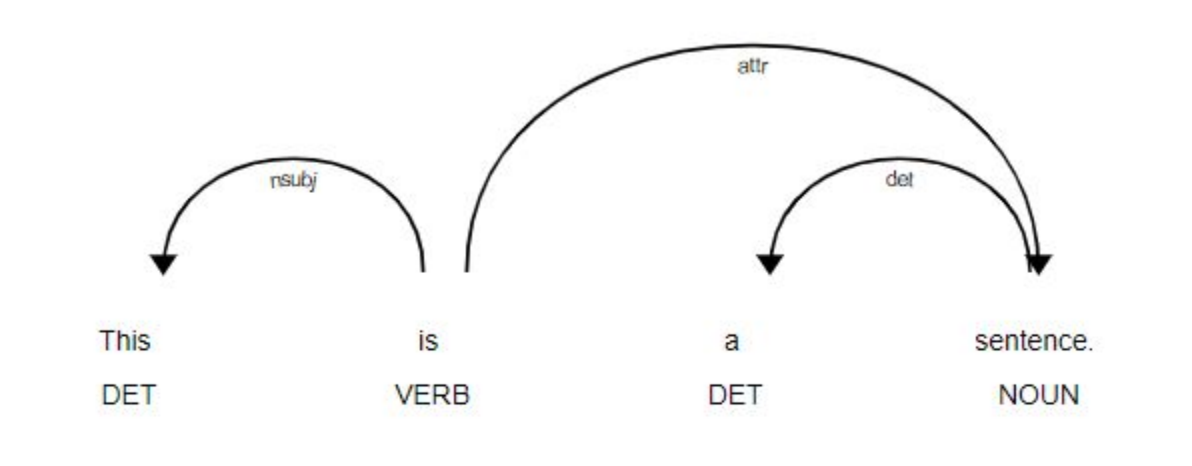
Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations

Leon, Shahin, Matias and Eliasz

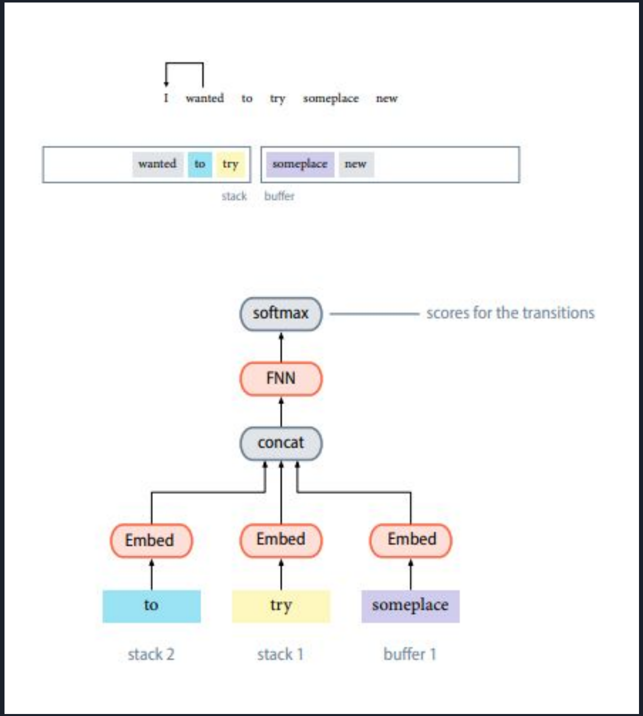
Dependency Parsing



(courtesy of spacy.io)

Overview

- Implemented the baseline
  - transition based dependency parsing
- Language data:
  - English
  - Arabic
  - Spanish
  - Swedish



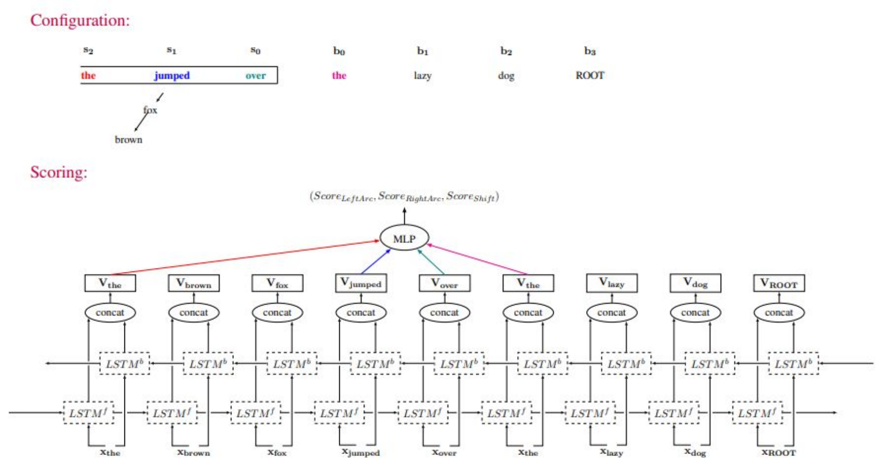
(courtesy of Marco)

