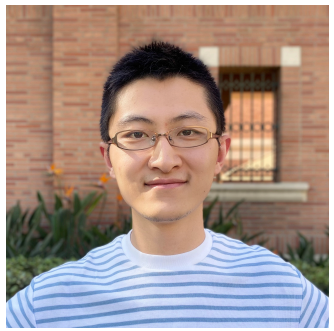
Do Syntax Trees Help Pre-Trained Transformers Extract Information?

EACL 2021

Devendra Singh Sachan, Yuhao Zhang, Peng Qi, William Hamilton



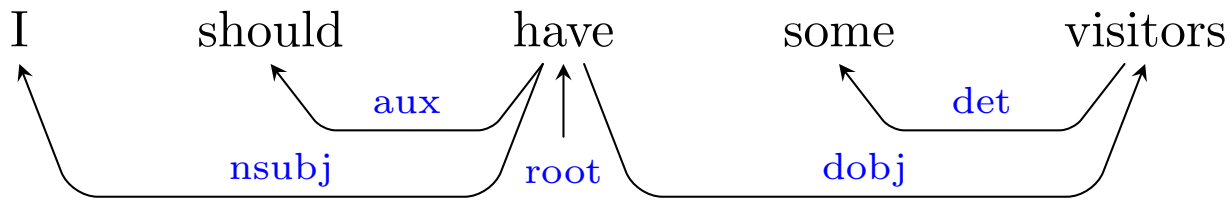



Introduction: Background and Problem Statement

Proposed Model

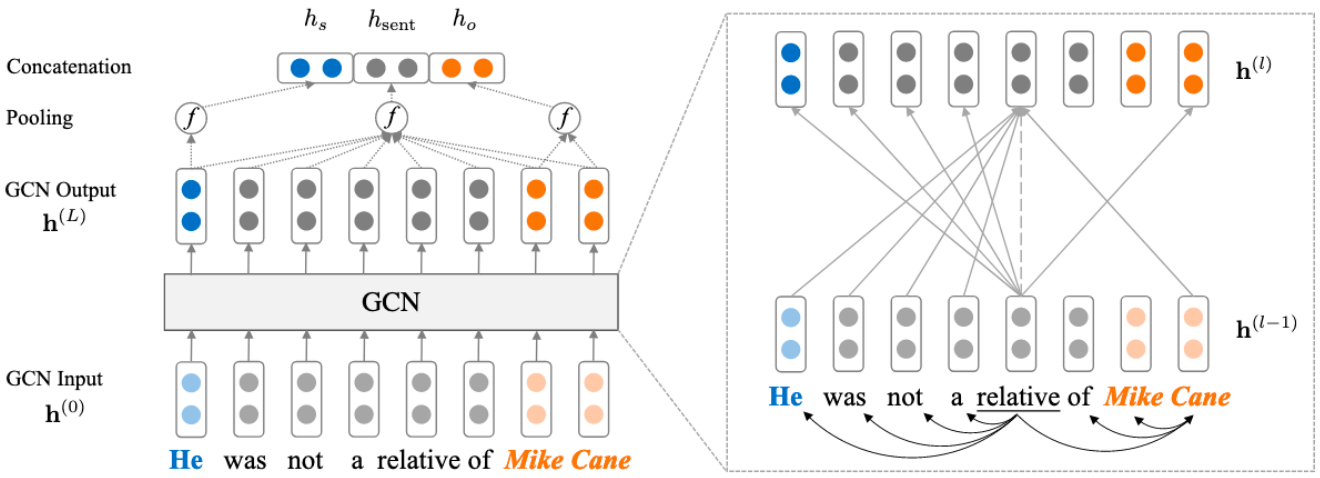
Experiments and Results

Syntax Formalism: Dependency Tree

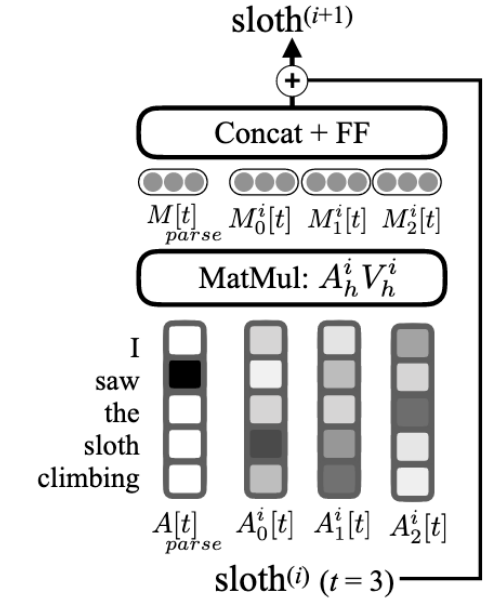


- Dependency trees is a form of linguistic syntax representation.
- Dependency trees encode a syntactic relation between words in a sentence.
- In NLP, information extraction tasks have benefitted from the use of dependency trees.
  - Ex: semantic role labeling, relation extraction.

