Syntactic and Semantic Parsing

Wanxiang Che
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology
2017-12-1
Why Do We Need Parsing?

- Parsing proposes the (syntactic or semantic) relations between words
- These relations are important for many applications

You are ignorant to retweet the article
The article you retweeted is ignorant
1. Syntactic and Semantic Parsing
2. Pseudo Data for Parsing
3. Applications of Parsing
4. Summary
Outline

1. Syntactic and Semantic Parsing
2. Pseudo Data for Parsing
3. Applications of Parsing
4. Summary
The analysis of a sentence into its constituents, resulting in a parse tree or graph showing their syntactic or semantic relation to each other.

A traditional and core NLP task.

http://ltp.ai/demo.html
Components of Parsing

- Algorithm
- Grammar
- Data
Constituency vs. Dependency

- Dependency Structures
  - Usually easier to be understood
  - More amenable to annotators
Syntactic vs. Universal Dependency

- Universal Dependencies pay more attention to relations between content words
- The universal annotation scheme for all languages

SemEval 2012 Task 5: Chinese Semantic Dependency (Tree)

SemEval 2016 Task 9: Chinese Semantic Dependency (Graph)

SemEval 2015 Task 18: Broad-Coverage Semantic Dependency (Graph)
Data

- Monolingual Single-domain
- Multilingual Multi-domain
- Universal Treebank
- Semantic Dependency Treebank

Rich-resource to Low-resource
Multilingual Treebanks

- CoNLL 2006, 2007 Shared Tasks
  - [http://ilk.uvt.nl/conll/](http://ilk.uvt.nl/conll/)
  - 10 - 12 Languages
CoNLL 2009 Shared Task


Syntactic and Semantic Dependencies in Multiple Languages

7 Languages

We achieved Rank 1

<table>
<thead>
<tr>
<th>Rank</th>
<th>System</th>
<th>Average</th>
<th>Catalan</th>
<th>Chinese</th>
<th>Czech</th>
<th>English</th>
<th>German</th>
<th>Japanese</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Che</td>
<td>82.64</td>
<td>81.84</td>
<td>76.38</td>
<td>83.27</td>
<td>87.00</td>
<td>82.44</td>
<td>85.65</td>
<td>81.90</td>
</tr>
<tr>
<td>2</td>
<td>Chen</td>
<td>82.52</td>
<td>83.01</td>
<td>76.23</td>
<td>80.87</td>
<td>87.69</td>
<td>81.22</td>
<td>85.28</td>
<td>83.31</td>
</tr>
<tr>
<td>3</td>
<td>Merlo</td>
<td>82.14</td>
<td>82.66</td>
<td>76.15</td>
<td>83.21</td>
<td>86.03</td>
<td>79.59</td>
<td>84.91</td>
<td>82.43</td>
</tr>
<tr>
<td>4</td>
<td>Bohnet</td>
<td>80.85</td>
<td>80.44</td>
<td>75.91</td>
<td>79.57</td>
<td>85.14</td>
<td>81.60</td>
<td>82.51</td>
<td>80.75</td>
</tr>
<tr>
<td>5</td>
<td>Asahara</td>
<td>78.43</td>
<td>75.91</td>
<td>73.43</td>
<td>81.43</td>
<td>86.40</td>
<td>69.84</td>
<td>84.86</td>
<td>77.12</td>
</tr>
<tr>
<td>6</td>
<td>Brown</td>
<td>77.27</td>
<td>77.40</td>
<td>72.12</td>
<td>75.66</td>
<td>83.98</td>
<td>77.86</td>
<td>76.65</td>
<td>77.21</td>
</tr>
<tr>
<td>7</td>
<td>Zhang</td>
<td>76.49</td>
<td>75.00</td>
<td>73.42</td>
<td>76.93</td>
<td>82.88</td>
<td>73.76</td>
<td>78.17</td>
<td>75.25</td>
</tr>
<tr>
<td>8</td>
<td>Dai</td>
<td>73.98</td>
<td>72.09</td>
<td>72.72</td>
<td>67.14</td>
<td>81.89</td>
<td>75.00</td>
<td>80.89</td>
<td>68.14</td>
</tr>
<tr>
<td>9</td>
<td>Lu Li</td>
<td>73.97</td>
<td>71.32</td>
<td>65.53</td>
<td>75.85</td>
<td>81.92</td>
<td>70.93</td>
<td>80.49</td>
<td>71.72</td>
</tr>
<tr>
<td>10</td>
<td>Lluis</td>
<td>71.49</td>
<td>56.64</td>
<td>66.18</td>
<td>75.95</td>
<td>81.69</td>
<td>72.31</td>
<td>81.76</td>
<td>65.91</td>
</tr>
<tr>
<td>11</td>
<td>Vallejo</td>
<td>70.81</td>
<td>73.75</td>
<td>67.16</td>
<td>60.50</td>
<td>78.19</td>
<td>67.51</td>
<td>77.75</td>
<td>70.78</td>
</tr>
<tr>
<td>12</td>
<td>Ren</td>
<td>67.81</td>
<td>59.42</td>
<td>75.90</td>
<td>60.18</td>
<td>77.83</td>
<td>65.77</td>
<td>77.63</td>
<td>57.96</td>
</tr>
<tr>
<td>13</td>
<td>Zeman</td>
<td>51.07</td>
<td>49.61</td>
<td>43.50</td>
<td>57.95</td>
<td>50.27</td>
<td>49.57</td>
<td>57.69</td>
<td>48.90</td>
</tr>
</tbody>
</table>
Syntactic Analysis of Non-Canonical Language (SANCL) 2012 Shared Task

