Is the End of Supervised Parsing in Sight? Twelve Years Later

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
Rens Bod. 2007. Is the End of Supervised Parsing in Sight?  
Twelve Years Later
Twelve Years Later

Is the end of all parsing (supervised or other) in sight?
Twelve Years Later

Is the end of all parsing (supervised or other) in sight?
Is the end of the parsers that we know and love in sight?
Back to Prague

Back to Prague

Back to Prague

Parsing algorithms matter!

Lots of cool work on parsing (and learning) algorithms

Graph-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

Transition-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

Higher-order chart parsing
Pruning
ILP
Dual decomp
Mildly non-projective
Etc.

Beam search
Perceptron
Dynamic oracles
Dynamic programming
More features
Etc.

LAS: 83.8 v. 83.6
[McDonald & Nivre 2007]

LAS: 85.8 v. 85.5
[Zhang et al. 2013]

Lots of cool work on parsing (and learning) algorithms

Graph-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

Transition-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

higher-order chart parsing
pruning
ILP
dual decomp
mildly non-projective
etc.

beam search
perceptron
dynamic oracles
dynamic programming
more features
etc.

LAS: 83.8 v. 83.6
[McDonald & Nivre 2007]

LAS: 85.8 v. 85.5
[Zhang et al. 2013]

Lots of cool work on parsing (and learning) algorithms

Graph-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

Transition-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

- beam search
- perceptron
- dynamic oracles
- dynamic programming
- more features
- etc.

- higher-order chart parsing
- pruning
- ILP
- dual decomp
- mildly non-projective
- etc.

LAS: 83.8 v. 83.6
[McDonald & Nivre 2007]

LAS: 85.8 v. 85.5
[Zhang et al. 2013]

Enter Deep Learning

- Dense continuous representations – embeddings
- Unbounded contextual information – encoders
- All features learned together – end-to-end training
Figure 2: Illustration of the neural model scheme of the graph-based parser when calculating the score of a given parse tree. The parse tree is depicted below the sentence. Each dependency arc in the sentence is scored using an MLP that is fed the BiLSTM encoding of the words at the arc’s end points (the colors of the arcs correspond to colors of the MLP inputs above), and the individual arc scores are summed to produce the final score. All the MLPs share the same parameters. The figure depicts a single-layer BiLSTM, while in practice we use two layers. When parsing a sentence, we compute scores for all possible \( n^2 \) arcs, and find the best scoring tree using a dynamic-programming algorithm.

Given the scores of the arcs the highest scoring projective tree can be efficiently found using Eisner’s decoding algorithm (1996). McDonald et al. and most subsequent work estimate the local score of an arc by a linear model parameterized by a weight vector \( w \), and a feature function assigning a sparse feature vector for an arc linking modifier \( m \) to head \( h \). We follow Pei et al. (2015) and replace the linear scoring function with an MLP.

The feature extractor \( \langle s, h, m \rangle \) is usually complex, involving many elements (see Section 2.1). In contrast, our feature extractor uses merely the BiLSTM encoding of the head word and the modifier word:

\[
\langle s, h, m \rangle = \text{BiLSTM}(x_1: n, h) \quad \text{BiLSTM}(x_1: n, m)
\]

The final model is:

\[
\text{parse}(s) = \arg \max_y \text{score}_{\text{global}}(s, y) = \arg \max_y \text{score}_{\text{local}}(s, h, m) = \arg \max_y \text{MLP}(v_h, v_m)
\]

Training

The training objective is to set the score function such that correct tree \( y \) is scored above incorrect ones. We use a margin-based objective (McDonald et al., 2005; LeCun et al., 2006), aiming to maximize the margin between the score of the gold tree \( y \) and the highest scoring incorrect tree \( y_0 \). We define a hinge loss with respect to a gold tree \( y \) as:

\[
E(y, y_0) = \max(0, \text{score}(y) - \text{score}(y_0) + \delta)
\]

where \( \delta \) is a margin parameter.

In this work, we focus on arc-factored graph-based approach presented in McDonald et al. (2005). Arc-factored parsing decomposes the score of a tree to the sum of the score of its head-modifier arcs \((h, m)\):

\[
\text{parse}(s) = \arg \max_y \sum_{(h,m)} \text{score}(s, h, m)
\]

Given the scores of the arcs the highest scoring projective tree can be efficiently found using Eisner's decoding algorithm (1996). McDonald et al. and most subsequent work estimate the local score of an arc by a linear model parameterized by a weight vector \(w\), and a feature function \((s, h, m)\) assigning a sparse feature vector for an arc linking modifier \(m\) to head \(h\). We follow Pei et al. (2015) and replace the linear scoring function with an MLP.

