Kernel Graph Attention Network for Fact Verification

Zhenghao Liu[1] **Chenyan Xiong**[2] **Maosong Sun**[1] *
[1]State Key Laboratory of Intelligent Technology and Systems
Beijing National Research Center for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Microsoft Research AI, Redmond, USA

Abstract

This paper presents Kernel Graph Attention Network (KGAT), which conducts more fine-grained evidence selection and reasoning for the fact verification task. Given a claim and a set of potential supporting evidence sentences, KGAT constructs a graph attention network using the evidence sentences as its nodes and learns to verify the claim integrity using its edge kernels and node kernels, where the edge kernels learn to propagate information across the evidence graph, and the node kernels learn to merge node level information to the graph level. KGAT reaches a comparable performance (69.4%) on FEVER, a large-scale benchmark for fact verification. Our experiments find that KGAT thrives on verification scenarios where multiple evidence pieces are required. This advantage mainly comes from the sparse and fine-grained attention mechanisms from our kernel technique.

1 Introduction

Online contents with unknown integrity, such as fake news, political deception, and online rumors, have been growing significantly and spread widely over the past several years. How to algorithmically and automatically "fact check" the integrity of textual contents, to prevent the spread of fake news, and to avoid the undesired social influences of maliciously fabricated statements, has drawn significant attention from the research community.

Recent research formulates this problem as the fact verification task, which targets to automatically verify the integrity of statements using trustworthy corpora, e.g., Wikipedia (Thorne et al., 2018a). A popular solution is to use natural language inference (NLI) techniques. For example, as shown in Figure 1, a system could first retrieve related evidence sentences from the background corpus and then leverages NLI techniques, such as transformer and graph neural networks, to conduct joint reasoning over the multiple pieces of retrieved evidence and predict the claim integrity (Nie et al., 2018; Zhou et al., 2019; Yoneda et al., 2018; Hanselowski et al., 2018).

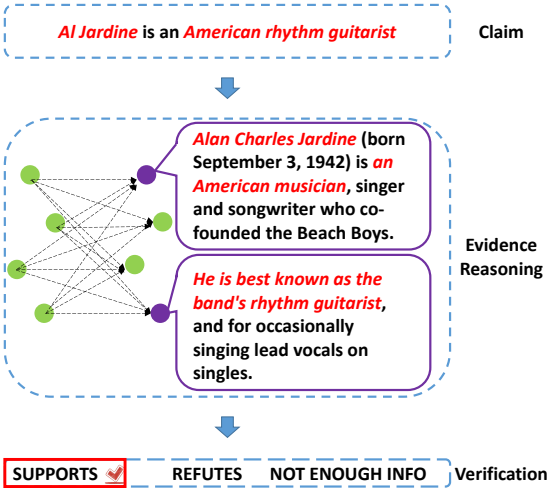


Figure 1: An Example of Fact Verification System.

Effective as they are, standard natural language inference (NLI) techniques are not designed to tackle some unique challenges of fact verification. One challenge is that no ground truth evidence is given and the evidence sentences are retrieved from background corpora, which inevitably includes noise. Another challenge is that the false claims are often deliberately fabricated; they may be semantically correct and related to background knowledge, only meant to deliver fake information. An ideal fact verification system should be able to conduct fine-grained evidence selection

*Corresponding author: M. Sun (sms@tsinghua.edu.cn)

and reasoning, to disentangle the useful evidence pieces from noisy ones, and to distinguish the subtle differences between trust-worthy claims and false statements, which is a challenge for existing neural NLI methods (Thorne et al., 2018b).

This paper presents a new neural structural reasoning model, Kernel Graph Attention Network (KGAT), that provides more fine-grained evidence selection and reasoning capability for fact verification using neural matching kernels (Xiong et al., 2017; Dai et al., 2018). Given retrieved evidence pieces, KGAT first constructs a graph attention network (GAT), using claim and evidence as graph nodes and fully-connected edges. It then utilizes two sets of kernels, one on the nodes and one on the edges, to improve the reasoning on the evidence GAT. The node kernels perform more accurate evidence selection by better matching them with the claim; the edge kernels selectively propagate information between the evidence nodes along the GAT edges, with more fine-grained interactions through the match kernels between the evidence pieces. The two kernels are combined by KGAT, which jointly learns and reasons on the reasoning graph.

In our experiments on FEVER (Thorne et al., 2018a), a large-scale fact verification benchmark, KGAT achieves the 69.4% FEVER score, significantly outperforming previous BERT and GAT based approaches (Zhou et al., 2019). Our experiments demonstrate that KGAT's strong effectiveness on facts that require multiple evidence signals to verify; our ablation study shows that the main source of this effectiveness is the kernel technique KGAT introduced.