Previous Work Utilizing Dependency Tree



Graph convolutions applied to relation extraction (Zhang et al. 2018)



Biasing self-attention in the Transformer model with dependency tree (Strubell et al. 2018)

- Previous approaches train randomly initialized sequence models augmented with dependency tree encoders.
- The only pre-trained component was word embeddings.
- Results have demonstrated significant improvements over linear sequence models.

Advent of Pre-trained Transformers

- Pre-trained Transformer models have achieved state-of-the art results.
  - Ex: BERT, RoBERTa, and GPT
- Typical usage: pre-training and/or finetuning.

Pre-training	Finetuning
Self-supervised	Supervised
Predict masked tokens	Downstream task-specific
Compute and time expensive	Much cheaper (few epochs)

- This work: finetuning using open-source BERT / RoBERTa weights

Recent Work: Syntax Information within BERT

- Different linguistic information such as parsing, semantic roles is captured in different layers of BERT (*Tenney et al, 2019*).
- BERT's attention heads attend according to syntactic dependencies (*Clark et al, 2019*).
- BERT's output representation embeds syntactic trees (*Hewitt et al, 2019*).

This Work: Research Question

Recap:

1. External syntax trees has improved the performance of pre-BERT era models.
2. BERT contains some implicit knowledge of syntax.

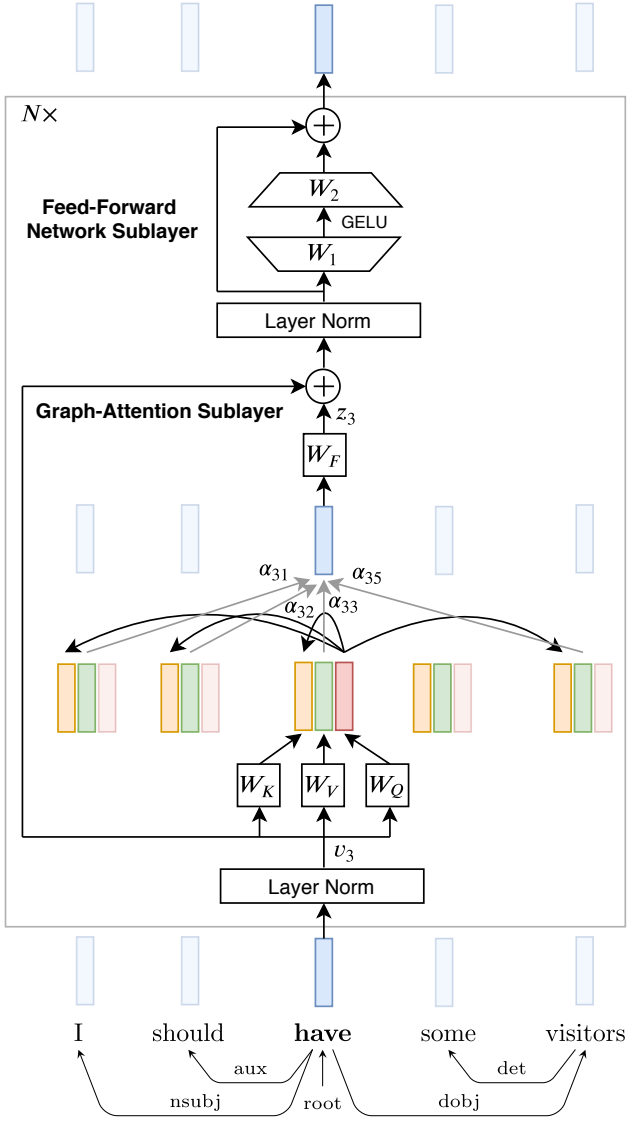
Does external syntax information help BERT improve performance on information extraction tasks?

