# Enhancing Generalization in Natural Language Inference by Syntax

**Qi He**[1]*, **Han Wang**[2]*, **Yue Zhang**[3,4]†
[1] University of Electronic Science and Technology of China, Chengdu, China
[2] New York University, New York, USA
[3] School of Engineering, Westlake University, Hangzhou, China
[4] Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, China
heqi@uestc.edu.cn, hwang@nyu.edu, zhangyue@westlake.edu.cn

## Abstract

Pre-trained language models such as BERT have achieved the state-of-the-art performance on natural language inference (NLI). However, it has been shown that such models can be tricked by variations of surface patterns such as syntax. We investigate the use of dependency trees to enhance the generalization of BERT in the NLI task, leveraging on a graph convolutional network to represent a syntax-based matching graph with heterogeneous matching patterns. Experimental results show that, our syntax-based method largely enhance generalization of BERT on a test set where the sentence pair has high lexical overlap but diverse syntactic structures, and do not degrade performance on the standard test set. In other words, the proposed method makes BERT more robust on syntactic changes.

## 1 Introduction

The task of natural language inference (NLI) targets at determining whether one sentence entails another (Condoravdi et al., 2003). Recently, large-scale pre-trained contextualized embeddings such as BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019) have given the state-of-the-art accuracy for this task. It has been shown that pre-trained models help to better capture heuristic patterns in a set of training data and therefore enhance in-domain performance (Wang et al., 2018). However, there are still limitations on the generalization of such models to examples under a different distribution. In particular, it has been shown that seemingly simple types of examples in a carefully designed evaluation set (i.e. HANS) can lead to significant degeneration and large variability in performance (McCoy et al., 2019b,a). Table 1 shows a set of test cases from HANS, where premise and hypothesis have high lexical overlap but different syntactic structures. The BERT model gives incorrect results on most cases. This issue can negatively affect NLI applications such as dialogue (Dziri et al., 2019; Welleck et al., 2019).

It has been shown that syntactic structures are useful for cross-domain generalization of NLP models (Wang et al., 2017; Strubell and McCallum, 2018). Intuitively, a more robust NLI model can be obtained by making use of structural information. We empirically investigate the effectiveness of syntactic features for enhancing the generalization of BERT-based matching models. In particular, given a pair of sentences, the dependency syntax of each sentence is obtained using a neural parser (Qi et al., 2018). The parse trees are then extended using four types of edge patterns, including a soft co-attention matching pattern that links the sentence pair into an integrated graph. A graph convolutional network (GCN) (Kipf and Welling, 2016) is used to represent the whole matching graph structure.

Experiments show that the performance of the proposed model is much better than BERT and other syntax-based baselines on the category in HANS where the premise and non-entailment hypothesis have high lexical overlap but different syntactic structures, when both models are trained on MNLI dataset. It proves that incorporating syntax by the proposed method enhances generalization of the BERT model on syntactic changes[‡].

## 2 Related Work

There has been much work based on deep neural networks for the NLI task. One straight-forward solution is to independently encode the premise

*Equal Contribution. Work is done when working at Westlake University.
†Corresponding author.
‡Our code will be available at: https://github.com/heqi2015/CA_GCN



*Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4973–4978
November 16 - 20, 2020. 

| Premise | Hypothesis | Gold | BERT-CLS | BERT+CAGCN |
|---|---|---|---|---|
| The student saw the managers. | The managers saw the student. | **N** | E | **N** |
| The judge in front of the manager saw the doctors. | The doctors saw the judge. | **N** | E | **N** |
| The bankers admired the lawyer that the students supported. | The lawyer admired the students. | **N** | E | **N** |
| The secretary and the managers saw the actor. | The secretary saw the managers. | **N** | E | **N** |
| The manager was introduced by the professor. | The manager introduced the professor. | **N** | E | E |

