Task #8

- WordNet is nifty
- NLTK is a good failsafe
- WordNet FAQ: “we were not well disciplined in building our vocabulary”
- “These models are pretty flawed, while providing some useful information.”
- Vector math:
  - on Twitter-trained corpus: human – ethics = monster
  - on big GloVe resource:
    - catholic – baptist = roman
    - buddhist – agnostic = monk
    - mormon – baptist = polygamist
    - obama – biden = bush

- horse --> steed 0.5463667125516541
- hot --> cold 0.59720458733136
- hand --> foot 0.48578523756174047
- bone --> muscle 0.5473219940821349
- head --> helmet 0.37887525659600746
- donkey --> mule 0.592143722977295

Coptic - Egypt + Asia = Syriac/Tibetan
Swahili - Africa + Europe = Shtokavian
Brain - Neuron + Heart ~ arteries
Beowulf - Anglo-saxon + Greek = Hesiod (also homeric)
JavaScript - Beginner + Advanced = EcmaScript
Physics - Math + Reading = Historiography
India - Hindu + Muslim = Arabia (also Pakistan was close)
DEPENDENCY GRAMMAR

Another way to look at trees
Dependency Grammar/Parsing

• A sentence is parsed by relating each word to other words in the sentence which depend on it.

• The idea of dependency structure goes back a long way
  • To Pāṇini’s grammar (c. 5th century BCE)

• Constituency is a new-fangled invention
  • 20th century invention

• Modern work often linked to work of L. Tesnière (1959)
  • Dominant approach in “East” (Eastern bloc/East Asia)

• Among the earliest kinds of parsers in NLP, even in US

• Still widely used today
The basic idea

- A syntactic structure consists of:
  - Lexical items
  - Dependency relations

- Translated words of Lucien Tesnière (1959):

The sentence is an organized whole, the constituent elements of which are words. Every word that belongs to a sentence ceases by itself to be isolated as in the dictionary. Between the word and its neighbors, the mind perceives connections, the totality of which forms the structure of the sentence. The structural connections establish dependency relations between the words.
Dependency structure

- Dependency structure shows which words depend on (modify or are arguments of) which other words.
Treebank vs. dependency parse

Estimated volume was a light 2.4 million

(Penn Treebank)
Two homologues of the rhombotin gene have now been isolated.
Dependency grammar

• Hierarchical structure is not as important
• What is: how words depend on each other
  • modification, arguments, etc.
• Draw directional arrows between words to represent dependencies
• Useful for when full-blown structure isn’t required
  • E.g. computer implementations; > 90% Eng.
Dependency Relations

• A -> B means A governs B or B depends on A

• Dependency relations can represent:
  • Syntactic properties
  • Semantic properties, such as:
    • Subject/Agent
    • Object/Patient

• A dependency relation is binary and directional
Robinson’s Axioms (1970)

• A dependency graph is a tree:
  1. One and only one element is independent.
  2. All others depend directly on some element.
  3. No element depends directly on more than one other.

• The *projectivity* axiom:

  If A depends directly on B and some element C intervenes between them (in the linear order of the string), then C depends directly on A or B or some other intervening element.
Dependency theories

• Long history, particularly for Slavic languages
• Take a more functional (less structural) view of syntax
• The basic rules:
  • H determines the category of C (syn and sem)
  • H is required; D is optional
  • H selects D (required or optional)
  • The form of D depends on H
  • Linear position of D is specified w/rt H
• Head-specifier, head-modifier, head-complement and lots of other relations
Dependency parsers

- Inter-word dependencies are the principal features, not structure
- No large superstructure, attendant decisions
- All relations grounded directly in words
- Generally more robust
- Often used in IR, partial parsing, shallow parsing, Q/A
Estimated volume was a light 2.4 million onces.
Varieties of Dependency Grammar

- Functional Generative Description (FDG)
- Dependency Unification Grammar (DUG)
- Meaning Text Theory (MTT)
- Word Grammar (WG)
- Functional Dependency Grammar (FDG)
- Bröker
- Extensible Dependency Grammar (XDG)
Economic news had little effect on financial markets.
Figure 3: Two analyses of coordination
The last epoch of the Cumean Song has now arrived.
Sharing arguments

(NP-SBJ (JJ massive) (JJ internal) (NN debt))
  (VP (VBZ has))
  (VP (VBN forced))
  (S
    (NP-SBJ-1 (DT the) (NN government))
    (VP
      (VP (TO to))
      (VP (VB borrow))
      (ADVP-MNR (RB massively))...)

```
force
  /   
arg0  arg1
  /       \    \  
massive internal debt the government borrow
  |        |    |   |
arg0  arg0 MNR massively
```
Bills were submitted by Brownback on ports, and immigration of Kansas.

