be partially attributed to the observation by Schuster et al. (2019) who showed that FEVER claims contain certain linguistic biases and BERT model fine-tuned on the claim texts only significantly outperforms the majority baseline. Schuster et al. (2019) proposed a debiased symmetric test set, but its instances are claim-evidence pairs. This means that $K = 1$, and thus we did not evaluate our baselines on it as with $K = 1$ they all become equivalent to **Local**.

Comparison to the state of the art. Tab. 5 compares the performance of **MaxPool** and **WgtSum** to that of the SOTA systems as of June 1st, 2021. Our simple baselines outperform all the other systems on DEV, but we may have overfitted on it, as we report the performance of the best checkpoint.

[12]It can be less, as for some claims fewer evidence pieces were retrieved.

System	DEV FEVER	DEV LA	TEST FEVER	TEST LA
Competition systems				
NSMN (#1) (Nie et al., 2019)[13]	66.59	69.6	64.23	68.16
Post-competition systems				
BERT Pair	68.90	73.30	65.18	69.75
BERT Concat	68.89	73.67	65.64	71.01
GEAR (Zhou et al., 2019)	70.69	74.84	67.19	71.60
DREAM (#2 LA) (Zhong et al., 2020)	n/a	79.16	70.60	*76.85*
* KGAT with				
* - BERT Base	75.88	78.02	69.40	72.81
* - RoBERTa Large (Liu et al., 2020)	76.11	78.29	70.38	74.07
Transformer-XH (Zhao et al., 2020)	74.98	78.05	69.07	72.39
KGAT with				
- CorefBERTBase	n/a	n/a	69.82	72.88
- CorefBERTLarge	n/a	n/a	70.86	74.37
- CorefRoBERTaLarge (Ye et al., 2020)	n/a	n/a	72.30	75.96
DOMLIN++ (Stammbach and Ash, 2020)	74.98	77.48	74.27	76.60
Codalab leaderboard as of June 1, 2021				
#1 dominiks	n/a	n/a	**76.78**	**79.16**
#2 h2oloo	n/a	n/a	75.87	79.35
#3 nudt_nlp	n/a	n/a	74.42	77.38
#4 krishnamrith12	n/a	n/a	74.37	79.25
#5 totopower	n/a	n/a	73.90	77.21
#6 gump	n/a	n/a	73.72	77.05
Our results				
Concat (roberta-large)	78.27	**80.31**	72.59	75.85
MaxPool (roberta-large)	78.85	81.16	72.77	76.55
WgtSum (roberta-large)	**79.02**	81.30	**73.44**	**77.18**

Table 5: FEVER state of the art. We mark results outperforming us with *underscore*. We mark the systems using exactly the same input to the ER component as us with *.

On the blind TEST data, **WgtSum** with *roberta-large* scores seventh in terms of FEVER and sixth in terms of LA on the official Codalab leaderboard.

Despite our best efforts, we were not able to track the publications related to the leaderboard submissions #1-#6. We do not know whether their superior performance is due to a better ER approach, a stronger LRM with billions of parameters, or to a better *DocIR/ES*. In the latter two cases, the baselines in this work still remain relevant.

The best-performing system with *published description*, DOMLIN++, uses *roberta-large* and the **Concat** approach. We cannot compare the results of our ER model directly, since they use a different ES system which might have better evidence recall. Note that we still marginally outperform them in terms of LA. This may indicate that even though our gold evidence recall may be lower due to a possibly less powerful DocIR/ES pipeline (resulting in lower FEVER score), we are still able to predict a correct label given the evidence sentences we have at our disposition. Then, they do additional pre-training on MultiNLI, while we do not exploit any external corpora.

Qualitative analysis We compared the outputs of the **Concat**, **MaxPool**, **WgtSum** and **KGAT** systems. We analyzed 50 DEV set examples where only one out of four systems produced the correct label. We aimed to understand the reason behind the correct prediction, but we have not observed any patterns explaining why one system outperforms the others. The systems seem to be equivalent in their abilities.

When analyzing the **WgtSum** output, we observed that when summing the weighted distributed representations of evidence pieces retrieved by KGAT ES for a specific claim (see Sec. 3), it tends to assign higher weights to the evidence pieces which are correct according to the gold standard.

5 Conclusion

We have proposed lightweight strong baselines for the FEVER fact-checking task and showed that they can outperform heavier models on the official leaderboard with blind TEST set. In our future work, we plan to capitalize from our results to build systems that can effectively trade-off efficiency for accuracy.

Acknowledgments.

We thank the anonymous reviewers for their valuable comments.

References

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, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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. *Computing Research Repository*, arXiv:1803.

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. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Computing Research Repository*, arXiv:1907.

Zhenghao Liu, Chenyan Xiong, and Maosong Sun. 2020. Kernel Graph Attention Network for Fact Verification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining Fact Extraction and Verification with Neural Semantic Matching Networks. *Association for the Advancement of Artificial Intelligence (AAAI)*.

Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards Debiasing Fact Verification Models. In *EMNLP*.

Dominik Stammbach and Elliott Ash. 2020. e-FEVER: Explanations and Summaries for Automated Fact Checking. In *Truth and Trust Online*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. 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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-XH: Multi-evidence Reasoning with Extra Hop Attention. In *The Eighth International Conference on Learning Representations (ICLR 2020)*.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning Over Semantic-Level Graph for Fact Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180, Online. Association for Computational Linguistics.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.

A Learning rate selection.

When experimenting with *roberta-base* we tried learning rates (lr) [1e-5; 5e-5] with the step of 1e-5. Table 6 summarizes the results. The results obtained with the learning rates in range [2e-5; 5e-5] are very similar, so we used the learning rate of 2e-5 in the majority of our experiments in this paper.

B Model complexity.

Tab. 7 reports the amount of trainable parameters in the ER component of each model when run on

Model	FEVER	LA	LR
	76.40	78.60	1e-05
	77.09	**79.25**	2e-05
Concat	76.97	79.14	3e-05
	76.81	79.00	4e-05
	76.76	78.95	5e-05
	77.39	79.74	1e-05
	77.66	79.98	2e-05
KGAT	**77.75**	**80.06**	3e-05
	77.53	79.79	4e-05
	75.80	77.93	5e-05
	76.97	79.40	1e-05
	77.48	79.82	2e-05
MaxPool	77.58	79.88	3e-05
	77.61	79.95	4e-05
	77.67	**79.98**	5e-05

Table 6: Experimenting with different learning rates with *roberta-base* as LRM.

Model	# parameters-base	# parameters-large
LRM parameters		
LRM (RoBERTa)	124,645,632	355,359,744
Parameters in the joint inference ER component		
KGAT	792,112	1,318,192
Concat/MaxPool	2,307	3,075
WgtSum	3,076	4,100

Table 7: Number of trainable parameters in the ER models with RoBERTa LRM. We report the amount of LRM and ER component parameters separately (i.e. the full ER model size is their sum). *-base*/*-large* refers to the LRM version.

top of different LRMs[14]. Our simple baselines perform comparably to SOTA using an ER inference component having 3K parameters only (in addition to LRM parameters).

[14]Naturally, we update the LRMs parameters as well.