LING 581: Advanced Computational Linguistics

Lecture Notes
February 26th
Adminstrivia

• No lecture next week

• Consequence:
  – extra time to produce that computationally *heavy duty* graph ...

• Reminder:
  – Resubmit EVALB results from section 23 on Bikel Collins
Homework from last time

• **Question:**
  – How much training data do you need?

• **Homework:**
  – How does Bikel Collins vary in precision and recall on test data if you randomly pick 1..24 out of 25 sections to do the training with?

• **Test section:**
  – I want you to pick a test section that’s not section 23 (*because that one was used before*)

• **Use EVALB**
  – plot precision and recall graphs
Sample submitted curves

- Recall/Precision (F) \( \uparrow \)
- Amount of training data \( \Rightarrow \)
Today’s Topic

- WordNet 3.0
  - (3.1 the latest version but only online or the database files only)
  - [http://wordnetweb.princeton.edu/perl/webwn](http://wordnetweb.princeton.edu/perl/webwn)
  - **Homework**: do the installations and use the programs in this lecture...
WordNet 3.0

- http://wordnet.princeton.edu/
WordNet 3.0

• Download:

**WordNet 2.1 for Windows**
WordNet browser, command-line tool, and database files with InstallShield self-extracting installer:
Download: WordNet-2.1.exe

**WordNet 3.0 for UNIX-like systems (including: Linux, Mac OS X, Solaris)**

**Prolog version of WordNet 3.0**
ANSI Prolog version of the WordNet database.
Download: WNprolog-3.0.tar.gz
WordNet 3.0

• **Steps**
  – Download.. from website

  ![Image of a file manager showing WordNet-3.0 and WordNet-3.0.tar]

  – **README**
    • Beginning with Version 2.1, we changed the Unix package to a GNU Autotools package.
    • The WordNet browser makes use of the open source Tcl and Tk packages. Tcl and Tk must be installed BEFORE you compile WordNet.
    • You must also have a C compiler before installing Tcl/Tk or WordNet. WordNet has been built and tested with the GNU gcc compiler.
WordNet 3.0

• Steps
  – GNU Autoconf
  – configure
    cd Downloads/WordNet-3.0
    ./configure
    • checking for gcc... gcc
    • config.status: creating Makefile
    • checking for Tcl configuration... configure: WARNING: Can't find Tcl configuration definitions
  – INSTALL
    If you're running OS X and installed the Aqua Tcl/Tk package from the web site above, use the following settings:
      --with-tcl=/System/Library/Frameworks/Tcl.framework
      --with-tk=/System/Library/Frameworks/Tk.framework
    ./configure
    make
    /usr/bin/make
  
OSX 10.9

need to have Xcode installed on Mac
WordNet 3.0

- **Steps**
  - **GNU Autoconf**
    ```
    ./configure -with-tcl=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.9.sdk/System/Library/Frameworks/Tcl.framework/ -with-tk=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.9.sdk/System/Library/Frameworks/Tk.framework/
    checking for gcc... gcc
    ...
    checking for Tcl configuration... found /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.9.sdk/System/Library/Frameworks/Tcl.framework/tclConfig.sh
    checking for Tk configuration... found /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.9.sdk/System/Library/Frameworks/Tk.framework/tkConfig.sh
    checking for existence of /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.9.sdk/System/Library/Frameworks/Tcl.framework/tclConfig.sh... loading
    checking for existence of /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.9.sdk/System/Library/Frameworks/Tk.framework/tkConfig.sh... loading
    configure: creating ./config.status
    config.status: creating Makefile
    ```
WordNet 3.0

Configure report:
• WordNet is now configured

• Installation directory: /usr/local/WordNet-3.0

• To build and install WordNet:
  • make
  • make install

• To run, environment variables should be set as follows:
  • PATH - include ${exec_prefix}/bin
  • WNHOME - if not using default installation location, set to /usr/local/WordNet-3.0

• See INSTALL file for details and additional environment variables
• which may need to be set on your system.
WordNet 3.0

- GNU Autoconf
- make

```
In file included from tkAppInit.c:16:
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/S
DKs/MacOSX10.9.sdk/usr/include/tk.h:78:11: fatal error: 'X11/Xlib.h' file not found
#
  include <X11/Xlib.h>

1 error generated.
```

Recall: from your EVALB compilation a .h file means a C compiler header file
X11 is now under /opt/X11,
include files are in /opt/X11/include
WordNet 3.0

• GNU Autoconf
• configure
  
  ./configure -with-tcl=/System/Library/Frameworks/Tcl.framework/ -with-tk=/System/Library/Frameworks/Tk.framework/ CFLAGS=-I/opt/X11/include

• make
  
gcc -I/opt/X11/include -o wishwn wishwn-tkAppInit.o wishwn-stubs.o -L../lib -lWN -F/System/Library/Frameworks -framework Tk -F/System/Library/Frameworks -framework Cocoa -lpthread -framework CoreFoundation -framework Carbon -framework IOKit -lpthread -framework CoreFoundation
WordNet 3.0

- **GNU Autoconf**
- **sudo make install**

```
...  
Making install in src
  test -z "/usr/local/WordNet-3.0/bin" || /Users/sandiway/Downloads/WordNet-3.0/install-sh -d "/usr/local/WordNet-3.0/bin"
    /usr/bin/install -c 'wn' "/usr/local/WordNet-3.0/bin/wn"
    /usr/bin/install -c 'wishwn' "/usr/local/WordNet-3.0/bin/wishwn"
  test -z "/usr/local/WordNet-3.0/bin" || /Users/sandiway/Downloads/WordNet-3.0/install-sh -d "/usr/local/WordNet-3.0/bin"
    /usr/bin/install -c 'wnb' "/usr/local/WordNet-3.0/bin/wnb"
```