Introduction: Background and Problem Statement

Proposed Model

Experiments and Results

Syntax-GNN: Graph Encoder over Dependency Tree



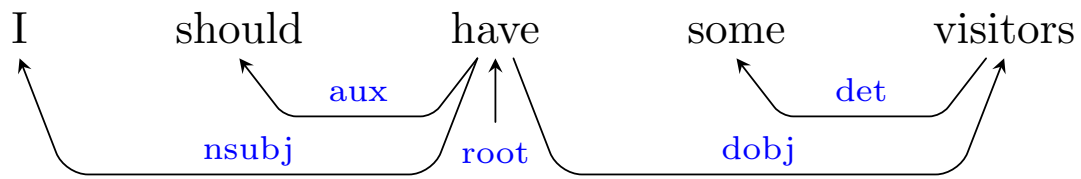
- Modification of the Transformer model
- Self-attention is replaced by graph attention

$$s_{ij} = (v_i \boldsymbol{W}_Q)(v_j \boldsymbol{W}_K)^\top$$ interaction score

$$\alpha_{ij} = \frac{\exp(s_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(s_{ik})}$$ graph attention score

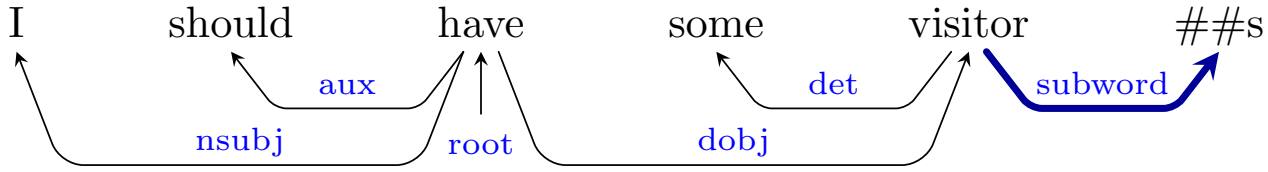
$$z_i = (\sum_{j \in \mathcal{N}_i} \alpha_{ij}(v_j \boldsymbol{W}_V))\boldsymbol{W}_F$$ aggregation step

Dependency Tree over Wordpieces



- Dependency tree is defined over linguistic tokens.
- Wordpiece tokenization can segment a linguistic token into multiple subwords.

Introduce new edges from the first subword (head) to the remaining subwords (tail)

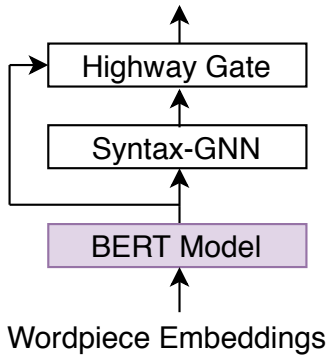


Syntax-Augmented BERT Models

- Methods to incorporate syntax-GNN representations in BERT
  1. Late Fusion
  2. Joint Fusion

- These methods introduce new parameters.
- During finetuning, new parameters are randomly initialized.

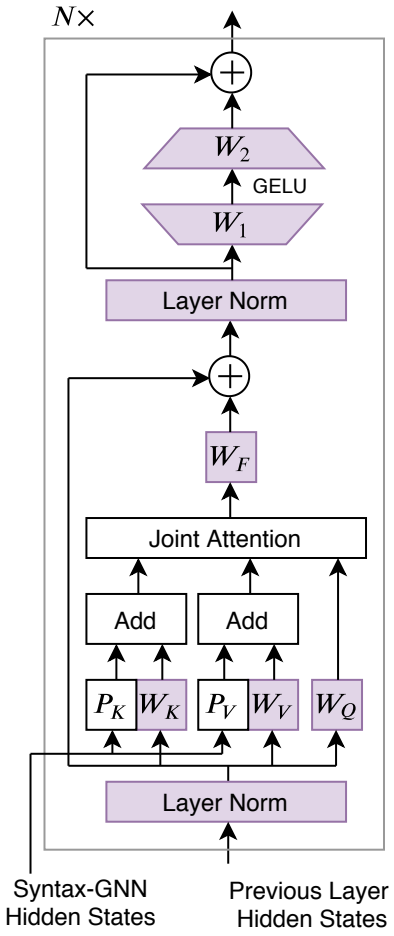
Model 1: Late Fusion



1. Stack **syntax-GNN** on top of the pre-trained Transformer.
2. Highway gate on top selects useful representations.
3. Add hidden states that map to the same linguistic token.