Furthermore, our analyses show the effectiveness of kernels by providing more sparse and focused attention. The edge kernels better distinguish useful clues from related evidence and propagate more fine-grained inference signals across the evidence pieces. The node kernels assign higher weights to the correct evidence pieces thus better combines the per-evidence prediction to the final fact verification. Both the edge kernels and node kernels significantly improve the fact verification accuracy; their fine-grained attention are more desired by the multiple evidence scenarios as the verification requires inferring and combining information at variant qualities. The source codes are released via GitHub[1].

[1] https://github.com/thunlp/KernelGAT

2 Related Work

The FEVER shared task (Thorne et al., 2018a) aims to develop automatic fact verification systems to check the veracity of human-generated claims by extracting evidence from Wikipedia. The recently launched FEVER shared task 1.0 is hosted as a competition on Codalab[2] with a blind test set and evaluates all system performance.

FEVER's official baseline (Thorne et al., 2018a) uses a three-step pipeline system (Chen et al., 2017a): document retrieval, sentence retrieval and claim verification. Previous models mainly focus on claim verification reasoning and aggregation over pieces of evidence and predicts claim label. Nie et al. (2018) concatenates all evidence together to verify the claim. One can also conduct reasoning for each claim evidence pair and aggregate them to the claim label (Luken et al., 2018; Yoneda et al., 2018; Hanselowski et al., 2018). TwoWingOS (Yin and Roth, 2018) proposes a joint training system for both the evidence identification and claim verification modules.

Graph Neural Networks (GNN) (Scarselli et al., 2008) excels with their ability to extend existing neural networks for processing the data represented with graph. In GNN, the nodes share information across edges and combine neighbour information for better representations. Much recent research (Veličković et al., 2017; Kipf and Welling, 2016) utilizes different neural architectures to model node interactions for the whole graph. GEAR (Zhou et al., 2019) reasons and aggregates over claim evidence pairs with Graph Attention Network (GAT) (Veličković et al., 2017). (Zhong et al., 2019) further establishes a semantic-level graph for claim and evidence reasoning. These models provide an opportunity to do reasoning over the whole claim-evidence graph.

Many fact verification systems leverage the NLI techniques (Chen et al., 2017b; Ghaeini et al., 2018; Parikh et al., 2016; Radford et al., 2018; Peters et al., 2018; Soleimani et al., 2019) to verify the claim: the NLI task is to classify the relationship between a pair of premise and hypothesis as either entailment, contradiction or neutral, similar to the FEVER task, though the later requires systems to find the evidence sentences themselves and there are often multiple evidence pieces. One of the most widely used NLI mod-

[2] https://competitions.codalab.org/competitions/18814

els in FEVER is Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017b), which employs some forms of hard or soft alignment to associate the relevant sub-components between premise and hypothesis. BERT has also been used for better text representation in FEVER (Devlin et al., 2019; Li et al., 2019; Zhou et al., 2019; Soleimani et al., 2019).

The recent development of neural IR models, especially the interaction base ones, have shown promising effectiveness in extracting soft match patterns from query-document interactions (Hu et al., 2014; Pang et al., 2016; Guo et al., 2016; Xiong et al., 2017; Dai et al., 2018). One of the effective ways to model text matches is to leverage matching kernels (Xiong et al., 2017; Dai et al., 2018), which summarize word or phrase interactions in the learned embedding space between query and documents. The kernel extracts matching patterns which provides a variety of relevance match signals and shows strong performance in various ad-hoc retrieval dataset (Dai and Callan, 2019). Recent research also has shown kernels can be integrated with contextualized representations, i.e., BERT, to better model the relevance between query and documents (MacAvaney et al., 2019).

3 Fact Verification with Kernel Graph Attention Network

Kernel Graph Attention Network (KGAT) performs claim verification by first constructing an evidence graph using retrieved evidence sentences $D = \{e^1, ..., e^p, ...e^l\}$ for claim c, then it conducts fine-grained joint reasoning on the evidence graph with Edge Kernels and Node Kernels, to predict the label y, as shown in Figure 2.

3.1 Evidence Graph Construction

KGAT constructs the evidence graph G by using each claim-evidence pair as a node and connects all node pairs with an edge, making it a fully-connected evidence graph with y nodes: $N = \{n^1, ..., n^p, ...n^l\}$.

Then KGAT leverages the evidence graph to produce the probability of claim label y as:

$$P(y|c, D) = \sum_{p=1}^{l} P(y|c, e^p, D)P(e^p|c, D), \quad (1)$$

or in the graph notation:

$$P(y|G) = \sum_{p=1}^{l} P(y|n^p, G)P(n^p|G). \quad (2)$$

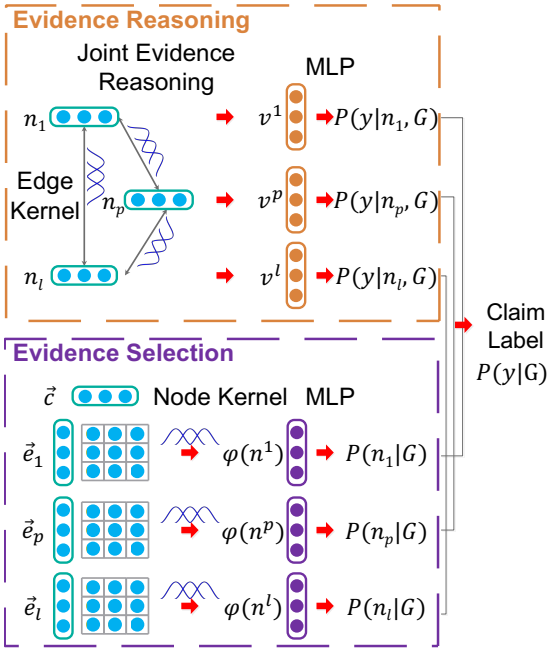


Figure 2: KGAT Architecture.

This derivation follows the standard graph label prediction setting in graph neural network (Veličković et al., 2017), which splits the prediction into two components: the prediction in each node jointly with the whole graph $P(y|n^p, G)$ and the evidence selection probability $P(n^p|G)$. The first is often considered as joint node label output. It propagates information through edges in a graph and enhances each node representation with its neighbors. Then the second readout module (Knyazev et al., 2019) aims to choose the most appropriate node with probability $P(n^p|G)$. The next part describes how KGAT instantiates the two components with kernels.

3.2 Fine-Grained Kernel GAT Reasoning.

KGAT conducts the fine-grained reasoning in the evidence graph by first using BERT for *Initial Node Representation*, and then performs information propagation and node level prediction with Edge-Kernels. The node level predictions are combined by Node-Kernels with different weights (evidence selection).

Initial Node Representations are constructed via encoding the claim, document (Wiki) title, and evidence sentence, as a concatenated sequence by BERT. In the node n_p, the claim and evidence contains m tokens (with "[SEP]") and n tokens (with Wikipedia title and "[SEP]") respectively. Trough the BERT encoder, we get the token hidden set H^p

with the given node n^p:

$$H^p = \text{BERT}(n^p). \quad (3)$$

The "[CLS]" token hidden state encodes node contents and we denote the "[CLS]" representation as z^p for node p:

$$z^p = H^p_0. \quad (4)$$

Then we further separate $H^p_{1:m+n}$ into two groups according to the claim and the evidence, to the claim token hidden state set $H^p_{1:m}$ and the evidence token hidden state set $H^p_{m+1:m+n}$.

Then `KGAT` calculates $P(y|n^p, G)$ and $P(n^p|G)$ with two kernel based architectures: `Edge-Kernel` and `Node-Kernel` respectively.

`Edge-Kernel` calculates the node inference representation v^p to perform a fine-grained reasoning over the whole evidence graph G:

$$v^p = \texttt{Edge-Kernel}(n^p, G). \quad (5)$$

`Node-Kernel` calculates node readout representation $\phi(n^p)$ for evidence selection among l nodes using each node's hidden state set $H^p_{1:m+n}$:

$$\phi(n^p) = \texttt{Node-Kernel}(n^p). \quad (6)$$

`Edge-Kernel` models the interactions between every node pair to improve their node representations using matching kernels. To propagate q-th node content to the p-th node, the kernel is used to get the q-th node's updated node representation $\hat{z}^{q\to p}$ according to the need of p-th node We first conduct a translation matrix $M^{q\to p}$ between q-th node hidden state set $H^q_{1:m+n}$ and p-th node hidden state set $H^p_{1:m+n}$. Each element $M^{q\to p}_{ij}$ in $M^{q\to p}$ is the cosine similarity:

$$M^{q\to p}_{ij} = \cos(H^q_i, H^p_j). \quad (7)$$

Then we use K kernels to extract the local semantic inference feature $\vec{K}(M^{q\to p}_i)$ from the translation matrix $M^{q\to p}$:

$$\vec{K}(M^{q\to p}_i) = \{K_1(M^{q\to p}_i), ..., K_k(M^{q\to p}_i)\}. \quad (8)$$

Each kernel K_k utilizes a Gaussian kernel to extract features and summarizes the translation score as a fine-grained kernel interaction:

$$K_k(M^{q\to p}_i) = \log \sum_j \exp(-\frac{M^{q\to p}_{ij} - \mu_k}{2\delta_k^2}), \quad (9)$$

where μ_k and δ_k are the mean and width for the k-th kernel (Xiong et al., 2017). Then we calculate each token representation weight $\alpha^{q\to p}_i$ for q-th node representation, targeting the p-th node:

$$\alpha^{q\to p}_i = \text{softmax}(w \cdot \vec{K}(M^{q\to p}_i) + b), \quad (10)$$

where w and b are parameters. The kernel based representation $\hat{z}^{q\to p}$ is calculated by weighting all token representations in the q-th node which is propagated to the p-th node:

$$\hat{z}^{q\to p} = \sum_{i=1}^{m+n} \alpha^{q\to p}_i \cdot H^q_i. \quad (11)$$