Table 1: Examples drawn from the "non-entailed lexical overlap" category in HANS (McCoy et al., 2019b) for the NLI task. In each example, the words in hypothesis are drawn from the premise but do not form a subsentence of premise, and thus the syntactic structures in hypothesis and premise are quite different. Both the BERT-CLS baseline and our GCN-based BERT model with co-attention links (BERT+CAGCN) are finetuned on MNLI dataset (Williams et al., 2018), and the *neutral* or *contradiction* labels are translated into *non-entailment* when evaluation (McCoy et al., 2019b). Note that *E* stands for *entailment*, and *N* stands for *non-entailment*.

and the hypothesis into embedding vectors, which are fed to a multi-layer neural network for classification (Bowman et al., 2015). It has been shown that alignment between local words in the premise and hypothesis benefits the aggregation of information (Chen et al., 2016; Parikh et al., 2016), and encoding the sentence pair simultaneously can capture more interaction and thus further improve the performance (Devlin et al., 2018). We thus adopt this model as our baseline.

Syntax has been proven beneficial for semantic tasks such as NLI (Bowman et al., 2016; Pang et al., 2019; Lei et al., 2019). Tree-based SPINN methods encode sentences by combining constituency phrases (Bowman et al., 2016). Recently, Pang et al. (2019) proposed to enhance the token representation by using contextual vector representations from a pretrained parser. The GCN method has also been used to represent syntax for sentence matching (Lei et al., 2019), where the syntax of each sentence is encoded separately. In this paper, we use a GCN to encode a whole matching graph with syntactic information, showing that integrating syntax by our method benefits the generalization of BERT-based method.

## 3 Method

The overall architecture of the proposed method is shown in the top of Figure 1. At the bottom layer, contextualized representations of the two sentences are obtained by using BiLSTM, ELMo or BERT. The representation is then fed into GCN to initialize the representations in the first layer.

### 3.1 GCN

The graph structure of each layer in GCN is depicted in the bottom of Figure 1. Each node in the graph represents one word in the sentence pair. We define four types of directed edges in the graph, as described in Equation 1, where $\mathcal{E}$ denotes the set of syntactic dependency arcs inside sentences, and $S(w_i)$ indicates which sentence the word $w_i$ belongs to. The first two edge types are introduced to allow information flow along and against syntactic arcs. Thirdly, the self-loop edge is added for better preserving information of each word across message passing iterations (Kipf and Welling, 2016).

$$E(i,j) = \begin{cases} \text{dependency}, & \text{if } (w_i, w_j) \in \mathcal{E} \\ \text{reversion}, & \text{if } (w_j, w_i) \in \mathcal{E} \\ \text{self-loop}, & \text{if } i == j \\ \text{co-attention}, & \text{if } S(w_i)! = S(w_j) \end{cases} \quad (1)$$

The last type of relation aims to enforce alignment of words between sentences, where the similarity between each word $w_i$ in sentence A and each word $w_j$ in sentence B at the $k$th layer is calculated by the co-attention operation as $C_{i,j}^{(k)} = \sigma(h_i^{(k)T} W_{co}^{(k)} h_j^{(k)})$, where $\sigma$ denotes the sigmoid function, $h$ the feature vector, and $W_{co}$ the affinity weight. The feature of node $i$ is updated at the $k$th layer by $h_i^{(k+1)} = f\Big(\sum_{j \in N(i)} g_{i,j}^{(k)} (W_{E(i,j)}^{(k)} h_j^{(k)} + b_{E(i,j)}^{(k)}\Big)$, where $f(\cdot)$ is ReLU activation function, $N(\cdot)$ is the neighbor set, and $g_{i,j}^{(k)}$ is a gate function that is described below.

