Knowledge-Enhanced Natural Language Inference Based on Knowledge Graphs

Zikang Wang[1,2], Linjing Li[1,2,3], and Daniel Zeng[1,2,3]

[1]State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
[3]Shenzhen Artificial Intelligence and Data Science Institute (Longhua), Shenzhen, China
{wangzikang2016, linjing.li, dajun.zeng}@ia.ac.cn

Abstract

Natural Language Inference (NLI) is a vital task in natural language processing. It aims to identify the logical relationship between two sentences. Most of the existing approaches make such inference based on semantic knowledge obtained through training corpus. The adoption of background knowledge is rarely seen or limited to a few specific types. In this paper, we propose a novel Knowledge Graph-enhanced NLI (KGNLI) model to leverage the usage of background knowledge stored in knowledge graphs in the field of NLI. KGNLI model consists of three components: a semantic-relation representation module, a knowledge-relation representation module, and a label prediction module. Different from previous methods, various kinds of background knowledge can be flexibly combined in the proposed KGNLI model. Experiments on four benchmarks, SNLI, MultiNLI, SciTail, and BNLI, validate the effectiveness of our model.

1 Introduction

Natural Language Inference (NLI) is a fundamental yet challenging task for natural language understanding. NLI aims to determine whether the logical relationship between a premise p and a hypothesis h is *entailment*, *neutral*, or *contradiction*. NLI requires reasoning and inference abilities which are crucial for artificial intelligence system.

Recent years have witnessed a large improvement in NLI models because of the release of several large-scale corpora, such as SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). These datasets enable various deep learning models to achieve state-of-the-art performances (Chen et al., 2017; Chen et al., 2018; Pan et al., 2018; Ghaeini et al., 2018).

Most of the existing NLI models are based on cross sentence attention (Chen et al., 2017). They learn alignments according to the attention between premises and hypotheses. The resulting alignments and semantic representations of sentences are aggregated and fed into a multilayer feed-forward neural network to judge logical relationships. However, only semantic knowledge between premises and hypotheses are utilized in most of these models.

Background knowledge can be utilized to facilitate inference as shown in Fig. 1. For the sentences located in the left part of Fig. 1, the logical relationship between the *Premise* and the *Hypothsis* largely depends on the relationship between "piano" and "music". However, the latter is not explicitly expressed in the sentences per se. The right part of Fig. 1 is a part of a large knowledge graph, in which paths between "piano" and "music" represent background knowledge that can be utilized to determine the relationship of the sentence pair. Previous work (Chen et al., 2018) shows that NLI models can benefit from leveraging external knowledge. But only restricted kinds of knowledge are considered in it. How to flexibly incorporate a variety of different background knowledge in NLI is still a challenging task.

In this paper, we propose a Knowledge Graph enhanced Natural Language Inference (KGNLI) model. KGNLI enhances the performance of NLI by introducing background knowledge stored in knowledge graphs. To be more specific, KGNLI first extracts entities such as subjects, predicates, and objects from

Proceedings of the 28th International Conference on Computational Linguistics, pages 6498–6508
Barcelona, Spain (Online), December 8-13, 2020

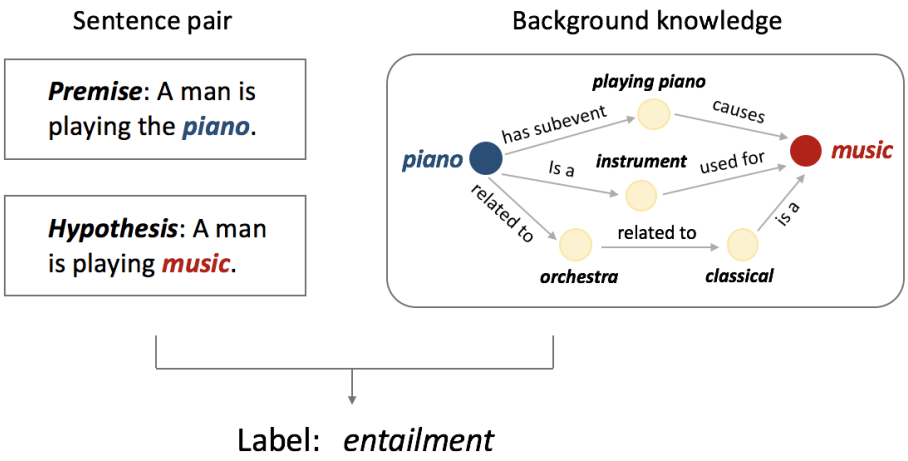

Figure 1: The role of knowledge graphs. Knowledge graphs can provide background knowledge for the NLI problem. For the sentence pair in the figure, their relationship largely depends on the relationship of entities "piano" and "music", which can be learned from the paths in the knowledge graph.

