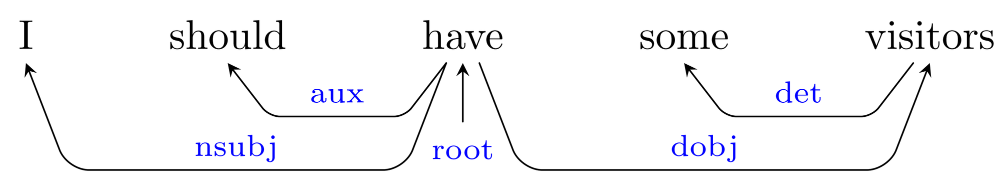
Do Syntax Trees Help Pre-Trained Transformers Extract Information?

Devendra Singh Sachan[1,2], Yuhao Zhang[3], Peng Qi[3], William Hamilton[1,2]
[1]McGill University, [2]Mila, [3]Stanford University
sachande@mila.quebec

Introduction

Syntax Formalism: Dependency Tree



- Dependency tree encode a syntactic relation between words.
- Information extraction (IE) tasks have benefitted from the use of dependency trees.

Previous Work Utilizing Dependency Tree

- Based on randomly initialized sequence models + dependency tree encoders.
- Ex1: Graph convolutions applied to relation extraction (*Zhang et al. 2018*).
- Ex2: Biasing Transformer's self-attention with dependency tree (*Strubell et al. 2018*).
- Demonstrated significant improvements over linear sequence models.

Recent Work: Syntax Information within BERT

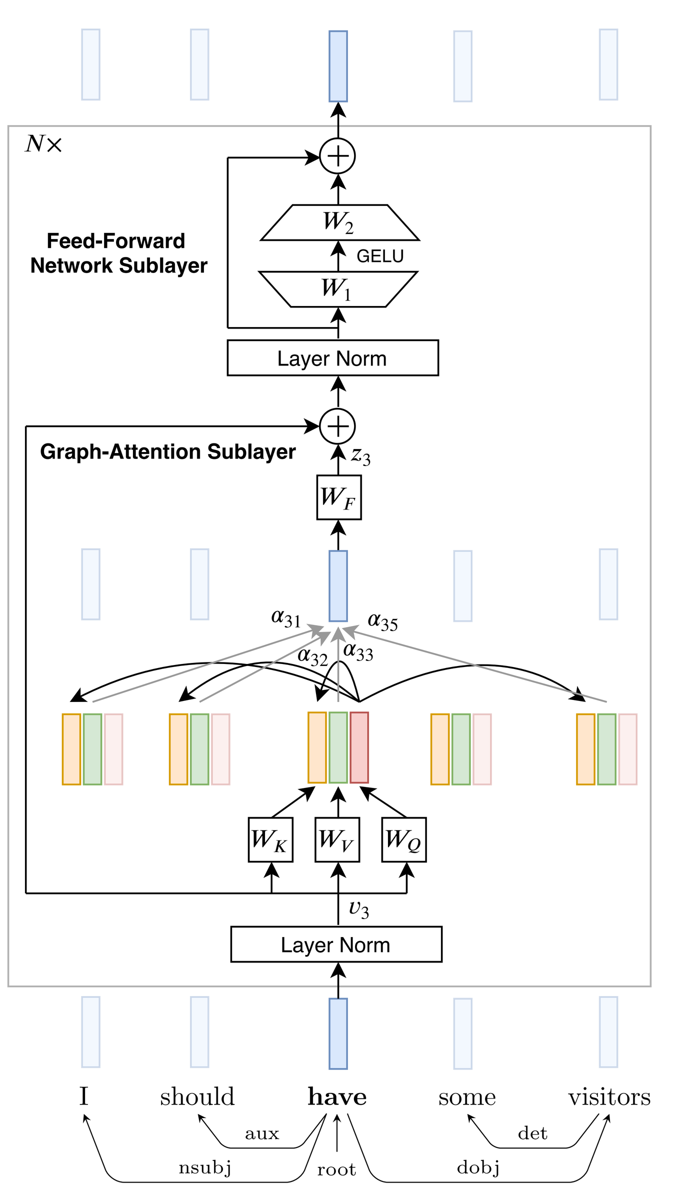
- Different linguistic information such as parsing, semantic roles is captured in different layers of BERT (*Tenney et al, 2019*).
- BERT's attention heads attend according to linguistic syntax (*Clark et al, 2019*).
- BERT's output representation embeds syntactic trees (*Hewitt et al, 2019*).

