Assignment 2 Tutorial

CSC485, Fall 2015

Krish Perumal
krish@cs.toronto.edu
(based on slides by Sean Robertson
(who based it on Katie Fraser's))

October 15, 2015

Part 1: Developing a grammar

- Develop a context-free grammar to model certain aspects of English syntax

- Applications
  - Corpus development
  - Grammars for specialized domains (a dialogue system for flight reservations)

The Most Important Thing

- Don’t be a hero!
  - Don’t try to make a universal grammar
  - If there is no example of it, it is not required
  - ~~Sean~~ I hate~~s~~ extra work

1.1. Simple sentences

- Sentences with intransitive verbs in simple past tense

- Start simple (pronouns and verbs)
  - *I ate*

- Then get slightly more complex (determiners and nouns)
  - *the elephant ate*

- Then add modifiers (adverbs and adjectives)
  - *the tall beautiful elephant ate slowly*

1.1. Simple sentences

- Then prepositions
  - *the elephant with the long trunk ate slowly*
  - *the cat in the parlour jumped*
  - Only modify noun phrases for now

1.2. Auxiliaries

- Modals + be, have
  (modals = should, could, would, etc.)

- Again, start with intransitive verbs

- Be careful of verb combinations and conjugations
  - *is leaving, will leave, has left, have been leaving*

1.2. Auxiliaries

- Don't have to handle passive voice

- Don't worry about auxiliary number agreement or subject-verb agreement
  - *the dogs has left*
  - Don't have to handle passive voice

1.2. Auxiliaries

- Present perfect tense
  - *I have eaten*

- Could be represented as:

```
S -> NP VP
NP -> Prp
VP_PresPerf -> AuxHavePres Vpp

AuxHavePres -> 'have' | 'has'
Vpp -> 'eaten'
Prp -> 'I' | 'she'
```

1.3. Subcategorization

- Verbs take different combinations of complements

- This depends on the verb itself
  - *she arrived*
  - *Nadia fondled the eggplant*
  - *the eggplant reminded Nadia*

- This is ungrammatical
  - **the eggplant reminded*

1.3. Subcategorization

- You will use regular context-free rules, not features, to handle these differences

- You can emulate features, though

```
VP -> V
VP -> V_NP NP

V -> 'jumped'
V_NP -> 'saw'
```

1.3. Subcategorization

- Sometimes one verb can take different numbers of complements
  - Intransitive: *Nadia ate*
  - (Mono)transitive: *Nadia ate* ***the pie***

- … or different kinds of complements
  - NP: *they told her* ***a secret***
  - Clause: *they told her* ***to go***

1.3. Subcategorization

- Such verbs should be listed multiple times in your lexicon

```
V -> 'ate' |  ...
V_NP -> 'ate' | ...
```

Nouns and verbs

- Some words act as nouns and verbs
  - *she jumped the jump*

- List them multiple times in your lexicon with each role

Getting started

- Use Sean's awesome code
  - See course website for zip archive

- Unpack zip contents:
  - `generate_tests.py, q1utils.py, unittest_prefix.py`
  - `Lexicon`
  - `Grammar`
  - `Sentences`

Getting started

- Run:

```
python generate_tests.py
```

This will create a bunch of unit tests (1 for each sentence in `Sentences`) saved to `tests.py`

- Run

```
python tests.py
```

This will create a bunch of parse trees (1 for every sentence that should have been parsed) saved to `ParseTrees`

Format

- `Grammar` contains rules to non-terminals

```
S -> NP VP
```

- …and `Lexicon` contains rules to terminals

```
Det -> 'a' | 'an' | …
```

- `Sentences` contains test sentences

– *Nadia fondled the eggplant*

- **Ross brought to him*

- Please keep these organized with spaces and comments

Testing

- Test on a number of sentences
  - Provided in assignment handout
  - Provided on course website

http://www.cs.toronto.edu/~frank/csc2501/Assignments/A2-test.txt

  - Created by you
    -- Try generating random sentences, guided by your intuition on what should/should not work

Testing

- I will test on a bunch of private sentences (similar to those given in the assignment)
  - no new grammatical constructs

- Only need to cover vocabulary from http://www.cs.toronto.edu/~frank/csc2501/Assignments/A2-vocab.txt

- Describe and give examples of
  - Overgeneration: Parsed when shouldn't
  - Undergeneration: Didn't parse when should

Testing

- If CDF's NLTK cannot parse your grammar with `parse_cfg`, you will receive a zero for your grammar (15/30 marks!)

Part 2: Features in Grammars

- Features can handle non-local dependencies in syntax gracefully

- Determine the intended features for the given grammar

Part 2: Features in Grammars

- Without features:

S -> NP VP

VP -> V NP

- Admits:
  - **I sees him, *they sees her, *he see those apples*

Part 2: Features in Grammars

- Lexical feature specification

he [Agr 3s]; they [Agr 3p]
sees [Agr 3s]; see [Agr 3p]

- Update rules with agreement conditions

S -> NP VP
(NP Agr) = (VP Agr)
VP -> V NP
(VP Agr) = (V Agr)

Syntactic case

- Model syntactic case with features
  - **He sees she, *Her sees him*

- Nominative case: the subject
  - **She** sees him

- Accusative case: the object
  - He sees **her**