the given sentence pair, then learns a knowledge-relation representation based on a predetermined knowledge graph which contains these entities as nodes. Besides it, KGNLI also learns a semantic-relation representation between the given sentences by Bi-directional Long Short-Term Memory (BiLSTM) network. Finally, KGNLI combines these two representations and feed it into a multilayer perceptron to determine the label of the relationship.

In order to evaluate our model, we conduct experiments on four datasets: SNLI, MultiNLI, SciTail, and BNLI. Our model gets competitive results on all the four datasets. On dataset SciTail and BNLI, where knowledge is crucial for inference (Glockner et al., 2018), our model achieves large improvements against baselines. We further conduct ablation tests to validate the effectiveness and necessity of each component in the proposed model.

2 Related Work

2.1 Natural Language Inference

Traditional NLI models are trained on small-scale datasets, like natural logic-based and co-occurrence statistics-based models, the former identifies inferences by lexical and syntactic features (MacCartney and Manning, 2008), while the latter considers the statistical features (Glickman and Dagan, 2005).

The emergence of large-scale datasets, such as SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018), has stimulated research and development of new models based on deep neural network. Various architectures have been proposed to capture the interaction and soft alignment between sentences. For instances, ESIM (Chen et al., 2017) considers recursive architectures in both local inference modeling and inference composition, CAFE (Tay et al., 2018) architecture propagates compressed alignment features to upper layers to enhance representation learning, DMAN (Pan et al., 2018) adopts reinforcement learning with discourse markers to help improve the performance. Although these models have achieved state-of-the-art results in NLI related tasks, they only rely on semantic relationship learned from training corpus, in other words, no external knowledge is utilized explicitly to facilitate inference.

2.2 Knowledge Enhanced NLI

To the best of our knowledge, the only neural model adopting external knowledge is KIM (Chen et al., 2018). This model incooperates basic knowledge about *synonymy*, *antonymy*, *hypernymy*, *hyponymy*, and *co-hyponyms* to help model soft-alignments between sentence pairs. However, it can only deal with a fixed number of types of knowledge, and pre-assigned scores for relationships are needed before training, which limits its applications in practice.

2.3 Knowledge Graph

A knowledge graph is a large knowledge base storing relational knowledge in a graph structure. Knowledge is formatted as triples in knowledge graph, a triple (h, r, t) indicates that head entity h and tail entity t have relation r. There are many open source knowledge graphs which can be easily employed in a variety of applications, such as WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), and Concept Graph (Cheng et al., 2015; Wu et al., 2012). Knowledge graph has been proved very helpful in various natural language processing tasks, such as machine reading comprehension (Yang et al., 2019), language modeling (Logan et al., 2019), and question answering (Xiong et al., 2019). In this paper, knowledge graph is employed to provide external background knowledge.

3 Methodology

Given a pair of sentences, premise p and hypothesis h, the goal of NLI is to predict a label y that indicates the logical relationship between sentences p and h. The set of labels includes *entailment* (h can be logically deduced from p), *neutral* (p and h do not have any logical relationship), and *contradiction* (p and h cannot be true simultaneously).

The proposed model consists of three major components, as showed in Fig. 2. The novel knowledge-relation representation module builds the relationship between p and h based on background knowledge, while the semantic-relation representation module captures sentence semantic relationship. Finally, a multilayer perceptron merges both knowledge and semantic relationships and predicts the label.