- **Running the WordNet browser:**
  - `/usr/local/WordNet-3.0/bin/wnb`
  - `/usr/local/WordNet-3.0/bin/wnb: line 3: wishwn: command not found`
WordNet 3.0

- PATH to wnb (WordNet browser)
  
  printenv PATH
  - /usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/opt/X11/bin
  export PATH=/usr/local/WordNet-3.0/bin:$PATH
  which wnb
  - /usr/local/WordNet-3.0/bin/wnb
  wnb
requires platform packages that have gcc
make configure TCL/TK
WordNet 3.1 vs 3.0

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: obama

Display Options: [Select option to change] ▼ ▼  Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relation
Display options for sense: (gloss) "an example sentence"

Noun

• S: (n) Obama, Barack Hussein Obama (44th President of the United States; first African–American President)
Perl Interface to WordNet 3.0

- http://www.cpan.org/
Perl Interface to WordNet 3.0

- http://search.cpan.org/~jrennie/WordNet-QueryData-1.49/QueryData.pm
Perl Interface to WordNet 3.0
Perl Interface to WordNet 3.0

cd WordNet-QueryData-1.49
perl Makefile.PL
  – Checking if your kit is complete...
  – Looks good
  – Writing Makefile for WordNet::QueryData
make
• cp QueryData.pm blib/lib/WordNet/QueryData.pm
• Manifying blib/man3/WordNet::QueryData.3pm
Perl Interface to WordNet 3.0

make test
• PERL_DL_NONLAZY=1 /usr/bin/perl "-llib/lib" "-llib/arch" test.pl
• ok 1
• ok 2
• ok 3
• ok 4
• ok 5
• ...
• ok 104
• ok 105
• ok 106
• ok 107
• ok 108
• ok 109
Perl Interface to WordNet 3.0

```bash
sudo make install
```

- Password:
- Installing `/Library/Perl/5.16/WordNet/QueryData.pm`
- Installing `/usr/local/share/man/man3/WordNet::QueryData.3pm`
- Appending installation info to `/Library/Perl/Updates/5.16.2/darwin-thread-multi-2level/perllocal.pod`
Perl Interface to WordNet 3.0

NAME

WordNet::QueryData - direct perl interface to WordNet database

SYNOPSIS

```perl
use WordNet::QueryData;

my $wn = WordNet::QueryData->new( noload => 1);

print "Synset: ", join(", ", $wn->querySense("cat#n#7", "syns")), "\n";
print "Hyponyms: ", join(", ", $wn->querySense("cat#n#1", "hypo")), "\n";
print "Parts of Speech: ", join(", ", $wn->querySense("run")), "\n";
print "Senses: ", join(", ", $wn->querySense("run#v")), "\n";
print "Forms: ", join(", ", $wn->validForms("lay down#v")), "\n";
print "Noun count: ", scalar($wn->listAllWords("noun")), "\n";
print "Antonyms: ", join(", ", $wn->queryWord("dark#n#1", "ants")), "\n";
```
Perl Interface to WordNet 3.0

perl wnqdata.perl
print "Synset: ", join(" ", $wn->querySense("cat#n#7", "syns")), " \n";
  • Synset: big_cat#n#1, cat#n#7
print "Hyponyms: ", join(" ", $wn->querySense("cat#n#1", "hypo")), " \n";
  • Hyponyms: domestic_cat#n#1, wildcat#n#3
print "Parts of Speech: ", join(" ", $wn->querySense("run")), " \n";
  • Parts of Speech: run#n, run#v
print "Senses: ", join(" ", $wn->querySense("run#v")), " \n";Senses:
  • run#v#1, run#v#2, run#v#3, run#v#4, run#v#5, run#v#6, run#v#7, run#v#8, run#v#9, run#v#10, run#v#11, run#v#12, run#v#13, run#v#14, run#v#15, run#v#16, run#v#17, run#v#18, run#v#19, run#v#20, run#v#21, run#v#22, run#v#23, run#v#24, run#v#25, run#v#26, run#v#27, run#v#28, run#v#29, run#v#30, run#v#31, run#v#32, run#v#33, run#v#34, run#v#35, run#v#36, run#v#37, run#v#38, run#v#39, run#v#40, run#v#41
Perl Interface to WordNet 3.0

perl wnqdata.perl
print "Forms: ", join(" ", $wn->validForms("lay down#v")), "\n";
• Forms: lay_down#v, lie_down#v
print "Noun count: ", scalar($wn->listAllWords("noun")), "\n";
• Noun count: 117798
print "Antonyms: ", join(" ", $wn->queryWord("dark#n#1", "ants")), "\n";
• Antonyms: light#n#9
wnb