Bills were submitted by Brownback on ports and immigration of Kansas.
Prague Dependency Bank

annotation on word level: lemmata, morphology
syntactic functions
dependency structure

semantic information
on constituent roles,
theme/rheme, etc.
Advantages of DG

• Closely related to semantics
• Easy to parse
• Can handle arbitrary word order (if non-projective relations are allowed)
• Closely tied to the lexical items
• Similar to human language processing???
Universal dependencies

- **Project** to develop cross-linguistically consistent treebank annotation for many languages
- Goal: facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective
- Based on Stanford dependency annotation scheme
- Develop a universal inventory of categories and guidelines to facilitate consistent annotation of similar constructions across languages, while allowing language-specific extensions when necessary.
- Lots of languages
Sample training data

# visual-style 4 2 nsubj:pass color:blue
# visual-style 4 7 obl color:blue
1 The dog DET - Definite=Def|PronType=Art 2 det - -
2 was be AUX - Mood=Ind|Number=Sing|Tense=Past|VerbForm=Fin 4 aux:pass - -
3 chased chase VERB - Tense=Past|VerbForm=Part 0 F0OT - -
4 by by ADP - 7 case - -
5 the cat DET - Definite=Def|PronType=Art 7 det - -
6 . . PUNCT - 4 punct - -
0

# visual-style 3 1 nsubj:pass color:blue
# visual-style 3 5 obl color:blue
1 Koniec name NOUN - Definite=Def|Gender=Neut|Number=Sing 3 nsubj:pass - -
2 od ol PRON - Case=Acc|PronType=Pr|Reflex=Yes 3 expl:pass - -
3 mpoezaas masks mpeezas mask VERB - Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin 4 aux:by - -
4 by by ADP - 5 case - -
5 nacha nacha NOUN - Definite=Def|Gender=En|Number=Sing 3 obl - -
6 . . PUNCT - 3 punct - -
0

# visual-style 3 1 nsubj:pass color:blue
# visual-style 3 4 obl color:blue
1 Pes pes NOUN - Animacy=Anim|Case=Non|Gender=Masc|Number=Sing 3 nsubj:pass - -
2 byl byt AUX - Aspect=Imp|Gender=Masc|Number=Sing|Tense=Past|VerbForm=Part|Voice=Act 3 aux:pass - -
3 honit honit VERB - Aspect=Imp|Gender=Masc|Number=Sing|VerbForm=Part|Voice=Pass 0 root - -
4 kočku kočka NOUN - Case=Ins|Gender=Fem|Number=Sing 3 obl - -
5 . . PUNCT - 3 punct - -
0
Nifty online tools

- **Web interface to Stanford coreNLP**
  - POS tags, PTB constituent parse, dependency parse
  - for English, Arabic, Chinese, French, Spanish
  - Configure a pipeline
  - Regex matching (semgrex, tregex)
- **brat corpus annotation tool**
- **displaCy Dependency Visualizer**
- **The VISL dependency parser**
- **CoNLL-U viewer**
- **Treebank viewer**
- Lots more…
Robust parsing

• Most parsers based on linguistic theory
• Linguistic theories assume grammatical input
• Much of language use not entirely grammatical
  • L2 English
  • Spoken language
  • Controlled language / sublanguage
  • Headlines
  • PowerPoint slide bullets
• Traditional parsers often don’t handle these types of language well
The LG parser

- Freely available for research purposes
- Robust (e.g. information retrieval, MT)
- Calculates simple, explicit relations
- Fast
- Written in C
- Sometimes more appropriate for large-scale task than traditional phrase-structure grammars
- Online parser

- Similar to, but technically not, a dependency parser
Exploring Link Grammar

• What is a link?
  • Two parts, + and –
  • Shows a relationship between pairs of words
    • Subject + verb
    • Verb + object
    • Preposition + object
    • Adjective + adverbial modifier
    • Auxiliary + main verb
  • Labels each relationship
• Potential links are specified by technical rules
• Possible to score linkages, penalize links
Sample LG parses

LEFT-WALL George Smith was v at Dartmouth in 1685.

LEFT-WALL Richard Smith, Boston 1657, had v wife n Sarah, probably widow n of John Strange.

LEFT-WALL James Smith, Windsor 1643, was v a poor a, thievish[?] a servant n.

LEFT-WALL John Smith, Plymouth, was v killed v by a cartwheel n, June 1661.
He was killed by the Indians 15 March 1698.
Grading EFL essays

Linkage 1, cost vector = (UNUSED=0 DIS=2 AND=0 LEN=23)

LEFT-WALL I.p 've been.v majoring.v in Material engineering.n at my University in Korea.