Model 2: Joint Fusion

1. Incorporate **syntax-GNN** representations within self-attention sublayer.
2. Introduce two projection weights per layer $\{\boldsymbol{P}_K, \boldsymbol{P}_V\}$
3. Project syntax-GNN representations and add with BERT layer's keys and values.
4. Joint attention over both syntax and BERT representations.



Introduction: Background and Problem Statement

Proposed Model

Experiments and Results

Tasks and Datasets

Semantic Role Labeling (SRL)	Relation Extraction (RE)	Named Entity Recognition (NER)

Assign semantic role labels to text spans in the sentence.
 	
• Setting: predicates are given
• Datasets:
  • CoNLL-2005 WSJ
  • CoNLL-2012 OntoNotes
	Predict the relation between the two entity mentions.
	
	• Dataset:
	  • TACRED (label corrected)
	  • 41 relation types and a "*no relation*" type
		Recognize and tag the named entities in a sentence.
		
		• Dataset:
		  • OntoNotes 5.0
		  • 18 entity types

Some examples:

SRL: [$_{A0}$ He] [$_{AM\text{-}MOD}$ would] [$_{AM\text{-}NEG}$ n't] [$_{V}$ **accept**] [$_{A1}$ anything of value] from [$_{A2}$ those he was writing about] .

RE: Baldwin declined further comment, and said JetBlue chief executive Dave Barger was unavailable; label: no relation

NER: [$_{PERSON}$ Laura] flew to [$_{LOCATION}$ Silicon Valley].

1. Gold Dependency Parses help on SRL

- Syntax-augmented BERT models achieve new state-of-the art F_1 scores
- Joint Fusion performs better than Late Fusion

CoNLL-2005

	P	R	F_1
Baseline Models (w/o Dependency Parses)			
SA + GloVe	84.2	83.3	83.7
SA + ELMo	86.2	86.0	86.1
$BERT_{BASE}$	87.0	88.0	87.5
Gold Dependency Parses			
Late Fusion	89.2	91.1	**90.1**
Joint Fusion	90.6	91.4	**91.0**

CoNLL-2012

	P	R	F_1
Baseline Models (w/o Dependency Parses)			
SA + GloVe	82.6	80.0	81.3
SA + ELMo	84.4	82.2	83.3
$BERT_{BASE}$	85.9	87.1	86.5
Gold Dependency Parses			
Late Fusion	88.1	90.3	**89.2**
Joint Fusion	89.3	90.4	**89.9**

Gold Dependency Parses don't help on NER

- No performance gains observed in syntax-augmented BERT models on NER

OntoNotes-5.0

	P	R	F_1
Baseline Models (w/o Dependency Parses)			
BiLSTM-CRF + ELMo	88.3	89.7	89.0
$BERT_{BASE}$	88.8	89.6	89.2
Gold Dependency Parses			
DGLSTM-CRF + ELMo	89.6	90.2	**89.9**
Late Fusion	88.8	89.2	89.0
Joint Fusion	88.6	89.3	88.9

Extracted Parses have Mixed Results on RE

- Late Fusion model improves over BERT by 0.3 F_1

- Extracted parses hurt the performance of Joint Fusion model.

TACRED

	P	R	F_1
Baseline Models (w/o Dependency Parses)			
$BERT_{BASE}$	78.0	76.4	77.1
Stanford CoreNLP Dependency Parses			
GCN	74.2	69.3	71.7
GCN + $BERT_{BASE}$	74.8	74.1	74.5
Late Fusion	78.6	76.3	**77.4**
Joint Fusion	70.2	75.1	72.5

2. Impact of Parsing Quality

- Three types of dependency parses:
  - **Gold parses**: human annotated
  - **Off-the-shelf parses**: extracted from Stanza toolkit
  - **In-domain parses**: train a biaffine parser using gold parses

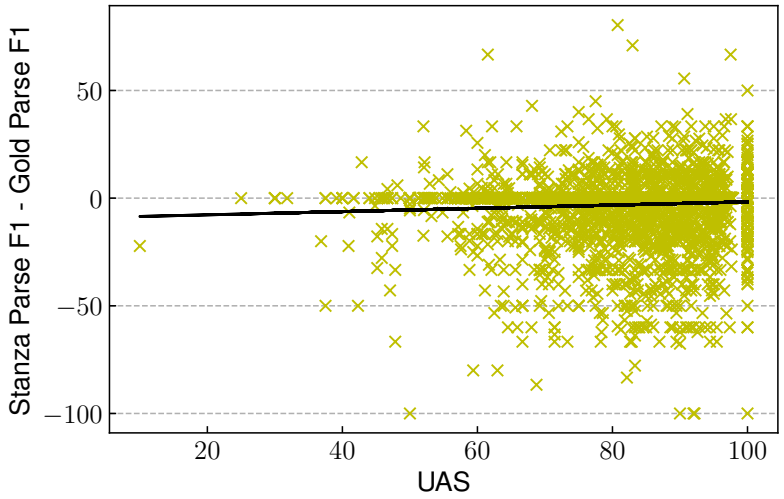
- Stanza and In-domain parses are not helpful