- https://sites.google.com/site/sancl2012/
- Organized by Google
- Data: Google Web Treebank (CQA, Newsgroup, Online Review)
- We achieved Rank 2 (Stanford) and 3 (HIT)
Universal Dependencies and POS Tags
  - http://universaldependencies.org/
  - 50+ Languages, 70+ Treebanks
CoNLL 2017 Shared Task

- [http://universaldependencies.org/conll17/](http://universaldependencies.org/conll17/)
- Multilingual Parsing from Raw Text to Universal Dependencies
  - Tasks: Sentence Segmentation, Word Segmentation, POS Tagging, Parsing
  - Training: 45 languages, 64 treebanks
  - Test: 81 treebanks
- 113 Registration Teams
  - Universities: Stanford, CMU, UW, Cornell, Toronto, Cambridge, Tokyo, ...
  - Companies: IBM Research, Facebook, ...
  - China: CAS, Fudan, Shanghai Jiaotong, ...
- Results
  - 33 Submission Teams
  - Rank 1-3: Stanford, Cornell, Stuttgart
  - HIT Rank 4
We organized SemEval 2012 and 2016 Shared Tasks
- [https://www.cs.york.ac.uk/semeval-2012/task5.html](https://www.cs.york.ac.uk/semeval-2012/task5.html)
- [http://alt.qcri.org/semeval2016/task9/](http://alt.qcri.org/semeval2016/task9/)
- Chinese Semantic Dependency Parsing

SemEval 2014 and 2015
- [http://alt.qcri.org/semeval2014/task8/](http://alt.qcri.org/semeval2014/task8/)
- [http://alt.qcri.org/semeval2015/task18/](http://alt.qcri.org/semeval2015/task18/)
- English Semantic Dependency Parsing
Graph-based Dependency Parsing

- Find the highest scoring tree from a complete dependency graph
- Maximum Spanning Tree (MST)
  - Some dynamic programming algorithms

\[ Y^* = \arg \max_{Y \in \Phi(X)} \text{score}(X, Y) \]
How to Calculate the Score of a Tree

- The score of a tree is the sum of each arc

\[ \text{score}(X,Y) = \sum_{(h,m) \in Y} \text{score}(X,h,m) \]

- An arc is represented as a feature vector

\[ \text{score}(h,m) = ? \]

- The score of the arc is dot product of weight vector by feature vector

\[ \text{score}(h,m) = \mathbf{w} \cdot \mathbf{f}(h,m) \]
Features for an Arc

As McGwire neared, fans went wild.
Greedily predict a transition action sequence from an initial parsing state to some terminal states

State (configuration)

= Stack + Buffer + Dependency Arcs
Traditional Features

Configuration

Stack
- ROOT
- has_VBZ
- good_JJ
- nsbj
- He_PRP

Buffer
- Control_NN

Need Tedious Feature Engineering!