Research Question

Does external syntax information from dependency trees help BERT improve performance on *information extraction tasks*?

Methods

Syntax-GNN: Graph Encoder over Dependency Tree



- Modification of the Transformer encoder
- Self-attention → Graph attention

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

interaction score

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

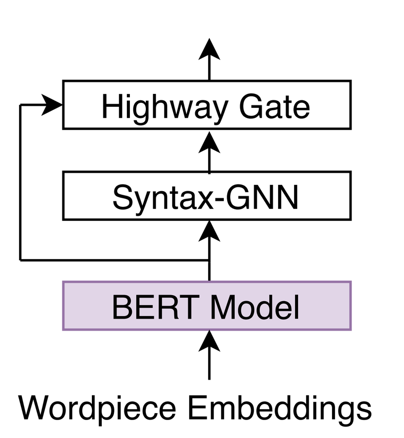
graph-attention score

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

aggregation

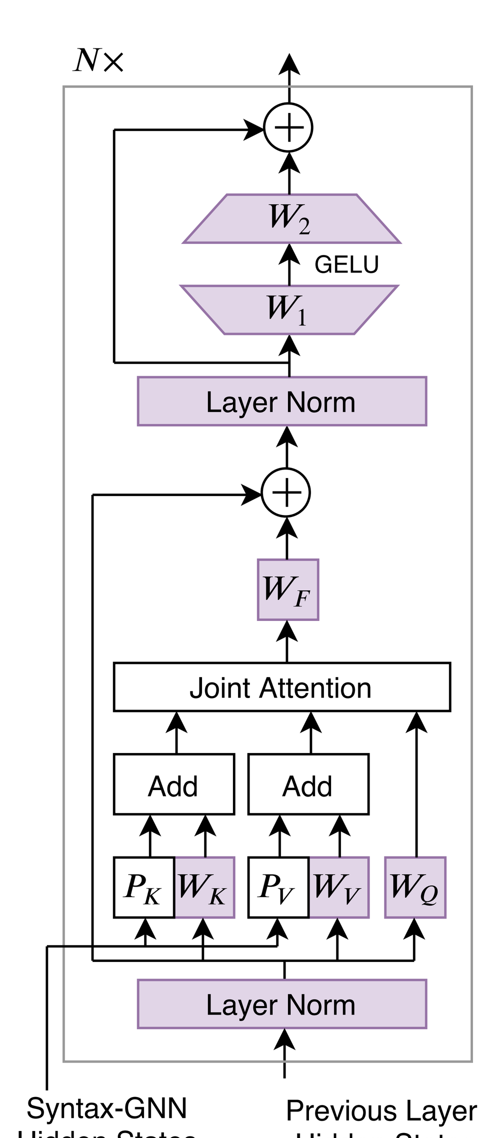
Syntax-Augmented BERT Models

Late Fusion



- Stack syntax-GNN over BERT.
- Highway gate selects useful representations.
- Add hidden states that map to the same linguistic token.

Joint Fusion



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

Tasks and Datasets

Semantic Role Labeling

Assign semantic role labels to text spans.

Predicates are given.

Datasets:
- CoNLL-2005 WSJ
- CoNLL-2012 OntoNotes

Relation Extraction

Predict the relation between the two entity mentions.

Dataset:
- TACRED (label corrected)
- 41 relation types and a "*no relation*" type

Named Entity Recognition

Recognize and tag the named entities in a sentence.

Dataset:
- OntoNotes 5.0
- 18 entity types

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].