The feature extractor \((s, h, m)\) is usually complex, involving many elements (see Section 2.1). In contrast, our feature extractor uses merely the BiLSTM encoding of the head word and the modifier word:

\[
(s, h, m) = \text{BiLSTM}(x_1: n, h) \quad \text{BiLSTM}(x_1: n, m)
\]

The final model is:

\[
\text{parse}(s) = \arg \max_y \sum_{(h,m)} \text{score}(s, h, m)
\]

\[v_i = \text{BiLSTM}(x_1: n, i)\]

The architecture is illustrated in Figure 2.
In this work, we focus on arc-factored graph-based approach presented in McDonald et al. (2005). Arc-factored parsing decomposes the score of a tree to the sum of the score of its head-modifier arcs \((h, m)\):

\[
\text{parse}(s) = \arg \max_y \sum_{(h,m)} \text{score}(s, h, m)
\]

Given the scores of the arcs the highest scoring projective tree can be efficiently found using Eisner's decoding algorithm (1996). McDonald et al. and most subsequent work estimate the local score of an arc by a linear model parameterized by a weight vector \(w\), and a feature function \((s, h, m)\) assigning a sparse feature vector for an arc linking modifier \(m\) to head \(h\). We follow Pei et al. (2015) and replace the linear scoring function with an MLP.

The feature extractor \((s, h, m)\) is usually complex, involving many elements (see Section 2.1). In contrast, our feature extractor uses merely the BiLSTM encoding of the head word and the modifier word:

\[
(s, h, m) = \text{BiLSTM}(x_{1:n}, h)
\]

The final model is:

\[
\text{parse}(s) = \arg \max_y \sum_{(h,m)} \text{score}(s, h, m)
\]

\[
\text{MLP}(v_h, v_m) = \text{BiLSTM}(x_{1:n})
\]
Figure 2: Illustration of the neural model scheme of the graph-based parser when calculating the score of a given parse tree. The parse tree is depicted below the sentence. Each dependency arc in the sentence is scored using an MLP that is fed the BiLSTM encoding of the words at the arc's end points (the colors of the arcs correspond to colors of the MLP inputs above), and the individual arc scores are summed to produce the final score. All the MLPs share the same parameters. The figure depicts a single-layer BiLSTM, while in practice we use two layers. When parsing a sentence, we compute scores for all possible \( n^2 \) arcs, and find the best scoring tree using a dynamic-programming algorithm.
Lots of cool work on parsing (and learning) algorithms

Graph-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

Transition-based Parsers
- Global Inference
- Global Learning
- Global Feature Scope

2014

higher-order chart parsing
pruning
ILP
dual decomp
mildly non-projective
etc.

beam search
perceptron
dynamic oracles
dynamic programming
more features
etc.

LAS: 83.8 v. 83.6
[McDonald & Nivre 2007]

LAS: 85.8 v. 85.5
[Zhang et al. 2013]

Do Parsing Algorithms Still Matter?

Do Parsing Algorithms Still Matter?

Further Evidence

Further Evidence

A Brief History of Parsing

Grammar

Algorithm
A Brief History of Parsing

CFG

Algorithm
A Brief History of Parsing

CFG  CKY
A Brief History of Parsing

PCFG

CKY
A Brief History of Parsing

PCFG

CKY

Corpus
A Brief History of Parsing

PCFG

CKY

Corpus
Annotation
A Brief History of Parsing

- LM($x, y$)
- MST
- Corpus Annotation
A Brief History of Parsing

DNN(x, y)

MST

Corpus Annotation
A Brief History of Parsing

DNN(x, y)

Corpus Annotation

???
A Brief History of Parsing

DNN(x, y)

Corpus

???
A Brief History of Parsing

DNN(x, y)  ???

Corpus  ???
Conclusion

• Parsing is no longer structured prediction?
• Parsing no longer requires explicit supervision?
• Higher accuracy with simpler and more general models!
Conclusion

• Parsing is no longer structured prediction?
• Parsing no longer requires explicit supervision?
• Higher accuracy with simpler and more general models!

Everything should be made as simple as possible, but not any simpler.
Conclusion

• Parsing is no longer structured prediction?
• Parsing no longer requires explicit supervision?
• Higher accuracy with simpler and more general models!

Everything should be made as simple as possible, but not any simpler.

But is it really so simple?