Syntax-aware Natural Language Inference with Graph Matching Network

Yan-Tong, Lin
Department of Computer Science
National Chiao Tung University
Hsinchu City, Taiwan
0312fs3@gmail.com

Meng-Tse, Wu
Institute of Information Science
Academia Sinica
Taipei City, Taiwan
moju@iis.sinica.edu.tw

Keh-Yih, Su
Institute of Information Science
Academia Sinica
Taipei City, Taiwan
kysu@iis.sinica.edu.tw

***Abstract*—The task of entailment judgment aims to determine whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given a premise. While previous methods strike successful in several benchmarks and even exceed the human baseline, recent researches show that it remains arguable if the methods learn statistical bias in the datasets. In this paper, we propose the syntax-aware NLI (SynNLI) model, which utilizes graph matching networks to obtain syntax-guided contextualized representation while aligning the premise and hypothesis accordingly. We show that the proposed method outperforms multiple baseline models on MNLI develop set, and visualize the model internal behavior.**

***Index Terms*—graph neural networks, recognize textual entailment, natural language inference, dependency tree**

I. Introduction

Entailment judgment is arguably one of the most fundamental language understanding tasks. It aims to determine inferential relationship (i.e., entailment, contradiction, or neutral) between a premise and a hypothesis. The task is considered an important mission in natural language understanding (NLU) because of the wide range of downstream applications such as question answering, and the ability for testing the understanding of natural language. With the release of large datasets like SNLI [1] and MultiNLI [2], many neural network approaches have been proposed in recent years. Many works [3]–[12] adopt complex neural architectures that decompose the task into subtasks (e.g., encoding, matching, etc.), and most of them adopt attention-based matching as a part of their modules. However, previous approaches fail to generate a precise token alignment among premises and hypotheses. The strategies they adopt to improve alignment accuracy are primarily generating better token representations; however, humans conduct alignment not only base on word sense similarity but also base on structure analogy [13]. For example, consider the premise and hypothesis pair "Allen travels from America to China" and "Allen travels from China to America", a human can easily tell that this is a contradiction, but according to our probing experiments, both Decomp-Att [3] and RoBERTa [14] give entailment predictions. A similar probing shows that current models struggle to handle negation when there is high lexical overlap as well, for example: "Alice does not like to eat pizza. Chris likes to eat pizza. Bob likes to eat pizza" and "Alice likes to eat pizza. Chris likes to eat pizza. Bob likes to eat pizza". To address the above-mentioned limitation, we explore the possibilities to utilize heterogeneous graph matching networks, which consider both structure analogy and word sense similarity, to encode and align premise and hypothesis. To the best of our knowledge, this is the first work of introducing graph matching methods to the task of entailment.

This work was done during his 2020 summer internship in Institute of Information Science, Academia Sinica.

Our contributions are 4-fold:

- We propose a novel model utilizing graph matching networks for the task of entailment judgment.
- Experimental results show that our model outperforms several baseline models.
- We visualize the model internals and provide interpretations.
- Through the experiments, we point out the remaining problems for future study.

II. Related Work

Natural language inference has been studied for many years. Early approaches focused on rule-based or statistical methods using hand-craft features [15].

With the release of large annotated corpora [1], [2], a lot of deep learning methods are proposed. We divide previous works into classes: sentence encoding based, sentence matching based, knowledge incorporation, representation improvement, etc. based on their architectures or how they improve.

Sentence encoding based methods [9]–[12], [16] adopt Siamese architecture [17] that maps each sentence into a vector space and do comparison afterwards.

Sentence matching based methods [3]–[8] consider cross sentence information and often outperform sentence encoding based models. This shows that lower-level interaction between sentence pairs is indispensable, and motivates us to adopt graph matching network [18] on the task of NLI.

External knowledge plays an important role in NLI. In recent years, works that try to incorporate organized knowledge bases (e.g., knowledge graphs (KG)) [19]–[23] are proposed. They insert external knowledge at different levels. KGA-Net

[22] concatenates word embedding with knowledge embedding. KIM [19] improves ESIM by modifying the attention mechanism to take KG into account. KCI-TEN [20], ConseqNet [21], and KES [23] tries to improve the performance of NLI models by providing knowledge background to the classifier.

In 2018, the pretrained transformer masked language model known as BERT [24] stroke successful in various fields of NLP, including NLI. Under the hood it utilizes intense self and cross multi-head attention along with the transformer block structure to achieve the amazing results.