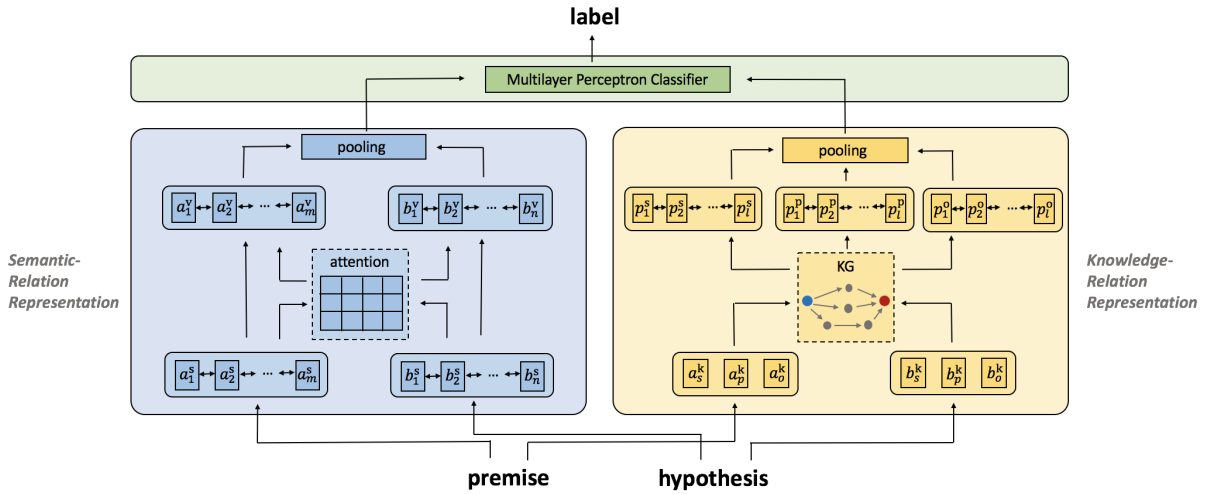

Figure 2: The overall architecture of the model. It contains three major components: knowledge-relation representation module, semantic-relation representation module, and label prediction module.

3.1 Knowledge-Relation Representation

To build the relationship between p and h based on background knowledge, we propose a novel knowledge-relation representation module. In this paper, we assume that the relationship between sentences is determined by the relationship of their subjects, predicates, and objects. The architecture of this module is shown in Fig. 3.

3.1.1 Sub-graph of background relationship

In the following section, we denote the subject pair, predicate pair, and object pair of p and h as (p_S, h_S), (p_P, h_P), and (p_O, h_O), respectively. For each sentence pair, the sub-graph of background relationship for subject pair (p_S, h_S) is extracted by finding paths between entities that denote p_s and h_s in the predetermined knowledge graph KG with the help of random walking. The sub-graphs for predicate pair (p_P, h_P) and object pair (p_O, h_O) are constructed in the same way. Fig. 3 shows three paths in the

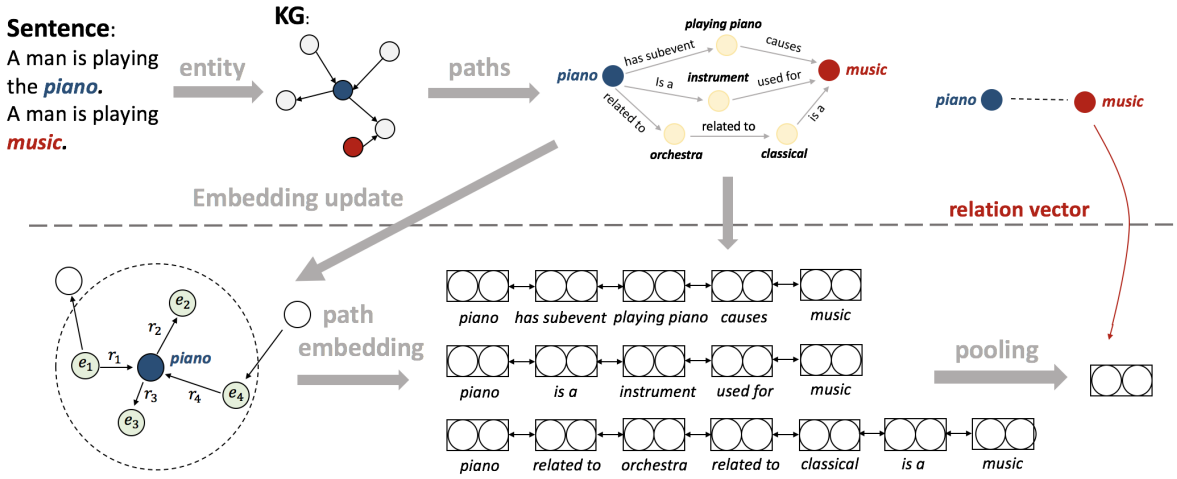

Figure 3: External Knowledge Encoding. For entities "piano" and "music", we first find the paths between them in the knowledge graph, and update their embeddings using graph neural networks. Then we learn the embeddings of the paths. In this way, we encode the knowledge into pooled embeddings.

sub-graph of the object pair (piano, music), "piano-has subevent-playing piano-causes-music", "piano-is a-instrument-used for-music", and "piano-related to-orchestra-related to-classical-is a-music". The lengths of the first two paths are 5, while it is 7 of the last one.