Feature
- Binary
- Sparse
- High-dimensional

Feature templates: a combination of elements from the configuration.
- For example: (Zhang and Nivre, 2011): 72 feature templates

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

<table>
<thead>
<tr>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0wpw; S_0wp_r; S_0wpv; S_0wpv_l; N_0wpv; N_0pv_l; )</td>
</tr>
<tr>
<td>( S_0wp; S_0wp_r; S_0wpv; S_0wpv_l; N_0wpv; N_0pv_l; )</td>
</tr>
</tbody>
</table>

unigrams

| \( S_0wp; S_0wp_r; S_0wpv; S_0wpv_l; N_0wpv; N_0pv_l; \) |
| \( S_0wp; S_0wp_r; S_0wpv; S_0wpv_l; N_0wpv; N_0pv_l; \) |

third-order

| \( S_0wp_r; S_0wpv; S_0wpv_l; N_0wpv; N_0pv_l; \) |
| \( S_0wp_r; S_0wpv; S_0wpv_l; N_0wpv; N_0pv_l; \) |

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

- **Softmax layer:**
  \[ p = \text{softmax}(W_2h) \]

- **Hidden layer:**
  \[ h = (W^w_w x^w + W^t_t x^t + W^l_l x^l + b_1)^3 \]

- **Input layer:** \([x^w, x^t, x^l]\)

---

Global Normalization

Training with Beam Search

\[ p(y_i \mid x, \theta) = \frac{e^{f(x, \theta)_i}}{\sum_{y_j \in \text{GEN}(x)} e^{f(x, \theta)_j}} \]

\[ f(x, \theta)_i = \sum_{a_k \in y_i} o(x, y_i, k, a_k) \]

SyntaxNet: Google

Just optimize the likelihood of the head, no structured learning
This is a local model, with global decoding using MST at the end

Changes of Performance

Test on PTB with Stanford Dependency

- Zhang & McDonald (2014)
- Chen & Manning (2014)
- Dyer et al. (2015)
- Zhou et al. (2015)
- Andor et al. (2016)
- Dozat & Manning (2017)

UAS - LAS
[Yuxuan Wang, Wanxiang Che, Jiang Guo and Ting Liu. A Neural Transition-Based Approach for Semantic Dependency Graph Parsing. AAAI 2018.]
Transition System for Dependency Tree [Choi and McCallum (2013)]

ROOT

Exp
mDegr
mPunc
eCau
mNeg
mMod
Datv
ePurp

他 太 小气，不肯请 我们 吃饭

ROOT 他

小气，不肯请 我们 吃饭

LEFT-REDUCE

他

Exp

ROOT

小气，不肯请 我们 吃饭

2017-12-1

SIGHAN
Transition System for Dependency Graph

Transition System for Dependency Graph

Root

Agt

Exp

mDegr

mPunc

eCau

mNeg

mMod

Datv

ePurp

他 太 小气 , 不 肯 请 我们 吃饭

[ Yuxuan Wang, Wanxiang Che, Jiang Guo and Ting Liu. A Neural Transition-Based Approach for Semantic Dependency Graph Parsing. AAAI 2018. ]

2017-12-1

SIGKAN

35
Transition System for Dependency Graph
IT-BS Classifier

\[ e_t = \max \{0, W[e_t \oplus h_t \oplus p_t \oplus a_t] + d\} \]

\[ \sigma \]

No-Shift No-Pass

Incremental Tree-LSTM

Bi-LSTM Subtraction

\[ \beta \]

请(treat) 我们(us) …

他(he) 太(too) 请(treat) 我们(us)
Experiments on TEXT Corpus of SemEval 2016 Task 9

- Baseline
- BS
- IT
- Our Model (BS-IT)

2017-12-1 SIGHAN
Experiments on TEXT Corpus of SemEval 2016 Task 9 (Chinese)

- HIS-RD-Belarus
- 2-stage (Ding et al. 2014)
- Our Model

LF vs NLF
DL for NLP: End-to-End Learning

Traditional Parser

Stack-LSTM Parser
1. Syntactic and Semantic Parsing
2. Pseudo Data for Parsing
3. Applications of Parsing
4. Summary
DL for NLP: Representation Learning

Applications
- Word Seg
- POS tagging
- Parsing
- QA
- MT
- Dialogue
- RC
- Caption
-...