There are many attempts to improve BERT [14], [25]–[29], some of them focus on the pretraining or training step of the transformer encoder. For example, RoBERTa [14] enlarges the corpora and batch size and achieved state-of-the-art on the GLUE benchmark [30] for natural language understanding, MT-DNN [28] presents that multi-task learning with shared parameters can boost the performance of the model on the end tasks, and T5 [29] converts all text-based language problems into a text-to-text format and uses them to train a universal encoder-decoder model.

Despite the high accuracy the models achieved, analyses [31]–[37] show that current NLI models tend to learn surface features, adopt false heuristics [38], and are vulnerable to adversarial attacks [39]. Also, the computational resource required to pretrain masked language models with a huge parameter size is inaccessible to the general public.

Recent researches show that graph neural networks [40]–[42] are successful in modeling graph-structured data. And there are works [43]–[46] taking graph neural networks to various domains of NLP.

In this work, inspired by the success of sentence matching models and graph neural networks, we explore the possibilities to apply graph matching networks to NLI.

III. Method

The proposed model is composed of 4 layers: (1) input layer, (2) contextualized embedding layer, (3) syntax enhance matching layer, and (4) output layer. In the following sections, we will describe the architecture layer by layer. Fig. 1 illustrates a high-level view of SynNLI.

A. Input Layer

Our model accepts two sentences and their dependency graphs as input, i.e., P (premise), G_p and H (hypothesis), G_h, in which the G_p and G_h are generated by an open-source dependency parser provided by Stanza [47]. The Stanza parser tokenizes the input text and decomposes the document into sentences, where each sentence is a dependency tree with tokens in the sentence. To introduce a constituent prior that nearby words tend to relate to each other, we connect the adjacent word pairs by adding special "const:next" and "const:prev" edges between them. An example is shown in Fig. 2.

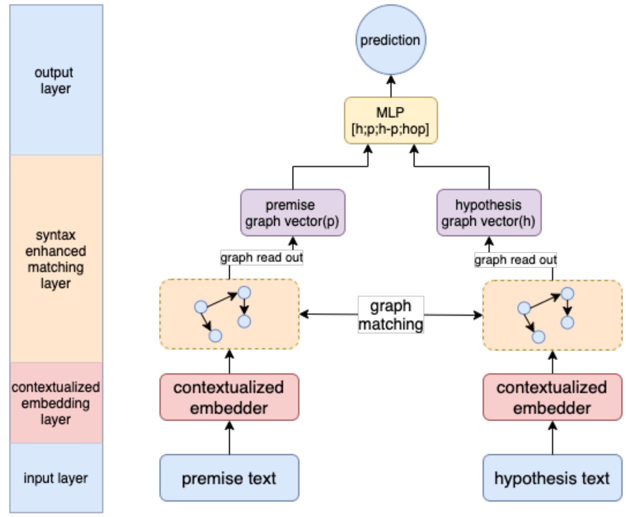


Fig. 1. Overall model architecture. Input hypothesis and premise first get their contextualized word embeddings by a contextualized embedder and is applied with a heterogeneous graph matching network to get graph embeddings of the premise graph and hypothesis graph. Then an MLP makes the final judgment by the two graph embedding vectors.

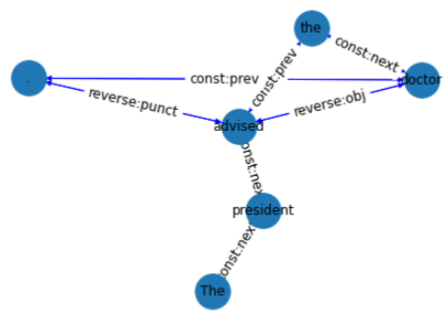


Fig. 2. An example of the input dependency graph.

B. Contextualized Embedding Layer

This layer aims to learn a d-dimensional contextualized word representation for each token (tokenized by the dependency parser) in P and H. We use ELMO [48] as the contextualized embedder. The outputs of the layer are two sequences of d-dimensional vectors $w_h, w_p \in \mathbb{R}^{d\times n}, \mathbb{R}^{d\times m}$, where n, m are the sequence length of hypothesis and premise respectively.