So how did we improve? - Bidirectional LSTM

1.
$$x_i = e(w_i) \circ e(p_i)$$

2.
input element as its (deep) BiLSTM vector, v_i:

$$v_i = \text{BiLSTM}(x_{1:n}, i)$$

3.


(Kiperwasser and Goldberg, 2016)

Why did we choose this project?

- Natural extension to baseline.
- The idea.

Algorithm 1 Greedy transition-based parsing

Input: sentence $s = w_1, \ldots, x_w,\ t_1, \ldots, t_n$, parameterized function $\textsc{Score}_\theta(\cdot)$ with parameters θ.
$c \leftarrow \textsc{Initial}(s)$
while not $\textsc{Terminal}(c)$ **do**
  $\hat{t} \leftarrow \arg\max_{t \in \textsc{Legal}(c)} \textsc{Score}_\theta\big(\phi(c), t\big)$
  $c \leftarrow \hat{t}(c)$
return $tree(c)$

(Kiperwasser and Goldberg, 2016)

Why did we choose this project?

- Natural extension to baseline.
- The idea.

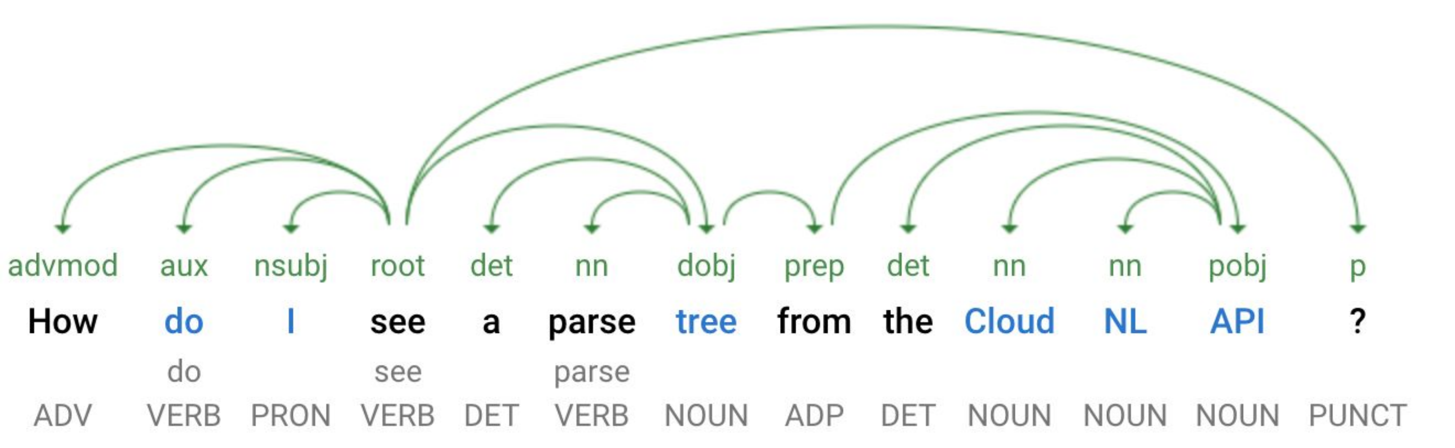
Algorithm 1 Greedy transition-based parsing

Input: sentence $s = w_1, \ldots, x_w,\ t_1, \ldots, t_n,$ parameterized function $\text{SCORE}_\theta(\cdot)$ with parameters θ.
$c \leftarrow \text{INITIAL}(s)$
while not $\text{TERMINAL}(c)$ **do**
  $\hat{t} \leftarrow \arg\max_{t \in \text{LEGAL}(c)} \text{SCORE}_\theta\big(\phi(c), t\big)$
  $c \leftarrow \hat{t}(c)$
return $tree(c)$