In this paper, we extract subjects, predicates, and objects of sentences based on their syntax tree, consider paths with length up to $\overline{L}$ and limit the total number of paths of each sub-graph as $\overline{N}$.

3.1.2 Knowledge Embedding

With the help of sub-graphs, we can learn the knowledge-based relationship of sentence pairs.

First, the knowledge-based embeddings for entities in sub-graphs are learned. Denote the knowledge embedded vectors of p and h as $\{a_S^{\mathrm{k}}, a_P^{\mathrm{k}}, a_O^{\mathrm{k}}\}$ and $\{b_S^{\mathrm{k}}, b_P^{\mathrm{k}}, b_O^{\mathrm{k}}\}$, where S, P and O are indices of subject, predicate, and object as above, k indicates that the embeddings are learned based on knowledge graphs. We initialize entity embeddings based on pre-trained vectors that are generated by TransE (Bordes et al., 2013),

$$a_i^{\mathrm{k}} = \mathrm{TransE}(p_i), i \in \{S, P, O\}, \tag{1}$$

$$b_j^{\mathrm{k}} = \mathrm{TransE}(h_j), j \in \{S, P, O\}. \tag{2}$$

We then update these embeddings based on the sub-graphs using graph neural network. For an entity, we retrieve all the neighboring relations of it in the sub-graph, and encode the neighboring knowledge into its embedding through the following propagation rule,

$$a_i^{\mathrm{k}} \leftarrow \gamma a_i^{\mathrm{k}} + (1-\gamma) \sum_{(e,r)\in\mathcal{S}_{p_i}} \varphi_p(e,r)\sigma(W[a_e^{\mathrm{k}}; a_r^{\mathrm{k}}]), \tag{3}$$

$$b_j^{\mathrm{k}} \leftarrow \gamma b_j^{\mathrm{k}} + (1-\gamma) \sum_{(e,r)\in\mathcal{S}_{h_j}} \varphi_h(e,r)\sigma(W[b_e^{\mathrm{k}}; b_r^{\mathrm{k}}]), \tag{4}$$

where a_i^{k} and b_j^{k} are the propagated entity embedding, γ is a tradeoff parameter, and $\mathcal{S}_{p_i}$ represents the set of all (e, r) pairs of p_i, where e is a neighobor of p_i, and r is the relation between p_i and e. $\sigma(\cdot)$ is the activation function, W is a transformation matrix, $[\cdot\,;\cdot]$ denotes the concatenation operator, and $\varphi_p(e, r)$ is an attention score over (e, r), which is calculated based on the embedding of the entity and its neighbors. For premise, $\varphi_p(e, r)$ is computed as

$$\varphi_p(e,r) = \frac{\exp\left(W[a_e^{\mathrm{k}}, a_r^{\mathrm{k}}]\right)}{\sum_{(e,r)\in\mathcal{S}_{p_i}} \exp\left(W[a_e^{\mathrm{k}}, a_r^{\mathrm{k}}]\right)}, \tag{5}$$

where a_e^{k} and a_r^{k} are the embeddings of entity e and relation r initialized by TransE. For hypothesis, the attention score is calculated in the same way,

$$\varphi_h(e,r) = \frac{\exp\left(W[b_e^{\mathrm{k}}, b_r^{\mathrm{k}}]\right)}{\sum_{(e,r)\in\mathcal{S}_{h_i}} \exp\left(W[b_e^{\mathrm{k}}, b_r^{\mathrm{k}}]\right)}. \tag{6}$$

3.1.3 Relationship Representation

We get the relationship representation based on the representation of paths in the corresponding subgraph. Denote the i-th path between subject pair (p_S, h_S) as

$$l_i^S = [p_S, r_1, e_1, r_2, e_2, \ldots, h_S], \tag{7}$$

where r_j and e_j denotes the j-th relation and entity on the path. The paths l_i^P and l_i^O of predicate and object pair are defined in the same way.

