

LING 520: Computational Analysis of English

Semester: FALL '16

Instructor: Sowmya Vajjala

Iowa State University, USA

3 November 2016

Class Outline

- ▶ Other parsing methods: Overview
- ▶ Parsing: Conclusion
- ▶ NLTK Exercises
- ▶ Announcement: Final project first report due on 5th November. Those who are still undecided: Please talk to me before tomorrow evening.

Partial Parsing

- ▶ Refers to grouping of word sequences together based on POS tags, without doing a full parse.
- ▶ Also known as "shallow parsing" or "chunking"
- ▶ NP chunking: process of identifying noun phrases (part of your assignment 5)
- ▶ Chinking: Removing a few unwanted tokens in a chunk.
- ▶ Purpose: Get something beyond POS tags, but not a full parse.
- ▶ Use: When parsers are not available for some custom dataset, or a new language etc.
- ▶ More information: Chapter 7, Section 2 in NLTK book.

Incremental Parsing

- ▶ Usually, a parser or a tagger works with full sentence representation and then go left to right, word by word.
- ▶ In incremental parsing, the parser tries to construct a syntactic representation of the sentence as soon as it sees each word.
- ▶ This comes from some psycholinguistics background, and usually seen as related to human sentence processing.

Incremental Parsing

- ▶ Usually, a parser or a tagger works with full sentence representation and then go left to right, word by word.
- ▶ In incremental parsing, the parser tries to construct a syntactic representation of the sentence as soon as it sees each word.
- ▶ This comes from some psycholinguistics background, and usually seen as related to human sentence processing.
- ▶ Advantage: the closeness to human sentence processing part, a different way of looking at syntactic structure.
- ▶ Disadvantage: Dead slow

Example Incremental Parser output

String: Anderson said: "The takeaway message is that guilty people fidget more and we can measure this robustly"

prefix	header	prefix	srprsl	SynSp	LexSp	ambig	open	rerank	toprr	stps
prefix:1	Anderson	13.551	13.551	1.595	11.956	0.928	1.66	1.00	0.61	3.7
prefix:2	said	18.481	4.850	3.185	1.666	0.019	2.00	1.64	1.64	3.0
prefix:3	:	24.062	5.660	5.636	0.024	0.019	2.00	1.00	1.00	1.0
prefix:4	"	40.802	16.740	12.613	4.127	1.567	4.04	0.87	0.45	3.0
prefix:5	The	45.809	5.007	2.839	2.168	1.889	4.39	0.08	0.00	3.1
prefix:6	takeaway	50.378	4.569	0.152	4.418	2.431	4.35	1.09	0.91	1.0
prefix:7	message	60.526	10.148	1.520	8.627	2.503	4.39	0.44	0.00	1.0
prefix:8	is	63.728	3.202	3.044	0.159	1.951	4.49	0.00	0.00	3.0
prefix:9	that	68.107	4.379	1.350	3.029	2.513	5.52	0.59	0.00	2.1
prefix:10	guilty	78.636	10.529	2.823	7.706	3.384	6.44	0.34	0.00	1.7
prefix:11	people	85.521	6.885	2.017	4.868	3.092	7.04	0.33	0.00	1.0
prefix:12	fidget	92.227	6.706	2.044	4.662	3.339	6.73	0.94	0.76	2.1
prefix:13	more	97.823	5.596	4.862	0.734	3.629	7.63	1.66	0.00	2.4
prefix:14	and	102.807	4.985	4.730	0.255	4.136	7.41	0.13	0.00	2.0
prefix:15	we	109.255	6.447	4.381	2.066	2.697	6.87	0.00	0.00	3.0
prefix:16	can	113.012	3.757	2.116	1.641	2.665	6.81	1.04	1.04	3.0
prefix:17	measure	122.448	9.436	0.403	9.033	2.666	7.81	1.00	1.00	2.0
prefix:18	this	128.761	6.313	1.713	4.600	3.388	9.12	1.00	0.72	2.3
prefix:19	robustly	137.365	8.604	0.232	8.372	5.178	9.49	0.79	0.23	1.8
prefix:20	.	140.285	2.920	2.915	0.005	4.519	2.44	0.10	0.00	7.5
prefix:21	"	161.548	21.262	17.135	4.127	3.640	4.35	0.09	0.00	5.4
prefix:22	</s>	167.439	5.891	5.891	0.000	2.878	0.00	1.40	0.00	5.0
prefix sent norm (tot words 21)			7.973	3.962	4.012	2.811	5.48	0.74	0.40	2.9
prefix sent norm (steps 66.4)			2.522	1.253	1.269	0.889	1.73	0.23	0.13	0.9

Full parses for string:

```
1 1 1.3696 168.8082 79.5237 89.2845 (TOP (S (S (NP (NNP Anderson)) (VP (VBD said) (: :) (S (X (SYM "")) (NP (DT The) (JJ takeaway) (NN message)) (VP (VBZ is) (SBAR (IN that) (S (S (NP (JJ guilty) (NNS people)) (VP (VBP fidget) (NP (JJR more)))) (CC and) (S (NP (PRP we) (VP (MD can) (VP (VB measure) (NP (DT this)) (ADVP (RB robustly)))))))))) ( . .))) (FRAG (X (SYM ""))))
```

Other Parsing Methods

based on different grammar formalisms

- ▶ Head-driven Phrase Structure Grammar parser
- ▶ Constraint Grammar, Construction Grammar based parsers
- ▶ Link Grammar parser
- ▶ Tree Adjoining Grammar parser
- ▶ Combinatory categorial grammar parser etc.

Parsing: Conclusion

Useful Resources:

- ▶ Theory: Chapters on parsing in J&M.
- ▶ Vague practice examples: NLTK Chapter 8 (and 7)
- ▶ NLTK code documentation and examples there (not textbook!)
- ▶ Explore other parsers in Python beyond NLTK
- ▶ Radev's or others' video lectures on parsing

NLTK Practice Exercises

Exercise-1

Do Section 1 –3 described in:

<http://coling.epfl.ch/TP/Parsing.html>

Exercise-2

How to use Stanford parser and Malt parser in Python with or without NLTK.

Useful Links:

1.

http://www.nltk.org/_modules/nltk/parse/stanford.html

2. http://www.nltk.org/_modules/nltk/parse/malt.html

Additional stuff: Notes I prepared on this. (on BB)

Exercise-3

Those who managed to install: Start figuring out what sort of outputs can you get with the parser(s). You can start thinking about your Assignment 5 last question too.

Next Week

1. Topics:

- ▶ Overview of Semantics and Discourse modeling in NLP
- ▶ Optional theoretical readings: Chapters 19–21 in J&M
- ▶ Practical Readings: Chapter 2 Section 5, Chapter 10 in NLTK (Chapter 10 is very theoretical according to me).
- ▶ Lectures: 3.1-3.4 in Radev's course (Week 3). 12.1-12.2 in Week 12.