Grammar as a Foreign Language

•••

Authors:- Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton

Presented by:- Ved Upadhyay

PaperLink:-https://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.pdf

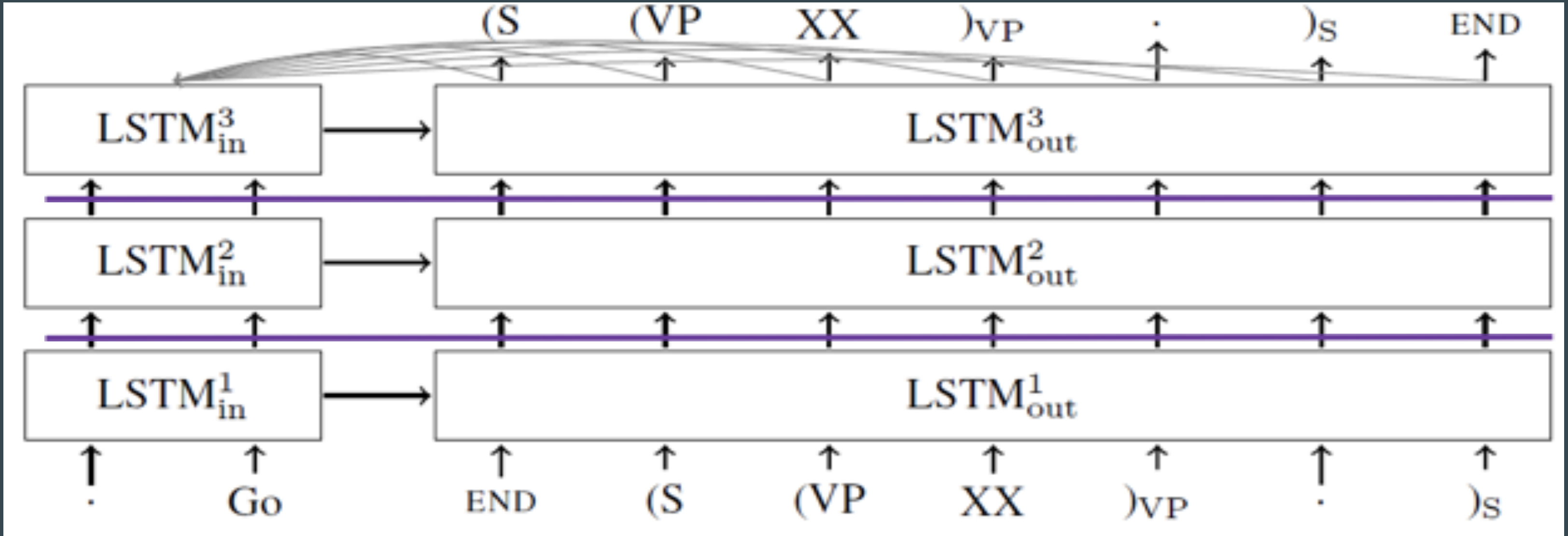
Contents

- Introduction and outline of paper
- Overview of LSTM+A Parsing Model
- Involved attention mechanism
- Experiments
    - Discussion about training data
    - Evaluation of model
- Further analysis
- Conclusion

Introduction and outline of paper

- Attention-enhanced Seq-to-Seq model gives state-of-the-art results on large synthetic corpus
- Matches the performance of standard parsers when trained only on a small human-annotated dataset
- Highly data-efficient, in contrast to Seq-to-Seq models without the attention mechanism

Overview of LSTM+A Parsing Model



Drop out layers are shown in purple.

Architecture of LSTM+A model

Quick Training Details:

- Used a model with 3 LSTM layers.
- **Dropout** between layers 1 and 2, 2 and 3
- **No POS tags**

  - F1 score is improved by 1 point by leaving them out
  - Since POS tags are not evaluated in syntactic parsing F1 score, they are replaced all by “XX” in training data

Dropout Layer

- A technique where randomly selected neurons are ignored during training
- Neurons are temporarily disconnected from the network.
- Other neurons step in and handle the representation required to make predictions for the missing neurons

Dropout Layer - Benefits

- Makes network less sensitive to the specific weights of neurons

- Network gets better generalization and is less likely to overfit the training data*

*http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

Attention Mechanism

- Important extension to Seq-to-Seq model

- Two separate LSTMs - One to encode input words sequence, and another one to decode the output symbols

- The encoder hidden states are denoted $(h_1, . . . , h_{T_A})$ and we denote the hidden states of the decoder by $(d_1, . . . , , d_{T_B}) := (h_{T_A+1}, . . . , h_{T_A+T_B})$

Attention Mechanism

To compute the attention vector at each output time t over the input words $(1, \ldots, T_A)$ *we define:*

$$u_i^t = v^T \tanh(w_1' h_i + W_2' d_t)$$
$$a_i^t = softmax(u_i^t)$$
$$d_t' = \sum_{t=1}^{T_A} a_i^t h_i$$

- Scores are normalized by softmax to create the attention mask a^t over encoder hidden states
- Concatenate d_t' $with$ d_t,to get the new hidden state for making predictions, which is fed to next time step in the recurrent model

Experiments (Training data)

- Model is trained on two different datasets - Standard WSJ training data set, high confidence corpus.