We encode the path sequence with BiLSTM. Relations are represented by the average of the representations of all paths

$$\omega_S^{\mathrm{p}} = \frac{1}{N}\sum_{i=1}^{N} \text{BiLSTM}(l_i^S) \tag{8}$$

$$\omega_P^{\mathrm{p}} = \frac{1}{N}\sum_{i=1}^{N} \text{BiLSTM}(l_i^P) \tag{9}$$

$$\omega_O^{\mathrm{p}} = \frac{1}{N}\sum_{i=1}^{N} \text{BiLSTM}(l_i^O) \tag{10}$$

where ω_S^{p}, ω_P^{p} and ω_O^{p} denote the relation representations between subjects, predicates and objects. We set entity embeddings as the updated embeddings and relation embeddings according to TransE.

3.1.4 Knowledge Composition

We use a composition layer to merge the relationship of subject pair, predicate pair, and object pair. Apart from relation representations ω_S^{p}, ω_P^{p} and ω_O^{p}, we also consider the correlation among them by using element-wise product $\odot$,

$$v^{\mathrm{k}} = G_{\mathrm{k}}([\omega_S^{\mathrm{p}}; \omega_P^{\mathrm{p}}; \omega_O^{\mathrm{p}}; \omega_S^{\mathrm{p}} \odot \omega_P^{\mathrm{p}}; \omega_O^{\mathrm{p}} \odot \omega_P^{\mathrm{p}}]) \tag{11}$$

where v^{k} is the composed representation, and G_{k} is a feed-forward neural network with ReLU as the activation function.

3.2 Semantic-Relation Representation

To capture the semantic relationship between premise p and hypothesis h, we follow the widely adopted framework to get the relationship representations (Chen et al., 2017). In the following, we denote sentences p and h as $p = [p_1, p_2, \ldots, p_m]$ and $h = [h_1, h_2, \ldots, h_n]$, where p_i and h_j are words, $1 \le i \le m,\ 1 \le j \le n$, m and n are the lengths of premise p and hypothesis h.

3.2.1 Semantic Embedding

We first initialize words into embeddings based on pre-trained word vectors GloVe (Pennington et al., 2014), then encode premise and hypothesis using bidirectional LSTM (BiLSTM). Denote a^{s} and b^{s} as embedding vectors of both premise and hypothesis, $a^{\mathrm{s}} = [a_1^{\mathrm{s}}, a_2^{\mathrm{s}}, \ldots, a_m^{\mathrm{s}}]$, $b^{\mathrm{s}} = [b_1^{\mathrm{s}}, b_2^{\mathrm{s}}, \ldots, b_n^{\mathrm{s}}]$, $a_i^{\mathrm{s}} \in \mathbb{R}^k$, $b_j^{\mathrm{s}} \in \mathbb{R}^k$. where m and n are the lengths of sentences. k is the embedding dimension, i and j index the position of words in sentences, s indicates that the embeddings are learned based on semantic. The semantic embedding of sentence p and h are,

$$a_i^{\mathrm{s}} = \text{BiLSTM}(\text{GloVe}(p_i)), i = 1, \ldots, m, \tag{12}$$

$$b_j^{\mathrm{s}} = \text{BiLSTM}(\text{GloVe}(h_j)), j = 1, \ldots, n. \tag{13}$$

3.2.2 Local Inference

The computing of local inference information between two sentences is based on their semantic embeddings a^{s} and b^{s}. No external knowledge is concerned in this part.

First, a soft alignment layer is employed to compute the similarity between words. For premise embedding a_i^{s} and hypothesis embedding b_j^{s}, their similarity is

$$E_{ij} = (a_i^{\mathrm{s}})^T b_j^{\mathrm{s}}, \tag{14}$$

all those E_{ij} form the co-attention matrix $E \in \mathbb{R}^{m \times n}$.