- WordNet browser:

```
The noun world has 8 senses (first 7 from tagged texts)

1. (49) universe, existence, creation, world, cosmos, macrocosm -- (everything that exists anywhere; "they study the evolution of the universe"; "the biggest tree in existence")
2. (34) world, domain -- (people in general; especially a distinctive group of people with some shared interest; "the Western world")
3. (31) world, reality -- (all of your experiences that determine how things appear to you; "his world was shattered"; "we live in different worlds"; "for them demons were as much a part of reality as trees were")
4. (26) Earth, earth, world, globe -- (the 3rd planet from the sun; the planet we live on; "the Earth moves around the sun"; "he sailed around the world")
5. (16) populace, public, world -- (people in general considered as a whole; "he is a hero in the eyes of the public")
6. (6) world -- (a part of the earth that can be considered separately; "the outdoor world"; "the world of insects")
7. (5) worldly concern, earthly concern, world, earth -- (the concerns of this life as distinguished from heaven and the afterlife; "they consider the church to be independent of the world")
8. world, human race, humanity, humankind, human beings, humans, mankind, man -- (all of the living human inhabitants of the earth; "all the world loves a lover"; "she always used 'humankind' because 'mankind' seemed to slight the women")

The adj world has 1 sense (first 1 from tagged texts)

1. (1) global, planetary, world, worldwide, world-wide -- (involving the entire earth; not limited or

Overview of world```
wnb

• Setup:
  – export PATH=/usr/local/WordNet-3.0/bin:$PATH
  – wnb
  – Options: see sense numbers
    (we’ll need these later)
• Hypernums (going up the hierarchy):

<table>
<thead>
<tr>
<th>Sense 1</th>
<th>Sense 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat#1, true cat#1</td>
<td>guy#1, cat#2, hombre#1, bozo#2</td>
</tr>
<tr>
<td>=&gt; feline#1, felid#1</td>
<td>=&gt; man#1, adult male#1</td>
</tr>
<tr>
<td>=&gt; carnivore#1</td>
<td>=&gt; male#2, male person#1</td>
</tr>
<tr>
<td>=&gt; placental#1, placental mammal#1, eutherian#1, eutherian mammal#1</td>
<td>=&gt; person#1, individual#1, someone#1, somebody#1, mortal#1, soul#2</td>
</tr>
<tr>
<td>=&gt; mammal#1, mammalian#1</td>
<td>=&gt; organism#1, being#2</td>
</tr>
<tr>
<td>=&gt; vertebrate#1, craniate#1</td>
<td>=&gt; living thing#1, animate thing#1</td>
</tr>
<tr>
<td>=&gt; chordate#1</td>
<td>=&gt; whole#2, unit#6</td>
</tr>
<tr>
<td>=&gt; animal#1, animate being#1, beast#1, brute#2, creature#1, fauna#2</td>
<td>=&gt; object#1, physical object#1</td>
</tr>
<tr>
<td></td>
<td>=&gt; physical entity#1</td>
</tr>
<tr>
<td></td>
<td>=&gt; entity#1</td>
</tr>
</tbody>
</table>
wnb

- Hyponyms (going down the hierarchy):

```
3 of 8 senses of cat

Sense 1
cat#1, true cat#1
  => domestic cat#1, house cat#1, Felis domesticus#1, Felis catus#1
  => wildcat#3

Sense 2
guy#1, cat#2, hombre#1, bozo#2
  => sod#4

Sense 7
big cat#1, cat#7
  => leopard#2, Panthera pardus#1
  => snow leopard#1, ounce#3, Panthera uncia#1
  => jaguar#1, panther#1, Panthera onca#1, Felis onca#1
  => lion#1, king of beasts#1, Panthera leo#1
  => tiger#2, Panthera tigris#1
  => liger#1
  => tigon#1, tiglon#1
  => cheetah#1, chetah#1, Acinonyx jubatus#1
  => saber-toothed tiger#1, sabertooth#1
```
wnb

- Holonyms (part-of hierarchy):

  1 of 8 senses of cat
  Sense 7
  big cat#1, cat#7
  MEMBER OF: Felidae#1, family Felidae#1
  MEMBER OF: Carnivora#1, order Carnivora#1
  MEMBER OF: Eutheria#1, subclass Eutheria#1
  MEMBER OF: Mammalia#1, class Mammalia#1
  MEMBER OF: Vertebrata#1, subphylum Vertebrata#1, Craniata#1, subphylum Craniata#1
  MEMBER OF: Chordata#1, phylum Chordata#1
  MEMBER OF: Animalia#1, kingdom Animalia#1, animal kingdom#1
• Meronyms (has parts, plus inherited):