Semantic Vector Space

Deep Learning
- Recurrent NN
  - I like red apple
- Convolutional NN
  - I like red apple
- Recursive NN
  - I like red apple

Big Data
- Mono-lingual Data
- Multi-lingual Data
- Multi-modal Data
Pseudo Data for Parsing

Labeled Data

Heterogeneous Data

Multi-task Data

Multilingual Data

Multiple Modalities
Transfer the parser trained on source language(s) to parse a target language

How to overcome word inconsistency?

Source: I like playing basketball

Rich-resource source language

Target: W₁ W₂ W₃ W₄

Low-resource target language
Learn bilingual word embeddings to overcome word inconsistency

Published papers: ACL 2015, AAAI 2016, JAIR 2016, CoNLL 2017
Deep Multi-Task Learning Architecture

Each task corresponds to a Treebank
- Multilingual universal
- Monolingual heterogeneous
- Multiple NLP tasks

Core Parameters
- LSTM(B), LSTM(S)
- LSTM(A)
- BiLSTM(chars)
- RecNN
- $W_A, W_B, W_S$
- $E_{pos}, E_{char}, E_{rel}, E_{act}$
- $e^t$
- $g$

Outline

1. Syntactic and Semantic Parsing
2. Pseudo Data for Parsing
3. Applications of Parsing
4. Summary
The final task, e.g., entity relation extraction
The final task, e.g., entity relation extraction
## A Question

Is Parsing or Structure Necessary?

<table>
<thead>
<tr>
<th></th>
<th>Bi-LSTM</th>
<th>Tree-LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford Sentiment TreeBank</td>
<td>49.8 / 50.7 (Segment)</td>
<td>50.4</td>
</tr>
<tr>
<td>Binary Sentiment Classification</td>
<td>79.0</td>
<td>77.4</td>
</tr>
<tr>
<td>Question-Answer Matching</td>
<td>56.4</td>
<td>55.8</td>
</tr>
<tr>
<td><strong>Semantic Relationship Classification</strong></td>
<td><strong>75.2</strong></td>
<td><strong>76.7</strong></td>
</tr>
<tr>
<td>Discourse Parsing</td>
<td>57.5</td>
<td>56.4</td>
</tr>
</tbody>
</table>

[Jiwei Li, Minh-Thang Luong, Dan Jurafsky and Eduard Hovy. When Are Tree Structures Necessary for Deep Learning of Representations? EMNLP 2015]
Language Technology Platform (LTP)

- [http://ltp.ai](http://ltp.ai)
- Rich and accurate Chinese NLP toolkits
  - Chinese word segmentation,
  - POS tagging, NER, Dependency parsing,
  - Semantic role labeling, semantic dependency parsing
- Open source for research
- Evaluation
  - 1\textsuperscript{st} place/13 at CoNLL 2009: syntactic and semantic dependency parsing
  - 4\textsuperscript{th} place/33/113 at CoNLL 2017: multilingual syntactic dependency parsing
### 在线演示

![LTP Demo](#)

#### 标签名义

<table>
<thead>
<tr>
<th>标签名义</th>
<th>说明</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
<td>关系类型</td>
</tr>
<tr>
<td>A</td>
<td>agent</td>
</tr>
<tr>
<td>C</td>
<td>content</td>
</tr>
<tr>
<td>D</td>
<td>demonstrative</td>
</tr>
<tr>
<td>D</td>
<td>objective</td>
</tr>
<tr>
<td>D</td>
<td>direction</td>
</tr>
<tr>
<td>D</td>
<td>event</td>
</tr>
<tr>
<td>D</td>
<td>event Sucessor</td>
</tr>
<tr>
<td>F</td>
<td>location</td>
</tr>
<tr>
<td>F</td>
<td>manner</td>
</tr>
<tr>
<td>f</td>
<td>point marker</td>
</tr>
<tr>
<td>m</td>
<td>time marker</td>
</tr>
<tr>
<td>n</td>
<td>name modifie</td>
</tr>
<tr>
<td>p</td>
<td>product</td>
</tr>
<tr>
<td>r</td>
<td>root</td>
</tr>
</tbody>
</table>