Then a two layer MLP is used to calculate the attention weight $\beta^{q\to p}$ of q-th node according to the p-th node n^p's "[CLS]" representation z^p:

$$\beta^{q\to p} = w_1 \cdot (\text{ReLu}(w_0 \cdot (z^p \circ \hat{z}^{q\to p}) + b_0)) + b_1, \quad (12)$$

where $\beta^{q\to p}$ weights q-th updated node representation for the p-th node. The w_0, w_1, b_0 and b_1 are parameters to calculate the node weight score. $\circ$ denotes the concatenate operator. Then the p-th node's representation is updated by weighting each node updated representation $\hat{z}^{q\to p}$) with weight $\beta^{q\to p}$. Finally, the aggregated fine-grained representation and p-th node "[CLS]" representation z^p are concatenated for the p-th node `KGAT`'s `Edge-Kernel` representation v^p:

$$v^p = (\sum_{q=1}^{k} \beta^{q\to p} \cdot \hat{z}^{q\to p}) \circ z^p \quad (13)$$

The v^p includes information from all nodes in G, propagated through `Edge-Kernel`.

We utilize p-th node `Edge-Kernel` representation v^p to calculate the claim label y probability $P(y|n^p)$ for p-th node among the claim label set:

$$P(y|n^p) = \text{softmax}_y(w^y \cdot v^p + b^y), \quad (14)$$

where w^y and b^y are parameters to learn.

`Node-Kernel` selects the most relevant nodes in the evidence graph for verification. The kernel with multiple semantic match signals can better help models select the most appropriate evidence. We conduct a translation matrix $M^{c\to e^p}$ between the p-th node claim hidden state set $\vec{c} = H^p_{1:m}$ and evidence hidden state set $\vec{e^p} = H^p_{m+1:m+n}$. Then we get the kernel based match feature $\vec{K}(M^{c\to p}_i)$ by summarizing the translation scores according to the i-th claim token in the p-th node. Finally, we average n claim token match features to get the node selection representation $\phi(n^p)$:

$$\phi(n^p) = \frac{1}{n} \cdot \sum_{i=1}^{m} \vec{K}(M^{c\to e^p}_i). \quad (15)$$

Then we leverage the `Node-Kernel` feature $\phi(n^p)$ to calculate p-th evidence selection probability $P(n^p|G)$:

$$P(n^p|G) = \text{softmax}_p(w^* \cdot \phi(n^p) + b^*), \quad (16)$$

where w^* and b^* are parameters.

3.3 Evidence Aggregation with Fine-grained Salience Modeling.

Finally, we calculate the claim label y probability $P(y|c, D)$ by combining the per node probability of $P(y|n^p, G)$ and node salience $P(n^p|G)$:

$$P(y|G) = \sum_{p=1}^{l} P(y|n^p, G)P(n^p|G). \quad (17)$$

Then we minimize the cross entropy loss L and optimize all parameters end-to-end:

$$L = \text{CrossEntropy}(y^*, P(y|G))), \quad (18)$$

using the ground truth verification label y^*.

4 Experimental Methodology

This section describes the dataset, evaluations, baselines, and implementation details of our experiments.

Dataset. A large scale public fact verification dataset FEVER (Thorne et al., 2018a) is used in our experiments. The FEVER consists of 185,455 annotated claims with 5,416,537 Wikipedia documents from the June 2017 Wikipedia dump. All claims are classified as SUPPORTS, REFUTES or NOT ENOUGH INFO by annotators. The dataset partition is kept the same with the FEVER Shared Task (Thorne et al., 2018b) as shown in Table 1.

Evaluation Metrics. The official evaluation metrics[3] for claim verification include FEVER score and Label Accuracy (LA). The FEVER score considers whether one complete set of golden evidence is provided and better reflects the inference ability. LA is a general evaluation metric, which calculates claim classification accuracy rate without considering retrieved evidence.

In addition, Golden FEVER (GFEVER) score is added in our experiments. GFEVER evaluates the FEVER score when golden evidence is given to the system, which is an easier setting. The FEVER dataset also provides the gold labels for evidence sentences and precision, recall and F1 are used to evaluate evidence sentence retrieval accuracy.

Baselines. The baselines include the top three models during FEVER 1.0 shared task, BERT based models and top models on the leaderboard.

Three top models in FEVER 1.0 shared task are compared. Athene (Hanselowski et al., 2018) and UNC NLP (Nie et al., 2018) utilize ESIM to encode claim evidence pairs. UCL MRG (Yoneda et al., 2018) leverages Convolutional Neural Network (CNN) to encode claim and evidence. These three models aggregate evidence by attention mechanism or label aggregation component. In addition, UNC NLP also incorporates other information, such as pageview frequency and WordNet.