```
Sense 1
cat#1, true cat#1
  => feline#1, felid#1
    HAS PART: paw#1
    HAS PART: pad#7
  => carnivore#1
    => placental#1, placental mammal#1, eutherian#1, eutherian mammal#1
    => mammal#1, mammalian#1
      HAS PART: coat#3, pelage#1
      HAS PART: hair#4, pilus#1
    => vertebrate#1, craniate#1
      HAS PART: belly#5
      HAS PART: tail#1
        HAS PART: dock#6
      HAS PART: caudal appendage#1
    HAS PART: rib#2, costa#2
    HAS PART: costal cartilage#1
    HAS PART: thorax#2, chest#1, pectus#1
    HAS PART: sternum#1, breastbone#1
      HAS PART: gladiolus#2, corpus sternum#1
    HAS PART: manubrium#1
    HAS PART: xiphoid process#1
```
• Meronyms (has parts, plus inherited):

5 senses of car

Sense 1
car#1, auto#1, automobile#1, machine#6, motorcar#1
  HAS PART: accelerator#1, accelerator pedal#1, gas pedal#1, gas#5, throttle#2, gun#6
  HAS PART: air bag#1
  HAS PART: auto accessory#1
  HAS PART: automobile engine#1
    HAS PART: exhaust#2, exhaust system#1
      HAS PART: exhaust manifold#1
      HAS PART: exhaust pipe#1
      HAS PART: exhaust valve#1
      HAS PART: silencer#2, muffler#1
      HAS PART: tailpipe#1
    HAS PART: automobile horn#1, car horn#1, motor horn#1, horn#11, hooter#2
    HAS PART: horn button#1
    HAS PART: buffer#6, fender#4
    HAS PART: bumper#2
      HAS PART: bumper guard#1
    HAS PART: car door#1
      HAS PART: armrest#1
      HAS PART: doorlock#1
      HAS PART: hinge#1, flexible joint#1
      HAS PART: pintle#1
wnb

• Derivationally related (different part-of-speech):

2 of 8 senses of cat

Sense 3
cat#3
   RELATED TO->(adj) catty#1#1
       => bitchy#1, catty#1, cattish#1

Sense 5
cat-o'-nine-tails#1, cat#5
   RELATED TO->(verb) cat#1#1
       => cat#1
wnb

• Domain:

Sense 6
Caterpillar#2, cat#6
   USAGE->(noun) slang#2, cant#3, jargon#1, lingo#1, argot#1, patois#1, vernacular#1
   USAGE->(noun) trademark#2

Sense 1
car#1, auto#1, automobile#1, machine#6, motorcar#1
   TOPIC TERM->(adj) hopped-up#1
   TOPIC TERM->(noun) rental#2, renting#1
   TOPIC TERM->(noun) alternator#1
   TOPIC TERM->(noun) backseat#2
   TOPIC TERM->(noun) road map#2
   TOPIC TERM->(noun) showroom#1, salesroom#1, saleroom#1
   TOPIC TERM->(noun) spark lever#1
   TOPIC TERM->(noun) tunnel#1
   TOPIC TERM->(noun) passenger#1, rider#4
   TOPIC TERM->(noun) grip#4, traction#1, adhesive friction#1
   TOPIC TERM->(noun) chattel#1, personal chattel#1, movable#1
   TOPIC TERM->(verb) prang#1
The verb destroy has 4 senses (first 3 from tagged texts)

1. (42) **destroy**#1, destruct#2 -- (do away with, cause the destruction or undoing of; "The fire destroyed the house")
2. (28) **destroy**#2, ruin#1 -- (destroy completely; damage irreparably; "You have ruined my car by pouring sugar in the tank!"; "The tears ruined her make-up")
3. (2) **demolish**#3, **destroy**#3 -- (defeat soundly; "The home team demolished the visitors")
4. **destroy**#4, put down#6 -- (put (an animal) to death; "The customs agents destroyed the dog that was found to be rabid"; "the sick cat had to be put down")
wnb

• Coordinate terms (up 1, down 1)

Sense 1
destroy#1, destruct#2 -- (do away with, cause the destruction or undoing of; house)
  -> unmake#1, undo#2 -- (deprive of certain characteristics)
    => destroy#1, destruct#2 -- (do away with, cause the destruction or undo the house)

Sense 3
demolish#3, destroy#3 -- (defeat soundly; "The home team demolished the vi
  -> get the better of#1, overcome#1, defeat#1 -- (win a victory over; "You n
difficulties"; "defeat your enemies"; "He overcame his shyness"; "He ov
anger got the better of her and she blew up")
  => demolish#3, destroy#3 -- (defeat soundly; "The home team demolishe
  => beat#1, beat out#1, crush#3, shell#6, trounce#2, vanquish#1 -- (come
race, or conflict; "Agassi beat Becker in the tennis championship"; "We
"Harvard defeated Yale in the last football game")
  => wallop#2 -- (defeat soundly and utterly; "We'll wallop them!")
  => down#3 -- (bring down or defeat (an opponent))
  => overrun#4 -- (seize the position of and defeat; "the Crusaders overran
  => lurch#5, skunk#1 -- (defeat by a lurch)
  => rout#1, rout out#3, expel#3 -- (cause to flee; "rout out the fighters fro
  => upset#6 -- (defeat suddenly and unexpectedly; "The foreign team ups
• Troponyms (ways to):

- **destroy**
  - destroy#1, destruct#2 -- (do away with, cause the destruction or undoing of; "The fire destroyed the house")
  - => kill#15 -- (destroy a vitally essential quality of or in; "Eating artichokes kills the taste of all other foods")
  - => extinguish#4, eliminate#1, get rid of#2, do away with#1 -- (terminate, end, or take out; "Let's eliminate the course on Akkadian hieroglyphics"; "Socialism extinguished these archaic customs"; "eliminate my debts")
  - => cancel out#1, wipe out#6 -- (wipe out the effect of something; "The new tax effectively cancels out my raise"; "The `A' will cancel out the `C' on your record")
  - => decouple#4 -- (reduce or eliminate the coupling of (one circuit or part to another))
  - => decouple#3 -- (eliminate airborne shock waves from (an explosive))
  - => obliterate#4 -- (do away with completely, without leaving a trace)
  - => knock out#1 -- (eliminate; "knock out a target")
  - => drown#2 -- (get rid of as if by submerging; "She drowned her trouble in alcohol")
  - => cut out#1 -- (delete or remove; "Cut out the extra text"; "cut out the newspaper article")
    - => excise#3 -- (remove by cutting; "The surgeon excised the tumor")
  - => cut#33, prune#2, rationalize#2, rationalise#5 -- (weed out unwanted or unnecessary things; "We had to lose weight, so we cut the sugar from our diet")
  - => snuff out#1, extinguish#1 -- (put an end to; kill; "The Nazis snuffed out the life of many Jewish children")
  - => stamp#6 -- (destroy or extinguish as if by stamping with the foot; "Stamp fascism into submission"; "stamp out tyranny")
  - => smother#5, put out#3 -- (deprive of the oxygen necessary for combustion; "smother fires")

- **exclude**
  - exclude#1, except#2, leave out#1, leave off#2, omit#1, take out#14 -- (prevent from being included or considered or accepted; "The bad results were excluded from the report"; "Leave off
wnb

• Derivationally related to (parts of speech):

  Sense 1
  do away with, cause the destruction or undoing of; "The fire destroyed the house"

  RELATED TO->(adj) destructible#1
  => destructible#1 -- (easily destroyed; "destructible glassware")

  RELATED TO->(adj) destructive#1
  => destructive#1 -- (causing destruction or much damage; "a policy that is destructive to the economy"; "destructive criticism")

  RELATED TO->(noun) destruction#1
  => destruction#1, devastation#5 -- (the termination of something by causing so much damage to it that it cannot be repaired or no longer exists)

  RELATED TO->(noun) destroyer#2
  => destroyer#2, ruiner#1, undoer#3, waster#2, uprooter#1 -- (a person who destroys or ruins or lays waste to; "a destroyer of the environment"; "jealousy was his undoer"; "uprooters of gravestones")

  RELATED TO->(noun) destroyer#1
  => destroyer#1, guided missile destroyer#1 -- (a small fast lightly armored but heavily armed warship)
• Adjective

The adj happy has 4 senses (first 2 from tagged texts)

1. (37) **happy**#1 -- (enjoying or showing or marked by joy or pleasure; "a happy smile"; "spent many happy days on the beach"; "a happy marriage")
2. (2) felicitous#2, **happy**#2 -- (marked by good fortune; "a felicitous life"; "a happy outcome")
3. glad#2, **happy**#3 -- (eagerly disposed to act or to be of service; "glad to help")
4. **happy**#4, well-chosen#1 -- (well expressed and to the point; "a happy turn of phrase"; "a few well-chosen words")
Antonyms:

Sense 1
happy#1 (vs. unhappy#1) -- (enjoying or showing or marked by joy or pleasure; "a happy smile"; "spent many happy days on the beach"; "a happy marriage")

unhappy#1 (vs. happy#1) -- (experiencing or marked by or causing sadness or sorrow or discontent; "unhappy over her departure"; "unhappy with her raise"; "after the argument they lapsed into an unhappy silence"; "had an unhappy time at school"; "the unhappy (or sad) news"; "he looks so sad")
   => lovesick#1 -- (languishing because of love; "strong men behaving like lovesick boys")
   => miserable#1, suffering#2, wretched#3 -- (very unhappy; full of misery; "he felt depressed and miserable"; "a message of hope for suffering humanity"; "wretched prisoners huddled in stinking cages")

Sense 2
felicitous#2, happy#2 -- (marked by good fortune; "a felicitous life"; "a happy outcome")

INDIRECT (VIA fortunate#1) -> unfortunate#1 -- (not favored by fortune; marked or accompanied by or resulting in ill fortune; "an unfortunate turn of events"; "an unfortunate decision"; "unfortunate investments"; "an unfortunate night for all concerned")
• Value of:

happy#1 (vs. unhappy#1) -- (enjoying or showing or marked by joy or pleasure; "a happy smile"; "spent many happy days on the beach"; "a happy marriage")

  => happiness#1, felicity#2 -- (state of well-being characterized by emotions ranging from contentment to intense joy)

  => happiness#2 -- (emotions experienced when in a state of well-being)

• Related to:

Sense 1

happy#1 (vs. unhappy#1) -- (enjoying or showing or marked by joy or pleasure; "a happy smile"; "spent many happy days on the beach"; "a happy marriage")

  RELATED TO->(noun) happiness#1#1

  => happiness#1, felicity#2 -- (state of well-being characterized by emotions ranging from contentment to intense joy)

  RELATED TO->(noun) happiness#2#2

  => happiness#2 -- (emotions experienced when in a state of well-being)

Sense 2

felicitous#2, happy#2 -- (marked by good fortune; "a felicitous life"; "a happy outcome")

  RELATED TO->(noun) happiness#1#1

  => happiness#1, felicity#2 -- (state of well-being characterized by emotions ranging from contentment to intense joy)
• Pertains to:

<table>
<thead>
<tr>
<th>3 senses of axial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sense 1</td>
</tr>
<tr>
<td>axial#1</td>
</tr>
<tr>
<td>Pertains to noun axis#6 (Sense 6)</td>
</tr>
<tr>
<td>=&gt; axis#6, axis of rotation#1</td>
</tr>
<tr>
<td>=&gt; mechanism#5</td>
</tr>
</tbody>
</table>

| Sense 2 |
| axile#1, axial#2 |
| Pertains to noun axis#2 (Sense 2) |
| => axis#2 |
| => stalk#2, stem#2 |

Search Word: abaxial

The adj abaxial has 1 sense (no senses from tagged texts)

1. **abaxial**#1, dorsal#2 -- (facing away from the axis of an organ or organism; "the abaxial surface of a leaf is the underside or side facing away from the stem")

in old WordNet version (1.7.1)

abaxial *see also* axial

no connection here... as we’ll see
Programmed Search

• using Perl Module Query Data
perldoc WordNet::QueryData

• Start up
  1. use WordNet::QueryData;
  2. my $wn = WordNet::QueryData->new( noload => 1);
  3. noload => 0 (caches indexes in computer memory, slower startup, faster lookup time)
perldoc WordNet::QueryData

• querySense, queryWord
  – 1st argument
    • (1) \textit{word} returns list of \textit{word#pos}
    • \textbf{Example:} \$wn-->querySense("run")
    • \textbf{Answer:} run#n, run#v

• (2) \textit{word#pos} returns list of \textit{word#pos#sense}
  • \textbf{Example:} \$wn-->querySense("run#v")
  • \textbf{Answer:} run#v#1, run#v#2, run#v#3, run#v#4, run#v#5, run#v#6, run#v#7, run#v#8, run#v#9, run#v#10, run#v#11, run#v#12, run#v#13, run#v#14, run#v#15, run#v#16, run#v#17, run#v#18, run#v#19, run#v#20, run#v#21, run#v#22, run#v#23, run#v#24, run#v#25, run#v#26, run#v#27, run#v#28, run#v#29, run#v#30, run#v#31, run#v#32, run#v#33, run#v#34, run#v#35, run#v#36, run#v#37, run#v#38, run#v#39, run#v#40, run#v#41

• (3) \textit{word#pos#sense} + 2nd argument: a relation
perldoc WordNet::QueryData

• querySense, queryWord
  – 1st argument
    • (3) word#pos#sense + 2nd argument: a relation returns list of word#pos#sense
    • Example: $wn->querySense("cat#n#1", "hypo")
    • Answer: domestic_cat#n#1, wildcat#n#3
perldoc WordNet::QueryData

• To find other members of the synsets:

```perl
cat
```

perl cat.perl
domestic_cat#n#1 house_cat#n#1 Felis_domesticus#n#1 Felis_catus#n#1 wildcat#n#3

Sense 1
cat, true cat -- (feline mammal usually having thick soft fur and no ability to roar: domestic cats; wildcats)
  => domestic cat, house cat, Felis domesticus, Felis catus -- (any domesticated member of the genus Felis)
  => wildcat -- (any small or medium-sized cat resembling the domestic cat and living in the wild)
perldoc WordNet::QueryData

• Possible relations for queryWord:
  1. “also” - also see
  2. “ants” – antonyms
  3. “deri” - derived forms (nouns and verbs only)
  4. “part” - participle of verb (adjectives only)
  5. “pert” - pertainym (pertains to noun) (adjectives only)
  6. “vgrp” - verb group (verbs only)
perldoc WordNet::QueryData

• Possible relations for querySense:
  – also - also see
  – glos - word definition
  – syns - synset words
  – hype – hypernyms
  – inst - instance of
  – hypes - hypernyms and "instance of”
  – hypo – hyponyms
  – hasi - has instance
  – hypos - hyponyms and "has instance”
  – mmem - member meronyms
  – msub - substance meronyms
  – mprt - part meronyms
  – mero - all meronyms
  – hmem - member holonyms
  – hsub - substance holonyms
  – hpert - part holonyms
  – holo - all holonyms
  – attr - attributes (?)
  – sim - similar to (adjectives only)
  – enta - entailment (verbs only)
  – caus - cause (verbs only)
  – domn - domain – all
  – dmnc - domain – category
  – dmnu - domain – usage
  – dmnr - domain – region
  – domt - member of domain - all (nouns only)
  – dmtc - member of domain - category (nouns only)
  – dmtu - member of domain - usage (nouns only)
  – dmtr - member of domain - region (nouns only)
perldoc WordNet::QueryData

- validForms: 1 argument `word` or `word#pos`
  - It returns a list of all alternate forms (alternate spellings, conjugations, plural/singular forms, etc.).