- WSJ dataset contains only 40k sentences but results from training on this dataset match with those obtained by domain specific parsers

Experiments (Training data):-

High-Confidence Corpus:-

A corpus parsed with existing parsers **BerkeleyParser** and **ZPar**, are used to process unlabeled sentences sampled from news appearing on the web.

- Selected sentences where both parsers produced the same parse tree and re-sample to match the distribution of sentence lengths of the WSJ training corpus.
- The set of ~11 million sentences selected in this way, together with the ~90K golden sentences , are called **the high-confidence corpus.**

Experimentation:-

- Training on WSJ only a baseline LSTM performs bad, even with dropout and early stopping.
- Training on parse trees **generated by the Berkeley Parser** gives 90.5 F1 score
- A single attention model gets to 88.3.
- An ensemble of 5 LSTM+A+D achieves 90.5 matching a single model BerkeleyParser on WSJ23
- Finally, when trained on **high-confidence corpus,** LSTM+A model gave a **new state-of-the-art of 92.1 F1 score.**

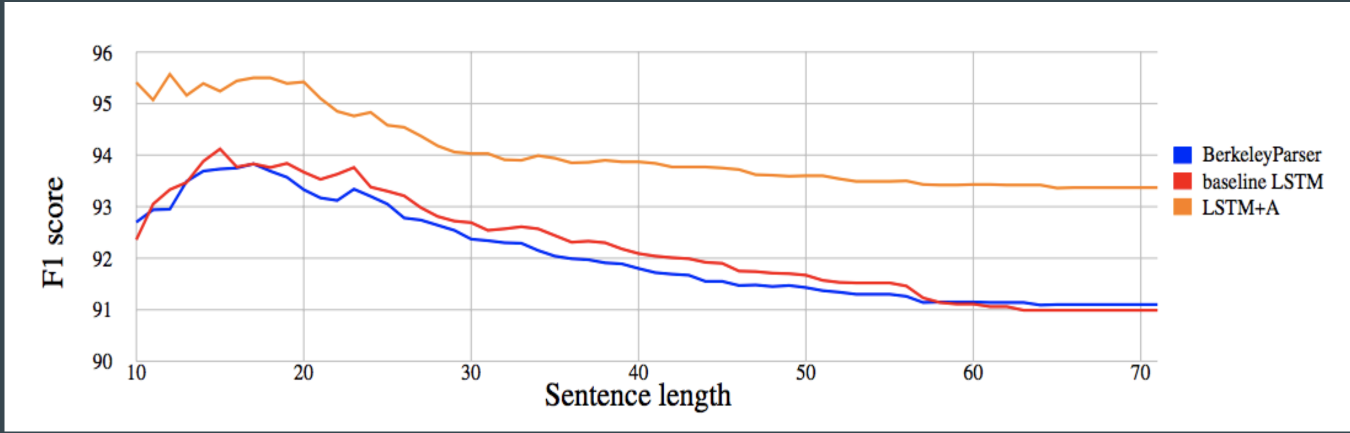
Results - F1 scores of various parsers

Parser	Training set	WSJ22	WSJ23
Baseline LSTM+D LSTM+A+D LSTM+A+D ensemble	WSJ only WSJ only WSJ only	<70 88.7 90.7	<70 88.3 90.5
Baseline LSTM LSTM+A	BerkeleyParser corpus high-confidence corpus	91.0 92.8	90.5 92.1
Petrov et al. (2006) Zhu et al. (2013) Petrov et al. (2010) ensemble	WSJ only WSJ only WSJ only	91.1 N/A 92.5	90.4 90.4 91.8
Zhu et al. (2013) Huang & Harper (2009) McClosky et al. (2006)	Semi-supervised Semi-supervised Semi-supervised	N/A N/A 92.4	91.3 91.3 92.1

Experimentation - Evaluation

- Standard **EVALB** tool is used for evaluation and F1 scores on the development set are reported

Experimentation - Evaluation



Effect of sentence length on the F1 score on WSJ 22.

- The difference between the F1 score on sentences of length up to 30 and 70 is 1.3 for the BerkeleyParser, 1.7 for the baseline LSTM, and 0.7 for LSTM+A
- LSTM+A shows less degradation with length than BerkeleyParser

Experimentation - Evaluation

Dropout Influence

- Used dropout when training on the small WSJ dataset and its influence was significant.

- A single LSTM+A model only achieved an F1 score of 86.5 on the development set, that is over 2 points lower than the 88.7 of a LSTM+A+D model.