在文档中查看全部标签信息
LTP-Cloud Service

- http://www.ltp-cloud.com/

- Advantages
  - Installation free, saving hardware, easy usage, cross-platform, cross-programming languages, update in time
import urllib2, urllib, sys
uri_base = 'http://api.ltp-cloud.com/analysis/?'
api_key = "YourAPIKey"
text = urllib.quote("我爱北京天安门")
format = sys.argv[1]
url = "{api_key}={text}={format}={pattern=all}".format(uri_base, api_key, text, format)
print urllib2.urlopen(url).read()
How to Use Tree or Graph Structures?

- As Information Extraction Rules
- As Input Features
- As Input Structures
- As Structured Prediction
How to Use Tree or Graph Structures?

- As Information Extraction Rules
- As Input Features
- As Input Structures
- As Structured Prediction
For example
- Polarity-target pair extraction

Problem
- The extraction rules are very complex
- The parsing results are inexact
Sentence compression based PT pair extraction
- Simplify the extraction rules
- Improve the parsing accuracy

Use a sequence labeling model to compress sentences
The PT pair extraction performance improves 3%

[Wanxiang Che, Yanyan Zhao, Honglei Guo, Zhong Su, Ting Liu. Sentence Compression for Aspect-Based Sentiment Analysis. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2015, 23(12)]
How to Use Tree or Graph Structures?

- As Information Extraction Rules
- As Input Features
- As Input Structures
- As Structured Prediction
Path Features

- For Example
  - Semantic Role Labeling (SRL), Relation Extraction (RC)

  ![Diagram](image)

  (a) Semantic Role Labeling.

  (b) Relation Classification.

- The parsing path features are very important
  - People <-> downtown: nsubj ← moved → nmod
- But they are difficult to be designed and very sparse
- Use LSTMs to represent paths
- All of word, POS tags and relations can be inputted

[Michael Roth and Mirella Lapata. Neural Semantic Role Labeling with Dependency Path Embeddings. ACL 2016]
The hidden units for parsing include **soft** syntactic information.

These can help applications, such as relation extraction.

---

How to Use Tree or Graph Structures?

- As Information Extraction Rules
- As Input Features
- As Input Structures
- As Structured Prediction
Recurrence Neural Networks
  - Composing sequentially

Recursive Neural Networks
  - Use parse trees as input structures
  - Composing according to parsing structures

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng and Christopher D. Manning. Parsing Natural Scenes And Natural Language With Recursive Neural Networks. ICML 2011.
Tree-LSTMs

- Standard LSTM

- Tree-LSTM

The deletion mutation on exon-19 of EGFR gene was present in 16 patients, while the L858E point mutation on exon-21 was noted in 10.

All patients were treated with gefitinib and showed a partial response.

How to Use Tree or Graph Structures?

- As Information Extraction Rules
- As Input Features
- As Input Structures
- As Structured Prediction
Disfluenecy Detection

- Disfluency detection for speech recognition

I want a flight [ to Boston + {um} to Denver ]

- Transition System < \( O, S, B, A \)>
  - output (\( O \)) : represent the words that have been labeled as fluent
  - stack (\( S \)) : represent the partially constructed disfluency chunk
  - buffer (\( B \)) : represent the sentences that have not yet been processed
  - action (\( A \)) : represent the complete history of actions taken by the transition system
    - OUT: which moves the first word in the buffer to the output and clears out the stack if it is not empty
    - DEL: which moves the first word in the buffer to the stack

[Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang and Ting Liu. Transition-Based Disfluency Detection using LSTMs. EMNLP 2017]
Outline

1. Syntactic and Semantic Parsing
2. Pseudo Data for Parsing
3. Applications of Parsing
4. Summary
Summary

- Syntactic and semantic parsing is one of the core tasks of NLP
- Recent advances
  - Grammar: universal dependency, semantic dependency graph
  - Data: large (pseudo) labeled data (multi-lingual/task, heterogeneous)
  - Algorithm: deep learning for semantic dependency graph parsing
- More and more applications
  - As Information Extraction Rules
  - As Input Features
  - As Input Structures
  - As Structured Prediction
Thanks!
http://ir.hit.edu.cn/~car/