Linkage 1, cost vector = (UNUSED=4 DIS=0 AND=0 LEN=11)

LEFT-WALL the class.n [most] [important] is.v Mathematical [for] [my].
LEFT-WALL during my schooling. I have taken many classes.
LG example parses

Linkage 1, cost vector = (UNUSED=0 DIS=2 AND=0 LEN=23)

```
+-----------------------------------------Xp----------------------------------------+
|                         +-----------------------MVp-----------------------+       |
|                         +---------------MVp--------------+                |       |
|                         |      +-------Jp-------+        +----Js---+      |       |
+--Wd--+Sp*+-PPf-+--Pg*b--+--MVp-+     +----AN----+        |  +---D--+      +-Js+   |
      |   |     |        |      |     |          |        |  |      |      |   |   |
LEFT-WALL I.p 've been.v majoring.v in Material engineering.n at my University in Korea.
```

```
+----------------------------------------------Xp----------------------------------------------+
|      +-----------Wdc-----------+               +------------------Opt-----------------+      |
|      |      +--------CO--------+               |        +--------------AN-------------+      |
|      |      |     +-----D*u----+-------Ss------+        |            +-------AN-------+      |
+--Wc--+      |     |     |      |      |    |   |        |            |      |         |      |
|      |      |     |     |      |      |    |   |        |            |      |         |      |
LEFT-WALL but probably the best.a class.n for.p me was.v medicine.n and first.n aid.n principles.n.
```
LG parser’s robustness (1)

Linkage 1, cost vector = (UNUSED=4 DIS=0 AND=0 LEN=11)

+--------------------------------Xp-------------------------------+
|                  Wd                   |                     |
| +-----D*u-----Ss---------Ost--------- |                     |
| |                       |                     |
LEFT-WALL the class.n [most] [important] is.v Mathematical [for] [my].
LG parser’s robustness  (2)

Linkage 1, cost vector = (UNUSED=1 DIS=0 AND=0 LEN=17)

+----------------------------------Xp----------------------------------+
|                       +--------------MVp-------------+               |
|             +----I----+------MVp------+              +----Js----+    |
|+------Wi-----+-Ox-+    +---Op--+-       +--Jp--+       |    +--Ds-+    |
|  |    |    |       |       |      |       |    |     |    |
|+-----Wi-----+Ox++  ++--Op++  ++--Jp++  |  ++--Ds+  |
|  |    |    |       |       |      |       |    |     |    |
|LEFT-WALL [it] help.v me make.v friends.n with people.p around the world.n
Syntax isn’t enough

Linkage 1, cost vector = (UNUSED=0 DIS=1 AND=0 LEN=13)

LEFT-WALL the practice in English is progressing in the life.
Sample LG rule entries

words/words.y: % year numbers
NN+ or NIa- or AN+ or MV- or ((Xd- & TY- & Xc+) or TY-)
or ({EN- or NIc-} & (ND+ or OD- or {{@L+} & DD-} & ([Dmcn+] or ((<noun-sub-xnoappositive> or TA-) & (JT- or IN-
or <noun-main-xnoyear>)))))

<vc-fill>: ((K+ & {[@MV+]} & O*n+) or ({O+ or B-} & {K+}) or
[@MV+ & {Xc+} & O*n+] & {Xc+} & {MV+};
Sample link specifications

<pref-asp1>: {{PRF- or STV- or PRG-}};
pref-asp2>: {HAB-} & {DUB-} & {AD-};
predprefs>: {NZ-} & {<pref-asp1>} & {SX-} & {<pref-asp2>} & {{FUT- or PT-}};

<root-main>:  <predprefs> & {DT-} & {LX+} & {BNF+} & {TX+} & {TC+} & {ACH+} & {TC+} & {TX+} & {ASP+};
<main-args>: {P-} & {GEN-} & {WH-} & {S0s+} & {MV+};

<root-ditrx>: <predprefs> & {DT-} & {LX+} & {BNF+} & {TX+} & {TC+} & {ACH+} & {TC+} & {TX+} & {ASP+};
<ditrx-args>: {P-} & {GEN-} & {WH-} & {S0s+} & {EX+} & {S0o+} & {MV+};

<root-middle>: <predprefs> & {DT-} & {LX+} & {BNF+} & {TX+} & {TC+} & {MD+} & {ACH+} & {TC+} & {TX+} & {ASP+};
<middle-args>: {P-} & {GEN-} & {WH-} & {((PA+} & {EM+}) or {EM+} & {PA+})} & {MV+};

<pred1>: {{<root-main> & <main-args>} or
 (<root-middle> & <middle-args>) or
 (<root-ditrx> & <ditrx-args>)
 ) & {Wd-};
Children of JAMES HARWOOD, No. 103.