Next, we compute the local relevance information according to the co-attention E. For premise, the relevant semantics in hypothesis is encoded into a context vector a^{c} based on co-attention matrix E and hypothesis semantic embedding b. The context vector b^{c} that encoded relevant semantics in premise can be calculated in the same way,

$$a_i^{\mathrm{c}} = \sum_{j=1}^{n} \frac{\exp\left(E_{ij}\right)}{\sum_{k=1}^{n} \exp\left(E_{ik}\right)} b_j^{\mathrm{s}}, \quad i = 1, \ldots, m, \tag{15}$$

$$b_j^{\mathrm{c}} = \sum_{i=1}^{m} \frac{\exp\left(E_{ij}\right)}{\sum_{k=1}^{m} \exp\left(E_{kj}\right)} a_i^{\mathrm{s}}, \quad j = 1, \ldots, n. \tag{16}$$

Local inference information is then enhanced by computing difference and element-wise product for $(a^{\mathrm{s}}, a^{\mathrm{c}})$ and $(b^{\mathrm{s}}, b^{\mathrm{c}})$,

$$a^{\mathrm{m}} = G([a^{\mathrm{s}}; a^{\mathrm{c}}; a^{\mathrm{s}} - a^{\mathrm{c}}; a^{\mathrm{s}} \odot a^{\mathrm{c}}]), \tag{17}$$

$$b^{\mathrm{m}} = G([b^{\mathrm{s}}; b^{\mathrm{c}}; b^{\mathrm{s}} - b^{\mathrm{c}}; b^{\mathrm{s}} \odot b^{\mathrm{c}}]). \tag{18}$$

where a^{m} and b^{m} are enhanced embeddings for p and h, and G is a non-linear function. We set it as a one-layer feed-forward neural network with ReLU as the activation function.

3.2.3 Semantic Composition

A composition layer is employed to learn the types of local inference relationship at sentence-level. The composed vectors a_i^{v}, b_j^{v} for premise and hypothesis are computed by BiLSTM,

$$a_i^{\mathrm{v}} = \text{BiLSTM}(a_i^{\mathrm{m}}), i = 1, \ldots, m, \tag{19}$$

$$b_j^{\mathrm{v}} = \text{BiLSTM}(b_j^{\mathrm{m}}), j = 1, \ldots, n. \tag{20}$$

The resulting vectors a^{v} and b^{v} are fed into the pooling layer which computes both average and max pooling for premise and hypothesis,

$$a_{avg} = \sum_{i=1}^{m} \frac{a_i^{\mathrm{v}}}{m}, \quad a_{max} = \max_{1 \le i \le m} a_i^{\mathrm{v}}, \tag{21}$$

$$b_{avg} = \sum_{j=1}^{n} \frac{b_j^{\mathrm{v}}}{n}, \quad b_{max} = \max_{1 \le j \le n} b_j^{\mathrm{v}}. \tag{22}$$

The semantic relationship v^{s} between p and h is,

$$v^{\mathrm{s}} = G_{\mathrm{s}}([a_{avg}; a_{max}; b_{avg}; b_{max}]), \tag{23}$$

where G_{s} is a feed-forward neural network with ReLU as the activation function.

3.3 Label Prediction

The label prediction layer is designed to determine the overall logical relationship between two sentences. The semantic representation v^{s}, knowledge representation v^{k}, and the correlation between them based on the element-wise product, are all combined and transformed based on a multilayer perceptron. The perceptron classifier predict a label y to be *entailment*, *contradiction*, or *neutral*,

$$y = \text{MLP}\left(\left[v^{\mathrm{s}};\ v^{\mathrm{k}};\ v^{\mathrm{s}} \odot v^{\mathrm{k}}\right]\right). \tag{24}$$

4 Experiments

4.1 Datasets

We use four widely adopted standard benchmarks: SNLI, MultiNLI, SciTail, and BNLI.

- **SNLI** Stanford Natural Language Inference (Bowman et al., 2015) is extracted from Flickr30k corpus. It is the largest corpus for NLI tasks, with more than 570k human annotated sentence pairs.
- **MultiNLI** Multi-Genre Natural Language Inference (Williams et al., 2018) is also a large-scale corpus containing 433k sentence pairs. In MultiNLI, the development/test sets whose genres appear in training set are referred to as "matched" dataset, and "mismatched" otherwise.
- **SciTail** is a small-scale dataset constructed from multiple-choice science exams and web sentences (Khot et al., 2018). It contains 24k sentence pairs and only classifies sentences into two relationships: *entailment* and *neutral*. SciTail is a difficult benchmark for NLI (Tay et al., 2018).
- **BNLI** is a dataset constructed based on SNLI (Glockner et al., 2018). In BNLI, the premises are taken from the SNLI training set, and hypotheses are generated by replacing a single word within the premise by a different word. Though much simpler and smaller than the SNLI dataset, the performance on BNLI is substantially worse across models trained on SNLI (Glockner et al., 2018).