Table 1: Statistics of FEVER Dataset (Thorne et al., 2018a)

Split	SUPPORTED	REFUTED	NOT ENOUGH INFO
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

The BERT based models are our main baselines, they have big improvements than previous methods without pre-training. BERT-pair BERT-concat and GEAR are three baselines from the previous work (Zhou et al., 2019). BERT-pair is trained to predict each evidence-claim pair label and only the most relevant evidence is leveraged to predict the claim label. The BERT-concat system concatenates all evidence together to predict the claim label. GEAR utilizes a graph attention network to extract supplement information from other evidence and aggregate all evidence through an attention layer. To evaluate kernel effectiveness, we replace KGAT's kernels with dot products, which leads to the vanilla GAT model. GAT is almost the same as GEAR in terms of architecture.

Some models achieve better performances on Codalab leaderboard. Zhong et al. (2019); Soleimani et al. (2019) use XLNet (Yang et al., 2019) and BERT (Large) for evidence reasoning, which are not comparable with previous BERT (Base) reasoning models (Zhou et al., 2019).

Implementation details. The rest of this section describes our implementation details.

Document retrieval. The document retrieval step retrieves related Wikipedia pages and is kept the same with previous work (Hanselowski et al., 2018; Zhou et al., 2019). For a given claim, it first utilize the constituency parser in AllenNLP (Gardner et al., 2018) to extract all phrases which potentially indicate entities. Then it uses these phrases as queries to find relevant Wikipedia pages through the online MediaWiki API[4]. The seven highest-ranked results for each query are used to

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

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

form a candidate article set. Then Hanselowski et al. (2018) drops the articles which are not in the offline Wikipedia dump and filters the articles by the word overlap between their titles and the claim (Hanselowski et al., 2018).

Sentence retrieval. The sentence retrieval part focuses on selecting related sentences from retrieved pages. There are two sentence retrieval models in our experiments: ESIM based sentence retrieval model and BERT based sentence retrieval model. The ESIM based sentence retrieval model keeps the same parameter setting as the previous work (Hanselowski et al., 2018; Zhou et al., 2019).

Claim verification. During training, we set the batch size to 4 and accumulate step to 8. All models are evaluated with LA on the development set and trained for two epochs. The training and development sets are built with golden evidence and higher ranked evidence with sentence retrieval. All claims are assigned with five pieces of evidence. To keep the same evaluation setting, we also evaluate our models with ESIM, the same sentence retrieval as GEAR (Zhou et al., 2019).

We use the base version of BERT (Devlin et al., 2019) in all experiments. The max length is set to 130. All models are implemented with PyTorch. BERT inherits huggingface's PyTorch implementation[5]. Adam is utilized to optimize all parameters with learning rate = 5e-5 and warm up proportion is set to 0.1. The kernel is set to 21.

5 Evaluation Result

Three experiments are conducted to study the overall performance of KGAT, its advantages on multiple and single inference scenarios, and the effectiveness of kernels. We provide case studies to illustrate KGAT's characteristics in Appendix A.

5.1 Overall Performance

The fact verification performances are shown in Table 2. It includes KGAT results with both ESIM as the sentence retrieval model, same as previous work (Zhou et al., 2019; Hanselowski et al., 2018), and also the BERT based sentence retrieval model, "BERT Retrieval". In the same document retrieval and sentence retrieval setting (ESIM), KGAT outperforms previous published BERT (Base) state-of-the-art GEAR, which also uses BERT and GAT, by more than 1% on FEVER score. With BERT sentence ranker, KGAT performs even better and its FEVER scores on test set improved another 1%. KGAT contains less parameters and presents comparable performance with BERT (Large) model (Soleimani et al., 2019) .

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

Table 2: Fact Verification Accuracy. Some results are not available. $*$ indicates not published work.

Model	Dev		Test	
	LA	FEVER	LA	FEVER
Athene (Hanselowski et al., 2018)	68.49	64.74	65.46	61.58
UCL MRG (Yoneda et al., 2018)	69.66	65.41	67.62	62.52
UNC NLP (Nie et al., 2018)	69.72	66.49	68.21	64.21
BERT Concat	73.67	68.89	71.01	65.64
BERT Pair	73.30	68.90	69.75	65.18
GEAR (Zhou et al., 2019)	74.84	70.69	71.60	67.10
leaderboard #1 (Zhong et al., 2019)	**79.16**	-	**76.85**	**70.60**
leaderboard #2 (Soleimani et al., 2019)	74.59	72.42	71.86	69.66
leaderboard #4 (cunlp)*	-	-	72.47	68.80
KGAT w. ESIM Retrieval	75.51	71.61	72.48	68.16
KGAT w. BERT Retrieval	78.02	**75.88**	72.81	69.40

Table 3: Evidence Sentence Retrieval Accuracy. Sentence level **Prec**ision, **Rec**all and **F1** are evaluated by official evaluation (Thorne et al., 2018a).