Note that we only take unlabelled dependencies into account to avoid over-parameterization (Marcheggiani and Titov, 2017), as shown in Equation 1. By bringing in sparse and unlabeled dependency relations, the embedding of each word is influenced by its immediately semantically or syntactically related words, which leads to a potentially more robust word representation. We apply a gate $g_{i,j}^{(k)}$ to each edge to calculate the importance

of information exchange (Marcheggiani and Titov, 2017).

$$g_{i,j}^{(k)} = \begin{cases} C_{i,j}^{(k)}, & \text{if } E(i,j) \text{ is co-attention;} \\ \sigma(h_i^{(k)T} v_{E(i,j)}^{(k)} + d_{E(i,j)}^{(k)}), & \text{otherwise.} \end{cases} \quad (2)$$

In addition, highway units are adopted in each layer to preserve information in multiple stacked GCN layers (Srivastava et al., 2015).

### 3.2 Co-Attention Layer

We denote the word representations of sentence A and sentence B in the GCN output as $H_A$ and $H_B$, respectively. An affinity matrix is calculated by $C = \tanh(H_A^T W_c H_B)$, which is used to calculate the co-attention maps between the sentence pair (Lu et al., 2016):

$$G_A = \tanh(W_A H_A + C^T(W_B H_B)), \quad (3)$$

$$a_A = \text{softmax}(w_A^T G_A), \quad (4)$$

$$G_B = \tanh(W_B H_B + C(W_A H_A)), \quad (5)$$

$$a_B = \text{softmax}(w_B^T G_B) \quad (6)$$

where $W_A, W_B, w_A, w_B$ are weight parameters, and each element in $a_A$ and $a_B$ is the attention probability of words in sentence A and B, respectively. Finally, the vector representations of the sentences are calculated by

$$h_A = \sum_{w_i \in A} a_A^i H_A^i, \; h_B = \sum_{w_j \in B} a_B^j H_B^j, \quad (7)$$

where $a^i$ denotes the $i$th element in $a$, and $H^i$ the $i$th column in $H$.

### 3.3 Output Classifier

With vector representations of the sentence pair, we obtain an overall representation by concatenating them with their element-wise difference and multiplication as $[h_A, h_B, h_A - h_B, h_A \odot h_B]$, which is fed to a linear layer with softmax activation to obtain the final classification output. The final model is trained using a cross-entropy loss.

## 4 Experiments

**Models for Comparison.** We consider three variants of the proposed model based on BERT, linking words between the premise and the hypothesis at the GCN layer in different ways: co-attention links as described in Section 3

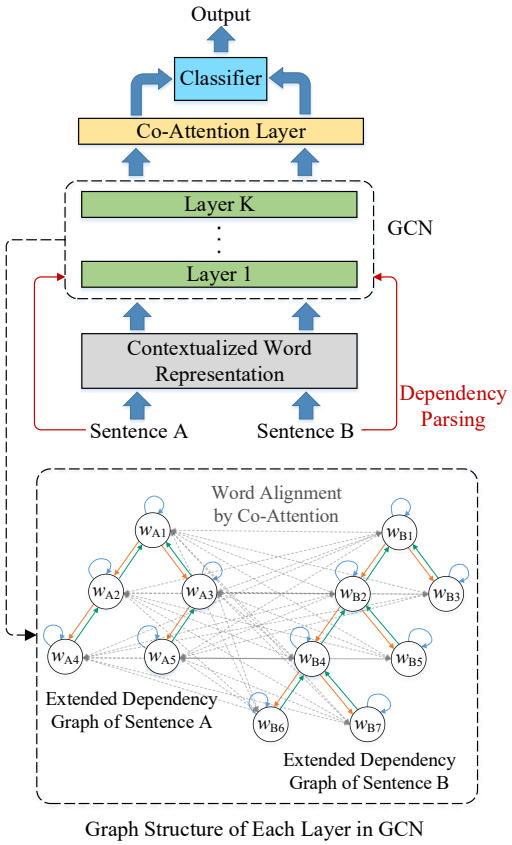


Figure 1: The proposed syntax-based architecture.