4.2 Implementation Details

We use Concept Graph (Cheng et al., 2015; Wu et al., 2012) as the external knowledge graph, as it has the largest coverage on datasets we used in this paper. We limit the path length and the number of paths both to 10. We intialize word embeddings by GloVe (Pennington et al., 2014) and entity embeddings by TransE(Bordes et al., 2013). Dimensions of embeddings are all set to be 300. Dropout is set between layers to avoid overfitting with rate 0.5. The optimizer is Adam with batch size 32 and learning rate 0.0004. We use early stopping according to the per-epoch accuracy on the validation set.

To extract subjects, predicates, and objects of sentences, we employ their syntax trees (Rusu et al., 2007). Some datasets provide hand-annotated syntax trees, while for others, we use StandfordNLP (Qi et al., 2018) to generate their syntax trees. For subject, we employ breadth first search to select the first descendent of NP that is a noun. Subjects are selected from entities labeled as NN, NNP, NNPS, or NNS. The deepest verb descendent of the VP subtree is considered as predicates. Predicates are chosen from verbs labeled as VB, VBD, VBG, VBN, VBP, or VBZ. Objects are found in three different subtrees, PP, NP, and ADJP, which are siblings of the VP subtree containing the predicate. In both NP and PP we search for the first noun, while in ADJP we just treat the first adjective to be an object. Finally, we stem the subjects, predicates, and objects to match entities in the knowledge graph.

4.3 Results

Table 1 shows the results on the benchmark SNLI. We do not consider ensemble models in this paper. Our model gets the best result. According to (Glockner et al., 2018), inference on SNLI dataset may not require much knowledge, thus results are not affected significantly by external knowledge. This explains why our model generates similar results with baselines.

Table 1: Performance of models on SNLI.

Model	LSTM Att.	Match-LSTM	LSTMN	BiMPM	ESIM	KIM	CAFE	KGNLI
Test	83.5	86.1	86.3	87.5	88.0	88.6	88.5	**88.9**

Performance on dataset MultiNLI is similar to the performance on SNLI due to the same reason with SNLI as explained in (Glockner et al., 2018). Though the proposed model does not precede much than baselines, it also achieves the best results among all the models, according to Table 2.

Table 2: Performance of models on MultiNLI.

Model	CBOW	BiLSTM	DiSAN	Gated BiLSTM	ESIM	KIM	CAFE	KGNLI
Matched	65.2	69.8	71.0	73.5	76.8	77.2	78.7	**79.1**
MisMatched	64.8	69.4	71.4	73.6	75.8	76.4	77.9	**78.2**

The performance on dataset SciTail is shown in Table 3. Our model achieves the state-of-the-art result with large margin of improvement against ESIM which only considers semantic knowledge. As SciTail consists of more factual sentences than SNLI and MultiNLI datasets (Tay et al., 2018), background knowledge plays a more important role on the inference. The result shows that the proposed model KGNLI can capture external knowledge and use them effectively.

Table 3: Performance of models on SciTail.

Model	Test
Majority	60.3
Ngram	70.6
ESIM (Chen et al., 2017)	70.6
DGEM (Khot et al., 2018)	77.3
CAFE (Tay et al., 2018)	83.3
KGNLI	**84.3**

Dataset BNLI is employed to test the knowledge usage. For a sentence pair in BNLI, only one word of hypothesis h is different with premise p. As a result, BNLI is highly biased towards *contradiction* relation. In practice, BNLI is used as test set, while training on SNLI, MultiNLI, and SciTail. The performance difference between BNLI and SNLI as test set is denoted as Δ. As shown in Table 4, under this setting, the result of our model, denoted as "original setting", is similar to that of ESIM. This is because the subjects, predicates, and objects are almost the same for a sentence pair in BNLI.

Table 4: Performance of models on BNLI.