C. Syntax Enhance Matching Layer

In the syntax enhance matching layer, we view each token in premise and hypothesis as a node in the dependency graphs. Then we strengthen the syntactic knowledge of representation by updating each node's representation with its neighbors. At the same time, we do cross attention at each layer to obtain cross-sentence information in various granularity [18]. Finally, a graph readout is performed to aggregate the information from P and G to two fix-sized vectors.

More formally, the output of previous layer are tensors $w_h, w_p \in \mathbb{R}^{d\times n}, \mathbb{R}^{d\times m}$. They pass through L layers of graph matching network similar to [18], which is explained in the following sections.

1) Graph Matching Net: Let $h_i^{(l)} \in \mathbb{R}^{d\times 1}$ be the representation of node i in the l-th layer of the graph network. We have the input in this layer $h^0 = w_p, w_h$, and the last layer is h^L, which is then aggregated to fix-sized vectors g_p, g_h as the output of this layer.

For all $l \in \{0, 1, ..., L-1\}$, h^{l+1} is obtained by the following equations, (4) from h^l, where each node i in the graphs is updated with messages from its neighbors ($m_{j\to i}$) and matched to all the nodes in the other graph ($\mu_{j\to i}$).

$$m_{j\to i} = f_{message}(h_i^l, h_j^l, \phi(i), \phi(j), \tau(e_{i,j})) \tag{1}$$

$$\mu_{j\to i} = f_{match}(h_i^l, h_j^l)\ \forall i \in G_p, j \in G_h \tag{2}$$

$$h_i^{l+1} = f_{node}(h_i^l, \sum_{j\in N(i)} m_{j\to i}, \sum_{j'\in G'} \mu_{j'\to i}) \tag{3}$$

$$g = f_G(\{h_i^L | \forall i \in G\}) \tag{4}$$

Here ϕ, τ are mapping from id to type for nodes and edges respectively, note that we model the dependencies as directed graphs, i.e. $\tau(e_{u,v}) \neq \tau(e_{v,u})$.

We now discuss the choice of functions in the proposed model.

2) Message Passing: To model different dependency relations, we adopt relational graph convolution network [49].

$$m_{j\to i} = \mathbf{\Theta}_{\tau(e_{i,j})} \cdot h_j \tag{5}$$

where $\mathbf{\Theta}_{\tau(e_{i,j})}$ is trainable parameter.

3) Graph Matching: An attention-based matching is applied to get cross sentence information from the other graph. Intuitively, $\sum \mu_{j\to i}$ measures the difference between h_i and its closest neighbor in the other graph.

$$\mu_{j\to i} = a_{j\to i}(h_i - h_j) \tag{6}$$

$$a_{j\to i} = \text{softmax}_j(s(h_i, h_j)) \tag{7}$$

In (7), s is a similarity function. We choose cosine similarity for the proposed model.

4) Node update: To update the representation from the aggregated message from graph message passing and graph matching, we have:

$$f_{node} = \text{GRU} \tag{8}$$

5) Graph Readout: Then we readout the graph embeddings for G_P and G_H using the aggregation module proposed in [50],

$$g_p = \text{MLP}_G\left(\sum_{v\in G_p} \sigma(\text{MLP}_{gate}(h_v^L)) \odot \text{MLP}_{value}(h_v^L)\right) \tag{9}$$

$$g_h = \text{MLP}_G\left(\sum_{v\in G_h} \sigma(\text{MLP}_{gate}(h_v^L)) \odot \text{MLP}_{value}(h_v^L)\right) \tag{10}$$

where MLP denotes multilayer perceptron.

D. Output Layer

With the graph embeddings g_p and g_h we can get the final prediction by the vector comparisons and a feed forward network.

$$y = \text{MLP}(w_{[CLS]}; g_p; g_h; g_p - g_h; g_p \odot g_h) \tag{11}$$

IV. Experiments

A. Data

To show the effectiveness of the proposed model, we evaluate on the MultiNLI dataset [2] and HANS dataset [38].

1) MultiNLI: The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information. The corpus is similar to the SNLI [1] corpus, but it covers a wider range of genres of spoken and written text, and supports cross-genre generalization evaluation. MultiNLI has become a popular choice for evaluating the performance of NLI models since its release.

2) HANS: While MultiNLI being one of the default choices for evaluating the performance of NLI models, recent researches ([32], [33], [38]) show that models trained on MultiNLI are lack of compositional knowledge of the language. In [38], the authors developed the Heuristic Analysis for NLI Systems (HANS) dataset to determine whether statistical NLI models adopt fallible syntactic heuristics: the lexical overlap heuristic, the subsequence heuristic, and the constituent heuristic. The dataset is adopted to test whether the proposed model adopts the same untrue heuristics.