(BERT+CAGCN), simply linking the same lemmarized words (BERT+SWGCN), and no links (BERT+SGCN) as in Lei et al. (2019). We also tried combining the outputs of BERT and GCN, which results in little performance improvement. The baseline models include "BERT-CLS", which adds classifier to the vector representation of [CLS] token in BERT model (Devlin et al., 2018), "BERT-Attn", which feeds word output of BERT sequentially to co-attention layer and classifier, "BERT+LF", which adds syntactic features to the input of the classifier layer (Pang et al., 2019), and "SPINN" which encodes sentences with a parse tree (Bowman et al., 2016).

**Datasets and Settings.** We train all the models on MNLI training data (Williams et al., 2018), and evaluate them on MNLI and HANS* (McCoy et al., 2019b). Evaluation examples in MNLI are divided into two categories: in-domain match (MNLI-m) and cross-domain mismatch (MNLI-mm). The evaluation set HANS is designed to

*The labels *neutral* or *contradiction* are translated into *non-entailment* for evaluation on HANS.

| **Model** | S/O | Pre | Rc | Conj | Pass | **Avg.** |
|---|---|---|---|---|---|---|
| BERT-CLS | 24.2 | 51.0 | 37.4 | 53.8 | 0.5 | 33.4 |
| BERT-Attn | 40.9 | 53.3 | 44.6 | 56.4 | 0.3 | 39.1 |
| BERT+LF | 29.8 | 70.6 | 48.1 | 68.2 | 8.1 | 45.0 |
| SPINN | 28.7 | 19.3 | 25.0 | 12.6 | 1.9 | 17.5 |
| BERT+CAGCN | **81.8** | **81.6** | **71.9** | 73.7 | **11.1** | **64.0** |
| BERT+SWGCN | 45.6 | 66.2 | 53.8 | **78.6** | 0.2 | 48.9 |
| BERT+SGCN | 58.5 | 66.3 | 49.0 | 69.6 | 2.7 | 49.2 |

Table 2: Results on the five subcategories of *non-entailed lexical overlap* examples in HANS.

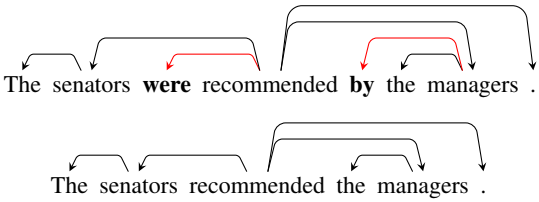


Figure 2: An example from *passive* subcategory in HANS with syntactic dependency edges.

| | | | HANS | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Correct: *Entailment* | | | Correct: *Non-entailment* | | | |
| Model | MNLI-m | MNLI-mm | Lexical | Subseq. | Const. | Lexical | Subseq. | Const. | Avg. |
| BERT-CLS | 84.3 | 84.7 | 97.5 | 99.8 | 99.8 | 33.4 | 4.3 | 12.9 | 57.9 |
| BERT-Attn | 84.6 | 84.4 | 95.3 | 99.5 | 99.3 | 39.1 | 6.0 | 16.4 | 59.3 |
| BERT+LF | 84.7 | **84.9** | 95.2 | 99.3 | 98.7 | 45.0 | 7.3 | 11.2 | 59.5 |
| SPINN | 68.0 | 67.2 | 93.8 | 96.3 | 93.1 | 17.5 | 13.1 | 9.6 | 53.9 |
| BERT+CAGCN | **85.0** | **84.9** | 94.9 | 99.5 | 98.9 | 64.0 | 8.8 | 14.6 | **63.5** |
| BERT+SWGCN | 84.9 | **84.9** | 93.7 | 98.9 | 98.6 | 48.9 | 9.8 | 23.9 | 62.3 |
| BERT+SGCN | 84.5 | 84.8 | 94.6 | 98.9 | 99.0 | 49.2 | 8.3 | 13.0 | 60.5 |

Table 3: Model accuracies on MNLI and HANS

diagnose whether an NLI model has learned specific invalid heuristics in the training data, in order to evaluate its generalization ability.