	Model	Prec@5	Rec@5	F1@5	FEVER
Dev	ESIM	24.08	86.72	37.69	71.70
	BERT	**27.29**	**94.37**	**42.34**	**75.88**
Test	ESIM	23.51	84.66	36.80	68.16
	BERT	**25.21**	**87.47**	**39.14**	**69.40**

The sentence retrieval performances of ESIM and BERT are further compared in Table 3. The recall indicates the number of golden evidence in top 5 retrieved sentences, which is crucial for claim verification. On the test scenario, the BERT based methods outperform ESIM by about 3% for the evidence recall, which illustrates BERT based model can include more useful evidence for claim verification. Nevertheless, the BERT retrieval model achieves a dramatic improvement on development set and boosts FEVER score a lot, but not as much on test. The gap of performance on development set and test set between ESIM and BERT retrieval models is discrepant. Therefore, all our following experiments evaluate models on the ESIM based sentence retrieval result in the following experiments to avoid this discrepancy.

5.2 Performance on Different Scenarios

This experiment studies the effectiveness of kernel on multiple and single evidence reasoning scenar-

Table 4: Claim Verification Accuracy on Claims that requires Multiple evidence pieces and Single evidence Pieces. Standard GAT with no kernel (GAT), with only node kernel (KGAT-Node), with only edge kernel (KGAT-Edge) and the full model (KGAT-Full) are compared.

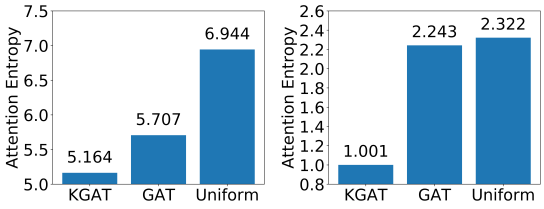
Reasoning	Model	LA	GFEVER	FEVER	
Multiple	GAT	**66.12**	84.39	38.21	-
	KGAT-Node	65.51	83.88	38.52	0.31%
	KGAT-Edge	65.87	84.90	39.08	0.87%
	KGAT-Full	65.92	**85.15**	**39.23**	1.02%
Single	GAT	79.79	81.96	77.42	-
	KGAT-Node	79.92	82.29	77.73	0.31%
	KGAT-Edge	79.90	82.41	77.58	0.16%
	KGAT-Full	**80.33**	**82.62**	**78.07**	0.65%

ios, as well as the contribution of kernels. The results are shown in Table 4.

The verifiable instances are separated (except instances with "NOT ENOUGH INFO" label) into two groups according to the golden evidence set. If more than one evidence is required, the instance is divided into the multiple evidence reasoning set. The single evidence reasoning set and the multiple evidence reasoning set contain 11,372 (85.3%) and 1,960 (14.7%) instances respectively.

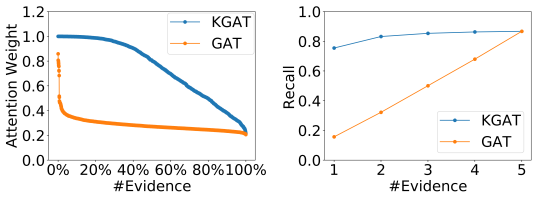
The multiple and single evidence reasoning scenarios evaluate model reasoning ability from different aspects. The single evidence reasoning scenario mainly focuses on how to infer with single evidence and select the most relevant evidence for claim verification. It mainly evaluates model denoising ability with retrieved evidence set. The multiple evidence reasoning is a harder and more complex scenario, requiring the model to summarize necessary clues and reason over multiple evidence. It pays more attention to evaluate the evidence interactions for the joint reasoning.

KGAT-Node outperforms GAT by more than 0.3% for both single and multiple reasoning scenarios and shows its effectiveness for evidence selection. On the contrary, the KGAT-Node does not help much on GFEVER, because the golden evidence is given. It illustrates the KGAT-Node model mainly focuses on choosing appropriate evidence to help model improve. KGAT-Edge outperforms GAT by more than 0.8% and 0.1% on multiple evidence reasoning and single evidence reasoning scenarios respectively, which demonstrates KGAT-Edge can better extract and emphasize important clues from neighbour evidence



(a) Edge Attention. (b) Node Attention.

Figure 3: Attention Weight Entropy on Evidence Graph, from KGAT and GAT, of graph edges and nodes. Uniform weights' entropy is also shown for comparison.



(a) Attention Weight Distribution. (b) Evidence Selection Recall.

Figure 4: Evidence Selection Effectiveness of KGAT and GAT. Fig 4(a) shows the distribution of attention weights on evidence nodes $p(n^p)$, sorted by their weights; Fig 4(b) evaluates the recall of selecting the golden standard evidence nodes at different depths.