B. Implementation Details

We use Pytorch [51], AllenNLP [52], and Pytorch Geometric [53] in our model implementation. The source code is available at https://github.com/EazyReal/2020-IIS-internship.

Since the dimension of the graph matching network is set to 300 in the experiment, to make compatible the output of the contextualized embedder and the input of the graph matching network, we use a simple linear projection layer. The number of layers L is set to 3.

We explored some different choices of modules, including BiMPM with 10 perspectives [8] for graph matching and

CGConv [54] for message passing, but no substantial improvement is observed.

We train our model on MultiNLI [2] dataset for 8 epochs with an AdamW optimizer provided in [55]. The learning rate is set to 0.0005 and the weight decay is set to 0.01.

C. Results

1) MultiNLI: The performance comparison (accuracy) on MultiNLI dataset [2] is shown in Tab. I. Since pretrained transformer models [14], [24]–[27] have an enormous amount of parameters and are thus computational expensive in terms of both memory and training/evaluation time, we do not adopt them as baseline models in this paper. We show that our proposed method outperforms multiple baseline models.

TABLE I
PERFORMANCE COMPARISON (ACCURACY) ON MULTINLI.

Model/Accuracy Model	**MultiNLI**	
	Matched	***Mismatched***
Majority	36.5	35.6
CBOW*	64.8	64.5
BiLSTM*	66.9	66.9
Di-SAN [10]	71.0	71.4
Gated Att BiLSTM# [9]	73.2	73.6
HBMP [12]	73.7	73.0
ESIM† [4]	76.8	75.8
KIM(ESIM+WordNet) [19]	77.2	76.4
SynNLI(proposed)	77.4	77.5
BERT base [24]	84.6	83.4

[a] If not specified, the accuracy is reported by the original paper. Models with *, †, # are reported from [2], [7], [19] respectively.

2) HANS: The performance comparison on HANS [38] dataset (models trained on MultiNLI) is shown in Tab. II. While adopting strong inductive bias by propagating information alone dependencies in the proposed model, the experiment result shows that our model still performs poorly on the HANS development set. However, when we trained the proposed model on the HANS training set, the validation accuracy is close to 100%, emphasizing the weakness of MultiNLI [2] training data revealed in related studies [32], [33], [38].

TABLE II
PERFORMANCE COMPARISON (ACCURACY) ON HANS DATASET.THE NEUTRAL AND CONTRADICTION LABELS ARE MERGED INTO A SINGLE LABEL, NON-ENTAILMENT. THE ACCURACY EXCEPT FOR THE PROPOSED MODEL IS REPORTED BY THE ORIGINAL PAPER [38].

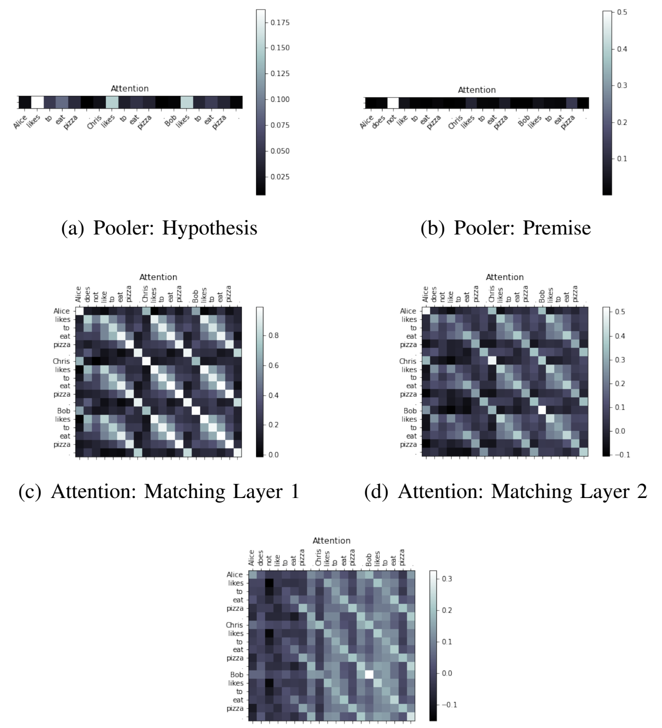
Model	**Correct: Entailment**			**Correct: Non-entailment**		
	Lexical	***Subseq.***	***Const.***	***Lexical***	***Subseq.***	***Const.***
Decomp-Att [3]	1.00	1.00	0.98	0.00	0.00	0.03
ESIM [4]	0.99	1.00	1.00	0.00	0.01	0.00
SPINN [11]	0.94	0.96	0.93	0.06	0.14	0.11
BERT [24]	0.98	1.00	0.99	0.04	0.02	0.20
Proposed	0.97	0.99	0.98	0.02	0.06	0.03