CoNLL-2005

	P	R	F_1
***Baseline Models* (w/o Dependency Parses)**			
$BERT_{BASE}$	87.0	88.0	87.5
Stanza Dependency Parses (UAS: 84.2)			
Late Fusion	86.9	88.1	87.5
Joint Fusion	86.9	87.9	87.4
In-domain Dependency Parses (UAS: 92.7)			
Late Fusion	86.8	88.0	87.4
Joint Fusion	87.1	88.0	87.5
Gold Dependency Parses			
Late Fusion	89.2	91.1	**90.1**
Joint Fusion	90.6	91.4	**91.0**

SRL: Parse Accuracy vs Performance

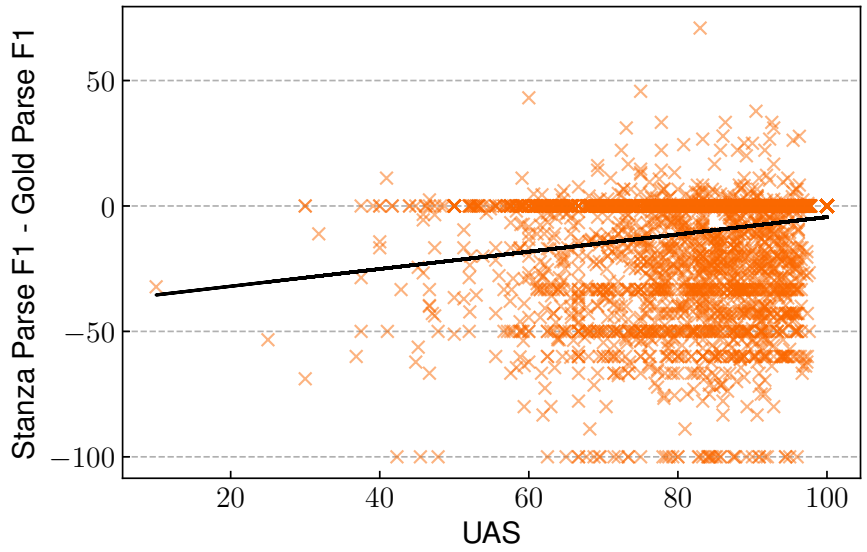
- Small positive correlation between F_1 difference and parse accuracy.
- As the parse accuracy increases, the performance improves.
- Model trained on Stanza parses tends to rely less on the noisy dependency parses.



SRL: Parse Accuracy vs Performance

- Setting: Inference is done using Stanza parses on a model trained with gold parses.

- The model trained on gold parses is more sensitive to the Stanza parses.



3. Generalization to BERT Variants

Is the syntactic information equally useful for more powerful BERT models?

Some examples of other models:

- BERT-large
- BERT-large trained with whole word masking
- RoBERTa

Gains from Late Fusion also generalize to other pre-trained Transformer models.

CoNLL-2005 SRL

	P	R	F_1
BERT			
$BERT_{BASE}$	87.0	88.0	87.5
Late Fusion	89.2	91.1	**90.1**
$BERT_{LARGE}$	88.1	88.8	88.5
Late Fusion	89.9	91.6	**90.7**
$BERT_{WWM}$	88.0	88.9	88.5
Late Fusion	89.9	91.6	**90.8**
RoBERTa			
$RoBERTa_{LARGE}$	89.1	89.9	89.5
Late Fusion	90.9	92.1	**91.5**

Discussion

- We obtain state-of-the-art results on SRL using gold dependency parses.
- Our results show marginal gains from using extracted parses on IE tasks.
- Syntax-Augmented BERT models are sensitive to parse accuracy.
- Future work can leverage *soft edges* in the extracted dependency graphs.

Thank You!

- Paper: https://arxiv.org/abs/2008.09084
- Code: https://github.com/DevSinghSachan/syntax-augmented-bert

- Contact: sachande@mila.quebec