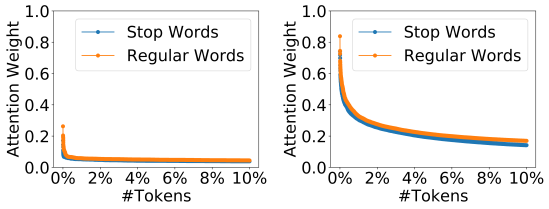
nodes. The KGAT-Edge model shows the joint reasoning ability to help model improve. KGAT-Node and KGAT-Edge show their advancements on single and multiple evidence reasoning scenarios, which illustrates their abilities for evidence selection and joint reasoning, respectively.

The KGAT presents its effectiveness especially for the multiple reasoning scenario, which illustrates KGAT's strong reasoning ability. The KGAT-Node and KGAT-Edge play different roles in the whole model and cooperate for the further improvement. The next experiment further studies the effectiveness of the kernels in KGAT.

5.3 Effectiveness of Kernel in KGAT

This experiment explores the kernel's roles in KGAT by their overall performance, effectiveness of KGAT-Node and effectiveness of KGAT-Edge.

More Concentrated Attention. This experiment studies kernels' properties by their attention entropy, which is shown in Figure 3. The attention entropy reflects whether the learned attention weights are focused or broad. All experiments



(a) GAT. (b) KGAT.

Figure 5: The Attention Weight Distribution from GAT and KGAT on evidence sentence tokens. Top 10% tokens are presented. The rest follows standard long tail distributions.

compare KGAT, GAT and the uniform attention weight distribution to illustrate our KGAT's effectiveness. Edge attention and Node attention are presented to evaluate model attention is focused or broad from token highlight and evidence selection.

The attention entropy for Edge is shown in Figure 3(a). Both GAT and KGAT perform a smaller entropy for the token attention than the uniform distribution. It illustrates that GAT and KGAT both focus on some tokens instead of uniform weight distribution. On the other hand, KGAT shows a smaller entropy value than GAT. It demonstrates KGAT highlights less words with more weight than GAT. For the Node attention, as shown in Figure 3(b), the GAT almost shares the same entropy with uniform distribution. It seems that the dot product based evidence attention perform a uniform attention weight to all evidence, which shows scattered selection evidence. On the contrary, KGAT shows strong ability to focus on few evidence with a less attention entropy value.

This experiment shows that kernels provide more focused attention. Next experiments further study what the kernels focus on.

More Accurate Evidence Selection. The kernel effectiveness of KGAT-Node is presented with Node attention distribution and evidence recall, as shown in Figure 4.

We first calculate the maximum Node attention score among retrieved l evidence for an instance. Then the random sampled instances are sorted according to the maximum Node attention score. The maximum evidence score distribution of both GAT and KGAT are plotted in Figure 4(a). KGAT puts more weight on a few evidence nodes, which further presents the KGAT-Node has a focused attention to select evidence. Then the selection performance is evaluated by the evidence recall. We sort all evidence according to KGAT-Node score and evaluate recall from top 1 to 5 evidence, as shown in Figure 4(b). The evidence recall evaluates the evidence selection accuracy. GAT evidence recall is increased when more evidence is included. On the contrary, KGAT achieves the high recall on top picks. KGAT-Node sorts golden evidence more ahead than GAT and KGAT-Node is more effective to select the golden evidence.

More Focused Attention on Evidence Contents. Figure 5 presents the token attention distribution from the Edges.

We sort words from random sampled 10% instances according to GAT-Edge score and KGAT-Edge score, as shown in Figure 5(a) and Figure 5(b), respectively. KGAT salience focuses on a few words. The GAT word distribution is uniform and only a few words are weighted with higher (Clark et al., 2019), but less than major probability. The different Edge attention distributions of KGAT and GAT demonstrate KGAT-Node with a better fine-grained attention has a strong ability to emphasize and extract important clues from neighbor evidence nodes, which is more suitable for the multiple evidence scenario.

6 Conclusion

This paper presents Kernel Graph Attention Model (KGAT) for fact verification. KGAT brings in the kernel match technique from neural information retrieval to a structured natural language inference technique, Graph Attention Network. It provides more focused attention mechanisms in the evidence graph, improving both the evidence selection accuracy and the joint reasoning accuracy, thus more accurately verifies the claim integrity.

Our experiments on the fact verification benchmark, FEVER, demonstrate KGAT's effectiveness. Its advantages are more observed on claims that require multiple evidence pieces to verify. Our analyses reveal that the kernel technique is the main source of this effectiveness.

Kernels provide rather interesting attention patterns. They lead to attention mechanisms more concentrated on a selective set of contents or nodes, helping the model better absorb the training signals for more task-specialized attentions. We believe this phenomenon leads to the effectiveness of KGAT in the fact verification task, and will explore more usages of kernels in neural networks and the mechanisms behind in the future.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1870–1879.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. 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.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341* .

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. *arXiv preprint arXiv:1905.09217* .

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018)*. ACM, pages 126–134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. pages 1–6.