Results and Analysis

Impact of Parsing Quality

- Three types of parses: (a) **Gold parses**: human annotated (b) **Stanza parses**: extracted from ***Stanza toolkit*** (c) **In-domain parses**: train a parser using gold parses

CoNLL-2005 SRL

Test Set	P	R	F$_1$
Baseline Models (without 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**

CoNLL-2012 SRL

Test Set	P	R	F$_1$
Baseline Models (without dependency parses)			
BERT$_{BASE}$	85.9	87.1	86.5
Stanza Dependency Parses (UAS: 82.7)			
Late Fusion	85.7	87.2	86.5
Joint Fusion	85.9	87.1	86.5
In-domain Dependency Parses (UAS: 93.6)			
Late Fusion	86.1	86.9	86.5
Joint Fusion	85.8	86.9	86.3
Gold Dependency Parses			
Late Fusion	88.1	90.3	89.2
Joint Fusion	89.3	90.4	**89.9**

- **Using gold parses, syntax-augmented models achieve new best results.**
- Stanza and in-domain parses are not much helpful for SRL.

Relation Extraction

Test Set	P	R	F$_1$
Baseline Models (without 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

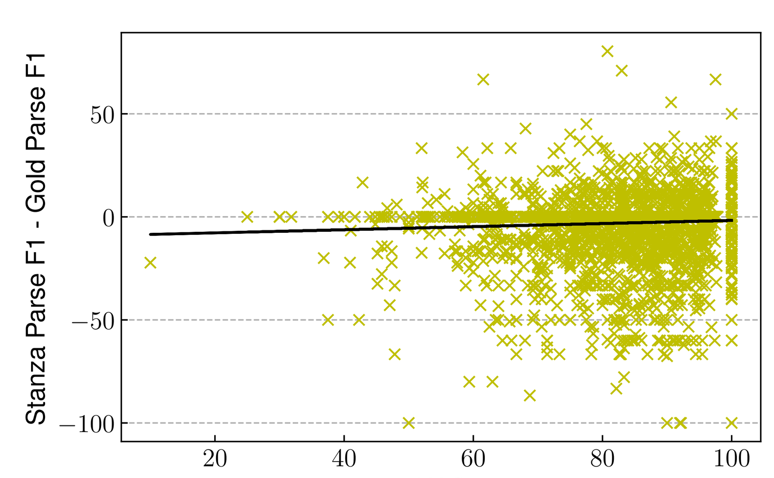
- Late Fusion improves over BERT by 0.3 F$_1$
- Extracted parses hurt performance of the Joint Fusion model.

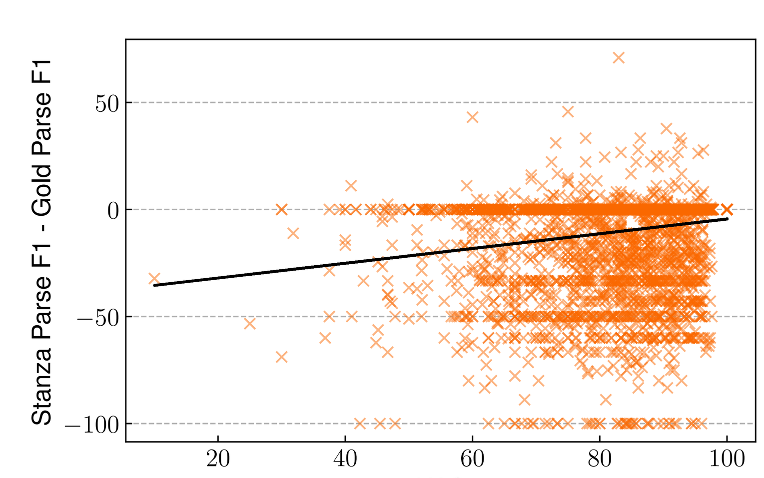
Named Entity Recognition

Test Set	P	R	F$_1$
Baseline Models (without dependency parses)			
BERT$_{BASE}$	88.8	89.6	89.2
Stanza Dependency Parses (UAS: 83.9)			
Late Fusion	88.8	89.4	89.1
Joint Fusion	88.6	89.4	89.0
Gold Dependency Parses			
Late Fusion	88.8	89.2	89.0
Joint Fusion	88.6	89.3	88.9

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

SRL: Parse Accuracy vs Performance





- Small positive correlation between F$_1$ difference and parse accuracy.
- Model trained on Stanza parses tends to rely less on the noisy parses.

- Inference is done using Stanza parses on a model trained with gold parses.
- The model trained on gold parses is more sensitive to Stanza parses.

Conclusions

- 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.

Code: https://github.com/DevSinghSachan/syntax-augmented-bert