```perl
1 use WordNet::QueryData;
2 my $wn = WordNet::QueryData->new( noload => 1);
3 my @l = $wn->validForms("went#v");
4 print "@l\n";
```

perl validforms.perl
go#v

**Note:** in documentation “lay down#v” returns lay_down#v, lie_down#v

“bank” returns bank#n bank#v
perldoc WordNet::QueryData

- level: one argument word#pos#sense

```
1 use WordNet::QueryData;
2 my $wn = WordNet::QueryData->new( noload => 1);
3 print $wn->level("convertible#n#1"), " \n";
4 print $wn->level("vehicle#n#1"), " \n";
5 print $wn->level("container#n#1"), " \n";
6 print $wn->level("physical_object#n#1"), " \n";
```

- 3 senses of convertible
  - Sense 1: convertible
    - => car, auto, automobile, machine, motorcar
      - => motor vehicle, automotive vehicle
        - => self-propelled vehicle
          - => wheeled vehicle
  - Sense 2: vehicle
    - => conveyance, transport
      - => instrumentality, instrumentation
        - => artifact, artefact
          - => whole, unit
            - => object, physical object
              - => physical entity
                - => entity
  - Sense 3: container
    - => instrumentality, instrumentation
      - => artifact, artefact
        - => whole, unit
          - => object, physical object
            - => physical entity
              - => entity
• **Exercise:**
  – find the relationship between *minibike* and *convertible*
Programmed search

• We can write a program to find the minimum distance between word senses: bfs.perl (on website)