D. Model Interpretation by Attention Visualization

To provide an interpretation of the proposed model, we visualize the attention in the graph readout layers and the graph matching components. The result is shown in Fig. 3 and Fig. 4.

In Fig. 3, we can see by the attention score in the last matching layer that the representation of ”like” in the subsequence ”Alice does not like...” is different from the one in the subsequence ”Alice likes...”. We can also see that poolers (i.e. graph readout module) view ”not” and ”like” as important words.

For Fig. 4, in the first layer, our model match ”China” to ”China”, ”America” to ”America”; however, after message passing, the similarity between the ”China”s and the ”America”s in the sentence pair is lower since they have different roles in their dependency trees.



(a) Pooler: Hypothesis (b) Pooler: Premise

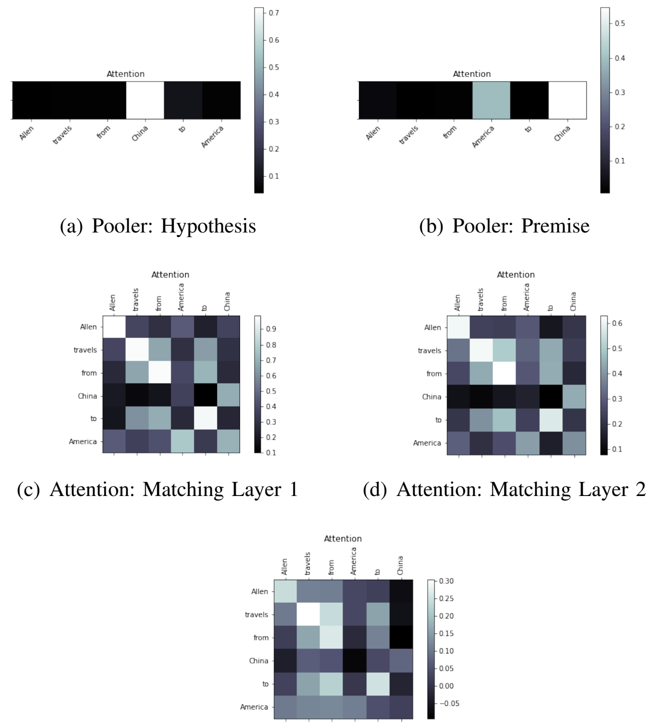
(c) Attention: Matching Layer 1 (d) Attention: Matching Layer 2

(e) Attention: Matching Layer 3

Fig. 3. An example of the proposed model solving a long sequence with partial negation instance that RoBERTa [14] fails. The premise is ”Alice does not like to eat pizza. Chris likes to eat pizza. Bob likes to eat pizza”, the hypothesis is ”Alice likes to eat pizza. Chris likes to eat pizza. Bob likes to eat pizza”, and the gold label is ”contradiction”.

V. CONCLUSION

In this work, we propose the syntax-aware NLI (SynNLI) model. Different from previous NLI models, we utilize graph matching network and view tokens as nodes and do encoding and matching according to dependency relations. We evaluated our model on the MultiNLI and the HANS datasets. Experimental results show that our model achieves competitive results and outperforms several baselines on MultiNLI. We



(a) Pooler: Hypothesis

(b) Pooler: Premise

(c) Attention: Matching Layer 1

(d) Attention: Matching Layer 2

(e) Attention: Matching Layer 3

Fig. 4. An example of the proposed model solving a lexical swap instance. The premise is ”Allen travels from America to China”, the hypothesis is ”Allen travels from China to America”, and the gold label is ”contradiction”.

also provide visualizations of the model internals for interpretation. However, though adopting strong inductive bias by propagating messages alone dependencies, the proposed model still fails to generalize to the HANS dataset when trained on MultiNLI corpora, calling for the need for quality datasets for NLP research.