The number of GCN layers is set at 3. The BERT components in BERT-related models are initialized with the same pre-trained weights. The BERT-related baselines are optimised using the Adam optimizer (Loshchilov and Hutter, 2017). For the proposed models, we adopt two different Adam optimizers for BERT and the other components in the model, respectively (Liu and Lapata, 2019).

### 4.1 Results

Intuitively, the effectiveness of syntax can be obvious when the sentence pair has high lexical overlap but are syntactically different, which leads to semantic diversity. In HANS, this category of examples is named as *non-entailment lexical overlap*, where the words in hypothesis are derived from premise and do not form a contiguous subsequence of the premise. The performance comparison on this category is shown in Table 2. It can be observed that our models outperform the baselines including BERT by a wide margin. This result proves that incorporating syntax by using GCN is indeed beneficial for the generalization of BERT, especially identifying different syntactic structures in the sentence pair. By comparing the results of GCN-based methods, it can be seen that linking words between the sentence pair by co-attention can lead to a better performance. Some examples in this category are shown in Table 1.

The overall results on MNLI and HANS are shown in Table 3. It can be seen that incorporating syntax by GCN improves the averaged precision on the six categories of HANS, and slightly improves the performance on the in-domain dataset MNLI.

### 4.2 Analysis

Despite the success in the first four subcategories in Table 2, the proposed GCN-based methods do not bring as much improvement compared to the baselines on the *Passive* subcategory, and neither on the *Non-entailment-subsequence* and *Non-entailment-constituent* categories in HANS, as shown in Table 3. One common characteristic in these categories is that the syntactic structures and the relative positions of words between the premise and the hypothesis remain basically unchanged. One example of *Passive* subcategory is shown in Figure 2. It can be the reason that both BERT baselines and syntax-based methods perform badly on this type of examples.

Similarly to HANS, the in-domain dataset MNLI also contains examples in which the sentence pair have high lexical overlap. However, most examples of this kind in MNLI have supporting labels rather than contradicting (McCoy et al., 2019b). Furthermore, more than a half of the contradicting cases contain negation in the premise but not the hypothesis, e.g. “I don’t care.” vs. “I care.”. Note that this bias of MNLI is also one main motivation of HANS. Thus, a model may account for the above trait of MNLI by both learning and evaluation on

it, and the syntactic feature might be learned as a secondary factor compared to content and negation words. This can be the main reason why the proposed syntactic model only improves the performance slightly on MNLI.

## 5 Conclusion

We have investigated the effectiveness of introducing syntax into the NLI task, by adopting GCN to enhance the text representation in existing models such as BERT. Results on HANS show that our method can improve the generalization of BERT, especially on examples where the sentence pair have high lexical overlap but different syntactic structures. It demonstrates that adding inductive biases such as dependency tree by GCN can make sentence encoding more robust.

## Acknowledgments

This work was supported by NSFC No.61976180 and Westlake-BrightDreams Robotics research grant.

## References

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Samuel Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. 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.

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

Cleo Condoravdi, Dick Crouch, Valeria De Paiva, Reinhard Stolle, and Daniel G Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pages 38–45.

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

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar R Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. 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 3806–3812.

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

Yangfan Lei, Yue Hu, Xiangpeng Wei, Luxi Xing, and Quanchao Liu. 2019. Syntax-aware sentence matching with graph convolutional networks. In *International Conference on Knowledge Science, Engineering and Management*, pages 353–364. Springer.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv:1703.04826*.

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2019a. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Deric Pang, Lucy H Lin, and Noah A Smith. 2019. Improving natural language inference with a pretrained parser. *arXiv preprint arXiv:1909.08217*.

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.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385.

Emma Strubell and Andrew McCallum. 2018. Syntax helps elmo understand semantics: Is syntax still relevant in a deep neural architecture for SRL? *arXiv preprint arXiv:1811.04773*.

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

Liangguo Wang, Jing Jiang, Hai Leong Chieu, Chen Hui Ong, Dandan Song, and Lejian Liao. 2017. Can syntax help? improving an LSTM-based sentence compression model for new domains. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1385–1393.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

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*.