229. MYRA, born July 26, 1835, in Eden, Vt. She married ELIJAH SPENCER, Dec. 25, 1851. They had five children: Arvilla, born in 1852, is not living; Mariette, born Dec. 25, 1854, married Jonathan Snyder, have a family; Leverett, born Feb. 6, 1857, married Cora Smith, Nov. 2, 1879, had two children, Perry F. and Ida I. Leverett died May 21, 1910; Rosa E., born Jan. 13, 1860, married Emmett Byers, and have children; and Harrison, born about 1862, is not living. Elijah Spencer died in the Union army in 1863, and his widow married JONATHAN SQUIRES, who was born in Ohio, July 25, 1823, by whom she had one son, J. Wilbur, born June 16, 1865, in DeKalb county, Ind., married Cora M. Thomas, Aug. 24, 1887, they reside in St. Joseph, Mich., five children. Mrs. Myra Squires died in Allen county, Ind., Feb. 13, 1874.
Thomas Smith, Haverhill, married at Andover 6 January 1659, Unice Singletary of Salisbury.
Sample link parse

He was killed by the Indians 15 March 1698.

he was.v killed.v by the Indians.n 15 March 1698.
Mary married I think, 23 November 1661, Samuel Gay.
No complete linkages found.

Mary married.v [I] [think] [,] 23 November 1661 , Samuel Gay .
Link interpretation operators

1. (Xp) sentence punctuation: sentence is declarative
2. (Wd) first word: no semantic contribution
3. (G) multi-word expression: proper noun is entity
4. (SS) associate subject with verb
5. (MX) postmodifier, perhaps appositive: not processed
6. (Xd) punctuation: no semantic contribution
7. (Xc) punctuation: no semantic contribution
8. (Pv) passive: associate auxiliary with main verb
9. (MVP) postmodifier: *by*-phrase; assume passive agent
10. (MVp) postmodifier: time expression
11. (Xc) punctuation: no semantic contribution
12. (Js) associate preposition with its object
13. (Ds) determiner: indefinite
14. (TY) year
Link interpretation operators

1. (Xp)  sentence punctuation: sentence is declarative
2. (Wd)  first word: no semantic contribution
3. (G)   multi-word expression: proper noun is entity
4. (SG)  associate subject with verb
5. (MX)  postmodifier, perhaps appositive: not processed
6. (Xd)  punctuation: no semantic contribution
7. (Xc)  punctuation: no semantic contribution
8. (Pv)  passive: associate auxiliary with main verb
9. (MVP) postmodifier: *by*-phrase; assume passive agent
10. (MVP) postmodifier: time expression
11. (Xc)  punctuation: no semantic contribution
12. (Js)  associate preposition with its object
13. (Dx)  determiner: indefinite
14. (TY)  year

LEFT-WALL John Smith, Plymouth, was v killed v by a cartwheel n, June 1661.
Deeper interpretation

- Arguments for passive
- Multi-word expressions
- Create and output object, event, modifier predicates

15. (resolve-passive)  subject -> object, oblique agent -> subject
16. (build-mw)  build multi-word expression
17. (generate-objects)  output object predicates: names & variables
    John_Smith(x1)
    cartwheel(x2)
    June_1661(x3)
18. (generate-events)  output event predicates: names & variables
    killed(e1,x2,x1)
19. (generate-modifiers)  output modification predicates: names & variables
    TIME(e1,x3)
21. (halt)
Sample LG parse and derived predicates

LEFT-WALL John Smith, Plymouth, was.v killed.v by a cartwheel.n, June 1661.

John_Smith(x1)
cartwheel(x2)
June_1661(x3)
killed(e1, x2, x1)
TIME(e1, x3)
Sample LG parse and derived predicates

LEFT-WALL George Smith was v at Dartmouth in 1685.

george_smith(x) & dartmouth(y) & at(x, y)
Persian example (1)

من رفتم "I went"

```
+-------Spn1-------+
|       +----VMP----|
|       +----VMT+   +-RW+
|       |    |     |   |
mn.pn rf.v t.vmt m.vmp .
```
Persian example (2)

<tu midAni kh mn mirum>
“you know that I am going”
LG parsing (Arabic)
An Arabic LG parse (ambiguous)

غير أحمد حسين

gyr > Hnd. Hayn

Found 3 linkages

Linkage 1, cost vector = (UNUSED=0 DIS=1 AND=0 LEN=0)

+---Sn---0---+
|     |     |
gyr.PV > Hnd. Mprop Hayn. Mprop

Linkage 2, cost vector = (UNUSED=0 DIS=2 AND=0 LEN=0)

+---On---O---+
|     |     |
gyr.PV > Hnd. Mprop Hayn. Mprop

Linkage 3, cost vector = (UNUSED=0 DIS=2 AND=0 LEN=1)

+---Sn---+---On---+
|     |     |     |
gyr.PV > Hnd. Mprop Hayn. Mprop