```perl
1 # Example of breadth-first search of WordNet relations in Perl
2 # Sandiway Fong, University of Arizona
3 #
4 # usage: bfs word#pos#sense word#pos#sense
5 # synopsis: finds the semantic distance in links between the two word senses
6 if ($#ARGV != 1) {
7   print "usage: perl bfs.perl word#pos#sense word#pos#sense\n";
8   exit
9 }
10 my $start = $ARGV[0]; # e.g. minibike#n#1
11 my $end = $ARGV[1];  # e.g. convertible#n#1
12```
Programmed search

```
13 # USER-SETTABLE PARAMETERS
14 my $max = 20000;             # max number of nodes to be explored
15 my @relations = ("hype", "hypo", "mero", "holo"); # semantic relations
16
17 use WordNet::QueryData;
18 my $wn = WordNet::QueryData->new( noload => 0);
19 my @queue = ($start, 'mark');
20 my %seen = {};               # mark is used to count the depth of breadth-first search
21 my $found = 0;
22 my $n = 0;
23 my $distance = 0;           # hash %seen holds already visited nodes so we don't loop

semantic distance = depth of breadth-first search when $end is matched
```
Programmed search

```perl
24 sub found {
25    my $node = shift;
26    foreach $syn ($wn->querySense($node, "syns")) {
27        if ($end eq $syn) {
28            return 1
29        }
30    }
31    return 0
32 }
33 }
34
```

Example

**Synset:** minibike#n#1 motorbike#n#1
Programmed search

```plaintext
35 if (found($start)) {
36   $found = 1;
37 } else {
38   FOUND: while ($n < $max) {
39     my $node = pop @queue;
40     if ($node eq 'mark') {
41       $distance++;
42       unshift @queue, 'mark';
43       $node = pop @queue;
44     }
45     $seen{ $node } = 1;
46   }
```
Programmed search

foreach $rel (@relations) {
  foreach $newnode ($wn->querySense($node, $rel)) {
    $n++;
    if (found($newnode)) {
      $found = 1;
      last FOUND;
    } else {
      if (!$seen{$newnode}) {
        unshift @queue, $newnode;
      }
    }
  }
}

node is related to newnode via rel

increment node count

if $end is found, jump out of nested loop up to scope FOUND

if we haven’t seen newnode before, queue it for later expansion
Programmed search

only two ways to get to the end of the program,
either we found our target node $end
or $n is no longer < $max

63 if ($found) {
64    print "Found at distance $distance ($n nodes explored)\n";  
65 } else {
66    print "Not found (distance $distance and $n nodes explored)\n";  
67 }
Programmed search

• **Exercise revisited:**
  – find the relationship between *minibike* and *convertible*

  `perl bfs.perl minibike#n#1 convertible#n#1`
  – Found at distance 4 (74 nodes explored)

  `perl bfs.perl convertible#n#1 car#n#1`
  – Found at distance 1 (1 nodes explored)

  Note:
  `perl bfs.perl convertible#n#1 minibike#n#1`
  – Found at distance 4 (255 nodes explored)

Semantic relatedness
can we compare concept distances?
Programmed search

• A more informative search
  – we can modify the program to make each node keep a path, i.e. history, of the relations and nodes
  – website: bfs2.perl (*quick and dirty implementation*)
  – example:
    – perl bfs2.perl *minibike\#n\#1 convertible\#n\#1*
    – Found at distance 4 (74 nodes explored)
    – convertible\#n\#1 *hypo car\#n\#1 hypo motor_vehicle\#n\#1 hype motorcycle\#n\#1 hype minibike\#n\#1*
Programmed search

• Code changes:
  (uses Perl references to implement each node becoming a list itself)
  – my @queue = ([$start], ['mark']);

```perl
41 42 43 44 45 46 47 48
FOUND: while ($n < $max) {
  $node_ref = pop @queue;
  if (@{$node_ref}[0] eq 'mark') {
    $distance++;
    unshift @queue, ['mark'];
    $node_ref = pop @queue;
  }
  $seen[@{$node_ref}[0]] = 1;
}
```

- square brackets builds a reference to a list containing $start, another for ‘mark’
- picks out first element of the list
Programmed search

• Code changes (main loop):

```java
50   foreach $rel (@relations) {
51       foreach $newnode ($wn->querySense(@[$node_ref][0], $rel)) {
52           $n++;
53           if (found($newnode)) {
54               $found = 1;
55               unshift @[$node_ref], ($newnode, $rel);
56               last FOUND;
57           } else {
58               if (!$seen{$newnode}) {
59                   my @new = @[$node_ref];
60                   unshift @new, ($newnode, $rel);
61                   unshift @queue, @new;
62               }
63           }
64       }
```
Programmed search

• Code changes (print out at the end):

```perl
69 if ($found) {
70     print "Found at distance $distance ($n nodes explored)\n";
71     print "@{$node_ref}\n";
72 } else {
73     print "Not found (distance $distance and $n nodes explored)\n";
74 }
```

because we’ve been adding relations and new nodes @{$node_ref} is a list that contains the entire history of how we got there
Programmed search

• Other examples:
  perl bfs2.perl minibike#n#1 roof#n#1
  – Found at distance 5 (509 nodes explored)
  – roof#n#1 hype sunroof#n#1 mero car#n#1 hype motor_vehicle#n#1 hype motorcycle#n#1 hype minibike#n#1

perl bfs2.perl convertible#n#1 roof#n#1
  – Found at distance 3 (222 nodes explored)
  – roof#n#1 hype sunroof#n#1 mero car#n#1 hype convertible#n#1
Programmed search