(Kiperwasser and Goldberg, 2016)

Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations

- Dependency Parsing: Simple vs Hard



Traditional state-of-the-art

from single words
S_0wp; S_0w; S_0p; N_0wp; N_0w; N_0p;
N_1wp; N_1w; N_1p; N_2wp; N_2w; N_2p;
from word pairs
S_0wpN_0wp; S_0wpN_0w; S_0wN_0wp; S_0wpN_0p;
S_0pN_0wp; S_0wN_0w; S_0pN_0p
N_0pN_1p
from three words
$N_0pN_1pN_2p$; $S_0pN_0pN_1p$; $S_{0h}pS_0pN_0p$;
$S_0pS_{0l}pN_0p$; $S_0pS_{0r}pN_0p$; $S_0pN_0pN_{0l}p$

Table 1: Baseline feature templates.
w – word; p – POS-tag.

distance
S_0wd; S_0pd; N_0wd; N_0pd;
S_0wN_0wd; S_0pN_0pd;
valency
S_0wv_r; S_0pv_r; S_0wv_l; S_0pv_l; N_0wv_l; N_0pv_l;
unigrams
$S_{0h}w$; $S_{0h}p$; S_0l; $S_{0l}w$; $S_{0l}p$; $S_{0l}l$;
$S_{0r}w$; $S_{0r}p$; $S_{0r}l$;$N_{0l}w$; $N_{0l}p$; $N_{0l}l$;
third-order
$S_{0h2}w$; $S_{0h2}p$; $S_{0h}l$; $S_{0l2}w$; $S_{0l2}p$; $S_{0l2}l$;
$S_{0r2}w$; $S_{0r2}p$; $S_{0r2}l$; $N_{0l2}w$; $N_{0l2}p$; $N_{0l2}l$;
$S_0pS_{0l}pS_{0l2}p$; $S_0pS_{0r}pS_{0r2}p$;
$S_0pS_{0h}pS_{0h2}p$; $N_0pN_{0l}pN_{0l2}p$;
label set
S_0ws_r; S_0ps_r; S_0ws_l; S_0ps_l; N_0ws_l; N_0ps_l;

Table 2: New feature templates.
w – word; p – POS-tag; v_l, v_r – valency; l – dependency label, s_l, s_r – labelset.

(Zang and Nivre 2011)

Scientific background

- Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations.
  - Nivre (2004 & 2008)
  - BiLSTMs (Graves, 2008)

Experiment and Results

Universal Dependencies (UD) Dataset

Training settings and Hyper-parameters:

- Max Sentences Per Epoch: 4000
- Epochs: 5
- Word Embedding Dimension: 100
- POS tag embedding dimension: 25
- Hidden Units in MLP: 100
- Bi-LSTM layers: 2
- Bi-LSTM Dimensions (Hidden / Output) : 125 / 125
- Learning Rate: 1e-3

Experiment and Results

Universal Dependencies (UD) Dataset

Results (UAS: unlabeled attachment score) on dev data:

Dataset	1 Epoch	2 Epoch	3 Epoch	4 Epoch	5 Epoch	Time per Sentence (s)
English	80.23%	81.59%	80.36%	81.86%	82.66%	0.40
Arabic	75.34%	78.07%	77.83%	79.36	-	0.75
Spanish	85.31%	86.37%	86.62%	86.59%	86.57%	0.65
Swedish	81.37%	82.34%	83.52%	82.43	82.74%	0.43

Experiment and Results

Universal Dependencies (UD) Dataset

Results UAS Larger Dataset:

- Spanish Dataset, 3 Epochs, 4000 sentences per epoch:
  - 3 * 4000 Same Sentences: 86.62%
- Spanish Dataset, 1 Epoch, 10000 sentences per epoch:
  - 10000 Different Sentences: 87.64%
- Spanish Dataset, 2 Epoch, 10000 sentences per epoch:
  - 10000 Different Sentences: 88.18%

Conclusion

- New classifier gives significant accuracy improvement compared to original
  - +14 percent points on English data
  - +1 percent points on Arabic data
  - +12 percent points on Spanish data
  - +14 percent points on Swedish data
- However, generally worse accuracy results compared to paper
  - Static vs. dynamic oracle
  - Cross-entropy loss vs. hinge loss
  - Dataset differences?