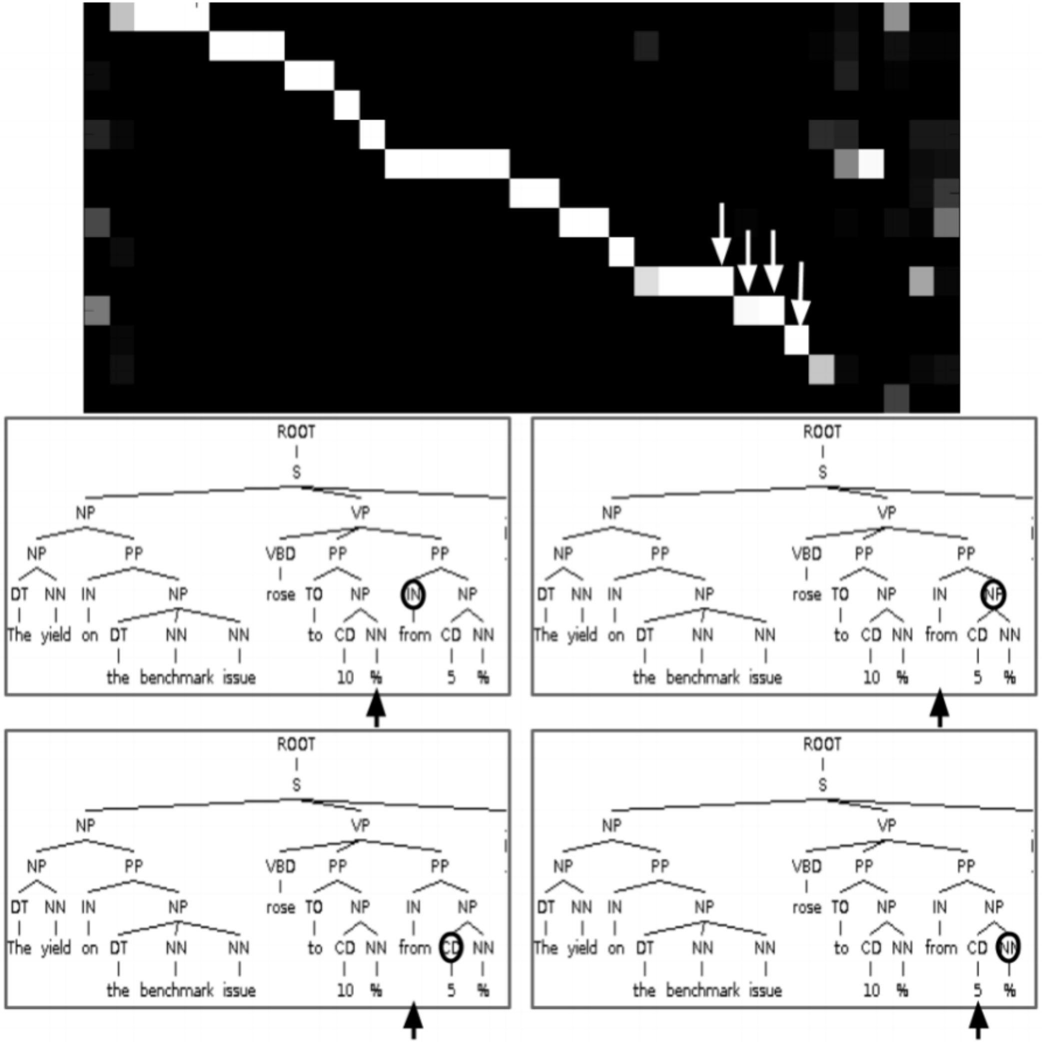
Experimentation - Evaluation

Performance on other datasets

- To check how well it generalizes, it is tested on two other datasets - QEB & WEB
- LSTM+A trained on the high-confidence corpus achieved an F1 score of 95.7 on QTB and 84.6 on WEB

Parsing speed

- Parser is fast
- LSTM+A model, running on a multi-core CPU using batches of 128 sentences on an unoptimized decoder, can parse over 120 sentences from WSJ per second for sentences of all lengths



- On top is the attention matrix, each column is the attention vector over the inputs.
- On bottom, shown outputs for four consecutive time steps, the attention mask moves to the right.
- Focus moves from the first word to the last monotonically, steps to the right when a word is consumed.
- On the bottom, we see where the model attends (black arrow), and the current output being decoded in the tree (black circle)

Analysis

- Model did not over fit; learned the parsing function from scratch much faster

- Better generalization compared to plain LSTM without attention.

- Attention allows us to visualize what the model has learned from the data.

- From the attention matrix, it is clear that the model focuses quite sharply on one word as it produces the parse tree

Conclusion

- Seq-to-Seq approaches can achieve excellent results on syntactic constituency parsing with little effort or tuning

- Synthetic datasets with imperfect labels can be highly useful, LSTM+A models have substantially outperformed the previously used models

- Domain independent models with excellent learning algorithms can match and even outperform domain specific models.

Questions ?