Model	Train set	SNLI test set	BNLI	Δ
ESIM (Chen et al., 2017)	SNLI	87.9	65.6	-22.3
	MultiNLI + SNLI	86.3	74.9	-11.4
	SciTail + SNLI	88.3	67.7	-20.6
Residual-Stacked-Encoder (Nie and Bansal, 2017)	SNLI	86.0	62.2	-23.8
	MultiNLI + SNLI	84.6	68.2	-16.8
	SciTail + SNLI	85.0	60.1	-24.9
KIM (Chen et al., 2018)	SNLI	88.6	83.5	-5.1
KGNLI (original setting)	SNLI	88.9	67.1	-21.8
	MultiNLI + SNLI	88.1	77.0	-11.1
	SciTail + SNLI	88.5	69.7	-18.8
KGNLI (unique-word setting)	SNLI	88.0	84.1	-3.9
	MultiNLI + SNLI	87.2	**84.7**	**-2.5**
	SciTail + SNLI	87.4	82.1	-5.3

In order to test the performance of our model, we conduct a new experiment under another setting, named "unique-word setting". For dataset BNLI, the two different words of sentence pair (p, h) are

extracted. An example is given in Table 5. We treat these words as key words, remove the knowledge composition layer, and directly set the composed vector as the relation vector of these key words. For other datasets, we choose keywords among subjects, predicates, and objects. As indicated by Table 4, our model achieves the best performance among all the models. This experiment also shows that the proposed model can capture the background knowledge and utilize it to improve model performance.

Table 5: Examples from BNLI dataset.

sentences	**p**: The man rides bicycle up the **brick** wall. **h**: The man rides bicycle up the **cement** wall.
original	**p**: *s*: man, *p*: ride, *o*: bicycle **h**: *s*: man, *p*: ride, *o*: bicycle
unique	**p**: brick **h**: cement

4.4 Ablation Study

Table 6 shows the results of ablation study on SciTail dataset. In the experiment, parts of sentences are removed. The results partially validate that parts of sentences are crucial in NLI related task.

Table 6: Ablation test on SciTail

Components used	Test
Only subject	82.4
Only predicate	81.7
Only object	82.0
subject & predicate	83.2
object & predicate	83.1
subject & object	83.1
subject & predicate & object	**84.3**

5 Conclusion and Future work

This paper proposed a knowledge enhanced NLI model based on knowledge graphs, which introduces background knowledge into NLI model. For a sentence pair, the proposed model learns a knowledge-relation representation based on paths of knowledge graph and a semantic-relation representation through BiLSTM. These two representations are then merged by a feed-forward neural network to predict the relationship label. Experimental results validated the effectiveness of the proposed model. As to the future work, we aim to find out how to decide the keywords in sentence pairs that determine their relationship.

Acknowledgment

This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0103405, the National Natural Science Foundation of China under Grants 71902179 and 71621002. Linjing Li is the corresponding author.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. Curran Associates, Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *ACL*. ACL, July.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *ACL*, pages 2406–2417, Melbourne, Australia, July. Association for Computational Linguistics.

Jianpeng Cheng, Zhongyuan Wang, Ji-Rong Wen, Jun Yan, and Zheng Chen. 2015. Contextual text understanding in distributional semantic space. In *CIKM*, October.

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 *NAACL*, pages 1460–1469, New Orleans, Louisiana, June. Association for Computational Linguistics.

Oren Glickman and Ido Dagan. 2005. Web based probabilistic textual entailment. In *In Proceedings of the 1st Pascal Challenge Workshop*, pages 33–36.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *ACL*, pages 650–655, Melbourne, Australia, July. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI*.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *ACL*, pages 5962–5971. Association for Computational Linguistics, July.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *COLING*, pages 521–528, Manchester, UK, August. Coling 2008 Organizing Committee.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 41–45, Copenhagen, Denmark, September. Association for Computational Linguistics.

Boyuan Pan, Yazheng Yang, Zhou Zhao, Yueting Zhuang, Deng Cai, and Xiaofei He. 2018. Discourse marker augmented network with reinforcement learning for natural language inference. In *ACL*, pages 989–999, Melbourne, Australia, July. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *CoNLL*, pages 160–170. Association for Computational Linguistics, October.

Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. 2007. Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference Information Society-IS*.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *EMNLP*, pages 1565–1575, Brussels, Belgium, October-November. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*. Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, May.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *ACL*, pages 4258–4264, Florence, Italy, July. Association for Computational Linguistics.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *ACL*. Association for Computational Linguistics, July.