VI. Future directions

- Due to the limited time, we did not do a thorough parameter search for the proposed method. Different hyperparameters and model settings (e.g., message passing module, graph matching module, dimensions, number of layers) can be explored for better performance.
- The experiment result on HANS emphasizes the need for quality datasets for NLP research.
- We would like to discuss the feasibility of a variant of graph matching network for NLP, which generates embeddings that are purely dependent on the structural information of nodes, without considering word sense. This can be achieved by using POS or SRL embeddings for nodes, dependency relation embeddings, and crystal graph convolution [54]. And we can use the similarity between the embeddings to gate the cross attention between tokens in different sentences.

References

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

[2] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

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

[4] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 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, July 2017. Association for Computational Linguistics.

[5] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*, 2017.

[6] Wenpeng Yin, Hinrich Schütze, and Dan Roth. End-task oriented textual entailment via deep explorations of inter-sentence interactions. *arXiv preprint arXiv:1804.08813*, 2018.

[7] Yichen Gong, Heng Luo, and Jian Zhang. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*, 2017.

[8] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*, 2017.

[9] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Recurrent neural network-based sentence encoder with gated attention for natural language inference. *arXiv preprint arXiv:1708.01353*, 2017.

[10] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. *arXiv preprint arXiv:1709.04696*, 2017.

[11] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany, August 2016. Association for Computational Linguistics.

[12] Aarne Talman, Anssi Yli-Jyrä, and Jörg Tiedemann. Sentence embeddings in NLI with iterative refinement encoders. *Natural Language Engineering*, 25(4):467–482, 2019.

[13] Dedre Gentner and Arthur B Markman. Structure mapping in analogy and similarity. *American psychologist*, 52(1):45, 1997.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[15] Michael Heilman and Noah A Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019, 2010.

[16] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[17] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a” siamese” time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[18] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. *arXiv preprint arXiv:1904.12787*, 2019.

[19] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. Neural natural language inference models enhanced with external knowledge. *arXiv preprint arXiv:1711.04289*, 2017.

[20] Daoyuan Chen, Yaliang Li, Min Yang, Hai-Tao Zheng, and Ying Shen. Knowledge-aware textual entailment with graph attention network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2145–2148, 2019.

[21] Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. Improving natural language inference using external knowledge in the science questions domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7208–7215, 2019.

[22] Meina Song, Wen Zhao, and E HaiHong. Kganet: a knowledge graph attention network for enhancing natural language inference. *Neural Computing and Applications*, pages 1–11, 2020.

[23] Pavan Kapanipathi, Veronika Thost, Siva Sankalp Patel, Spencer Whitehead, Ibrahim Abdelaziz, Avinash Balakrishnan, Maria Chang, Kshitij P Fadnis, R Chulaka Gunasekara, Bassem Makni, et al. Infusing knowledge into the textual entailment task using graph convolutional networks. In *AAAI*, pages 8074–8081, 2020.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[25] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[26] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[27] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[28] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[30] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[31] Arindam Mitra, Ishan Shrivastava, and Chitta Baral. Enhancing natural language inference using new and expanded training data sets and new learning models. In *AAAI*, 2020.

[32] Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*, 2018.

[33] Yixin Nie, Yicheng Wang, and Mohit Bansal. Analyzing compositionality-sensitivity of nli models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874, 2019.

[34] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. Help: A dataset for identifying shortcomings of neural models in monotonicity reasoning. *arXiv preprint arXiv:1904.12166*, 2019.

[35] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. Can neural networks understand monotonicity reasoning? *arXiv preprint arXiv:1906.06448*, 2019.

[36] Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. Stress-testing neural models of natural language inference with multiply-quantified sentences. *arXiv preprint arXiv:1810.13033*, 2018.

[37] Aarne Talman and Stergios Chatzikyriakidis. Testing the generalization power of neural network models across nli benchmarks. *arXiv preprint arXiv:1810.09774*, 2018.

[38] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[39] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2, 2019.

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

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

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

[43] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*, 2018.

[44] Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, 2019.

[45] Binxuan Huang and Kathleen M Carley. Syntax-aware aspect level sentiment classification with graph attention networks. *arXiv preprint arXiv:1909.02606*, 2019.

[46] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. Relational graph attention network for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.12362*, 2020.

[47] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*, 2020.

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

[49] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[50] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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

[53] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[54] Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018.

[55] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.