Reza Ghaeini, Sadid A Hasan, Vivek Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Yuan Ling, Aaditya Prakash, Xiaoli Fern, and Oladimeji Farri. 2018. Dr-bilstm: Dependent reading bidirectional lstm for natural language 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 1460–1469.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W.Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016)*. ACM, pages 55–64.

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.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS 2014)*. MIT Press, pages 2042–2050.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Boris Knyazev, Graham W Taylor, and Mohamed R Amer. 2019. Understanding attention in graph neural networks. *arXiv preprint arXiv:1905.02850* .

Tianda Li, Xiaodan Zhu, Quan Liu, Qian Chen, Zhigang Chen, and Si Wei. 2019. Several experiments on investigating pretraining and knowledge-enhanced models for natural language inference. *arXiv preprint arXiv:1904.12104* .

Jackson Luken, Nanjiang Jiang, and Marie-Catherine de Marneffe. 2018. Qed: A fact verification system for the fever shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. pages 156–160.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. *arXiv preprint arXiv:1904.07094* .

Yixin Nie, Haonan Chen, and Mohit Bansal. 2018. Combining fact extraction and verification with neural semantic matching networks. *arXiv preprint arXiv:1811.07039* .

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*. pages 2793–2799.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 2249–2255.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*. pages 2227–2237.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. In *Proceedings of Technical report, OpenAI*. https://www.cs.ubc.ca/ amuham01/LING530/papers/radford2018impr

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20(1):61–80.

Amir Soleimani, Christof Monz, and Marcel Worring. 2019. Bert for evidence retrieval and claim verification. *arXiv preprint arXiv:1910.02655* .

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. 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 809–819.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* .

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, pages 55–64.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* .

Wenpeng Yin and Dan Roth. 2018. Twowingos: A two-wing optimization strategy for evidential claim verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 105–114.

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.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019. Reasoning over semantic-level graph for fact checking. *arXiv preprint arXiv:1909.03745* .

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 Annual Meeting of the Association for Computational Linguistics, Volume 1 (Long Papers)*.

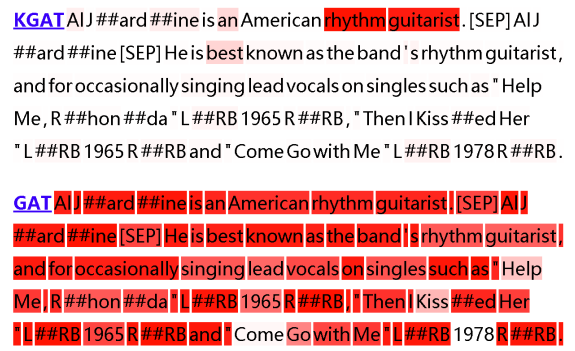


Figure 6: Edge Attention Weights on Evidence Tokens. Darker red indicates higher attention weights.

Table 5: An example claim (Zhou et al., 2019) whose verification requires multiple pieces of evidence.

Claim: ***Al Jardine*** is an ***American rhythm guitarist***.
(1) **[Wiki/Al Jardine]** ***Alan Charles Jardine*** (born September 3, 1942) is ***an American musician***, singer and songwriter who co-founded the Beach Boys.
(2) **[Wiki/Al Jardine]** ***He is best known as the band's rhythm guitarist***, and for occasionally singing lead vocals on singles such as "Help Me, Rhonda" (1965), "Then I Kissed Her" (1965) and "Come Go with Me" (1978).
(3) **[Wiki/Al Jardine]** In 2010, Jardine released his debut solo studio album, A Postcard from California.
(4) **[Wiki/Al Jardine]** In 1988, Jardine was inducted into the Rock and Roll Hall of Fame as a member of the Beach Boys.
(5) **[Wiki/Jardine]** Ray Jardine American rock climber, lightweight backpacker, inventor, author and global adventurer.
Label: SUPPORT

A Appendix

Table 5 shows an example claim whose verification needs to perform reasoning over multiple pieces of evidence. We leverage the same example used by GEAR for fair comparison (Zhou et al., 2019). The first two evidence sentences are ground truth evidence.

Figure 6 presents the attention distribution on the second evidence, required by the first evidence ($\alpha_i^{2\to1}$). It illustrates how much attention weights evidence (1) puts on evidence (2)'s tokens, in order to gather useful information to update the representation of evidence (1). Darker red colors indicate high attention weights.

The first evidence verifies that "Al Jardine is an American musician" but has not enough information to verify "Al Jardine is a rhythm guitarist". In KGAT, the edge kernels accurately pick up the information evidence (1) needed from evidence (2): "rhythm guitarist", which fills the missing information and completes the reasoning chain. Also, "Al Jardine" also receives more attention, which verifies the evidence is about the person in the claim. Such kernel attention is more intuitive and effective than the dot-product attention in GAT, where the attention is scattered almost uniformly across all tokens, similar to the observations in recent research (Clark et al., 2019).