• Not everything seems related:
  (even when search limit is upped to 80000)
  – perl bfs.perl chair#n#1 table#n#1
  – Not found (distance 8 and 80008 nodes explored)
  – perl bfs.perl table#n#1 chair#n#1
  – Not found (distance 9 and 80215 nodes explored)

perl bfs.perl table#n#1 chair#n#1
Found at distance 9 (140548 nodes explored)
perl bfs2.perl table#n#1 chair#n#1
Found at distance 9 (140548 nodes explored)
chair#n#1 hypo seat#n#3 holo upholstery#n#1 hypo covering#n#2 hypo artifact#n#1 hype decoration#n#1 hype flower_arrangement#n#1 hype arrangement#n#2 hype array#n#1 hype table#n#1

does the long chain still have meaning?
Programmed search

• So far ..
  – explored used querySense with relations hypo/hype/mero/holo

• More complete search
  – add more relations and queryWord

15 # USER-SETTABLE PARAMETERS
16 my $max = 100000;              # max number of nodes to be explored
17 my @relations = ("hype", "hypo", "mero", "holo", "enta", "caus");  # for querySense
18 my @relations2 = ("ants", "also", "deri", "pert");  # for queryWord

– Website: bfs3.perl
Programmed search

• Example: *axial*/*abaxial* revisited...
  – *surely should be related through noun axis*...

But

```perl
perl bfs3.perl abaxial#a#1 axial#a#3 500000
Max set to: 500000
Not found (distance 3 and 500029 nodes explored)
```

```perl
perl bfs3.perl axial#a#1 abaxial#a#1 500000
Max set to: 500000
Not found (distance 9 and 500002 nodes explored)
```
Programmed search

• Example:
  – John mended the torn dress
  – what can be deduced about the state of the world (situation) after the event of “mending”?
  – find the semantic relationship between mend and tear

bfs3.perl mend#v#1 tear#v#1
Found at distance 6 (58492 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hypo better#v#2 hype mend#v#1

perl bfs3.perl tear#v#1 mend#v#1
Found at distance 6 (33606 nodes explored)
mend#v#1 deri mender#n#1 hypo skilled_worker#n#1 hype cutter#n#3 deri cut#v#1 hypo separate#v#2 hype tear#v#1
Programmed search

• Find all minimal length solutions.
• Website: bfs4.perl
• Example:

```perl
perl bfs4.perl mend#v#1 tear#v#1
Found at distance 6 (58492 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hypo better#v#2 hype mend#v#1
Found at distance 6 (58552 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hypo better#v#2 hype mend#v#1
Found at distance 6 (67147 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype tinker#v#3 hypo mend#v#1
Found at distance 6 (67207 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype tinker#v#3 hypo mend#v#1
```
Programmed search

Found at distance 6 (67429 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype fill#v#9 hypo mend#v#1

Found at distance 6 (67489 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype fill#v#9 hypo mend#v#1

Found at distance 6 (71190 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype piece#v#5 hypo mend#v#1

Found at distance 6 (71250 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype piece#v#5 hypo mend#v#1

Found at distance 6 (74452 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype cobble#v#2 hypo mend#v#1
Programmed search

Found at distance 6 (74512 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype cobble#v#2 hypo mend#v#1

Found at distance 6 (75039 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype point#v#14 hypo mend#v#1

Found at distance 6 (75099 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype point#v#14 hypo mend#v#1

Found at distance 6 (75321 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype trouble-shoot#v#1 hypo mend#v#1

Found at distance 6 (75381 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype trouble-shoot#v#1 hypo mend#v#1
Programmed search

Found at distance 6 (75603 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype patch#v#3 hypo mend#v#1

Found at distance 6 (75663 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype patch#v#3 hypo mend#v#1

Found at distance 6 (76859 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype sole#v#1 hypo mend#v#1

Found at distance 6 (76919 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype sole#v#1 hypo mend#v#1

Found at distance 6 (78287 nodes explored)
tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants repair#v#1 hype vamp#v#4 hypo mend#v#1

Found at distance 6 (78347 nodes explored)
rupture#v#1 deri rupture#n#3 hypo breakage#n#3 deri break#v#4 ants repair#v#1 hype vamp#v#4 hypo mend#v#1
Programmed search

Found at distance 6 (78722 nodes explored)
tear\#v\#1 hypo separate\#v\#2 hype break\_up\#v\#10 also break\#v\#4 ants repair\#v\#1 hype heel\#v\#5 hypo mend\#v\#1

Found at distance 6 (78782 nodes explored)
rupture\#v\#1 deri rupture\#n\#3 hypo breakage\#n\#3 deri break\#v\#4 ants repair\#v\#1 hype heel\#v\#5 hypo mend\#v\#1

Found at distance 6 (79004 nodes explored)
tear\#v\#1 hypo separate\#v\#2 hype break\_up\#v\#10 also break\#v\#4 ants repair\#v\#1 hype darn\#v\#1 hypo mend\#v\#1

Found at distance 6 (79064 nodes explored)
rupture\#v\#1 deri rupture\#n\#3 hypo breakage\#n\#3 deri break\#v\#4 ants repair\#v\#1 hype darn\#v\#1 hypo mend\#v\#1

Found at distance 6 (86457 nodes explored)
tear\#v\#1 hypo separate\#v\#2 hype cut\#v\#1 deri cutter\#n\#3 hypo skilled\_worker\#n\#1 hype mender\#n\#1 deri mend\#v\#1

All minimal solutions found