## Programmed search

<table>
<thead>
<tr>
<th>Program</th>
<th>Semantic Relations</th>
<th>$\text{max}$</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>bfs.perl</td>
<td>4 querySense</td>
<td>20K</td>
<td>basic breadth-first search, displays (shortest)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>path length</td>
</tr>
<tr>
<td>bfs2.perl</td>
<td>4 querySense</td>
<td>20K</td>
<td>displays path as well as path length</td>
</tr>
<tr>
<td>bfs3.perl</td>
<td>6 querySense, 4</td>
<td>100K</td>
<td></td>
</tr>
<tr>
<td></td>
<td>queryWord</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bfs4.perl</td>
<td>6 querySense, 4</td>
<td>100K</td>
<td></td>
</tr>
<tr>
<td></td>
<td>queryWord</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CMDLINE PARAMETERS**

1. word#pos#sense
2. word#pos#sense

**OPTIONAL**

$max$ can be set using a 3rd cmdline parameter for **bfs3.perl** and **bfs4.perl**
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 }else{
10 my $start = $ARGV[0]; # e.g. minibike#n1
11 my $end = $ARGV[1]; # e.g. convertible#n1
12 ```
Programmed search

```perl
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 = {};
21 my $found = 0;
22 my $n = 0;
23 my $distance = 0;
```

mark is used to count the depth of breadth-first search

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 }
```

Example

Synset: minibike#n1 motorbike#n1
Programmed search

Recall Perl array: end (push/pop), beginning (shift/unshift)
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

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

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

Note:
```perl
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`

[https://wordnet.princeton.edu/wordnet/man/wngloss.7WN.html](https://wordnet.princeton.edu/wordnet/man/wngloss.7WN.html)

**hypernym**
The generic term used to designate a whole class of specific instances. 
Y is a hypernym of X if X is a (kind of) Y.

**hyponym**
The specific term used to designate a member of a class. 
X is a hyponym of Y if X is a (kind of) Y.
Programmed search

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

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

```plaintext
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 hypo
    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

Noun

- S: (n) **table#1** | **tabular array#1** (a set of data arranged in rows and columns) "see table 1"
- S: (n) **table#2** (a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs) "it was a sturdy table"
- S: (n) **table#3** (a piece of furniture with tableware for a meal laid out on it) "I reserved a table at my favorite restaurant"
- S: (n) **mesa#1**, **table#4** (flat tableland with steep edges) "the tribe was relatively safe on the mesa but they had to descend into the valley for water"
- S: (n) **table#5** (a company of people assembled at a table for a meal or game) "he entertained the whole table with his witty remarks"
- S: (n) **board#4**, **table#6** (food or meals in general) "she sets a fine table"; "room and board"

Verb

- S: (v) **postpone#1**, **prorogue#1**, **hold over#5**, **put over#2**, **table#1**, **shelve#1**, **set back#1**, **defer#1**, **remit#2**, **put off#1** (hold back to a later time) "let's postpone the exam"
- S: (v) **table#2**, **tabularize#1**, **tabularise#1**, **tabulate#1** (arrange or enter in tabular form)
Programmed search

• Note:
Programmed search

Noun

- **S: (n) chair#1** (a seat for one person, with a support for the back) "he put his coat over the back of the chair and sat down"
- **S: (n) professorship#1, chair#2** (the position of professor) "he was awarded an endowed chair in economics"
- **S: (n) president#4, chairman#1, chairwoman#1, chair#3, chairperson#1** (the officer who presides at the meetings of an organization) "address your remarks to the chairperson"
- **S: (n) electric chair#1, chair#4, death chair#1, hot seat#1** (an instrument of execution by electrocution; resembles an ordinary seat for one person) "the murderer was sentenced to die in the chair"
- **S: (n) chair#5** (a particular seat in an orchestra) "he is second chair violin"

Verb

- **S: (v) chair#1, chairman#1** (act or preside as chair, as of an academic department in a university) "She chaired the department for many years"
- **S: (v) moderate#1, chair#2, lead#14** (preside over) "John moderated the discussion"
Programmed search

• What do you think is the common link between *table* and *chair*?
Programmed search

perl bfs.perl chair#n#1 table#n#2
Found at distance 2 (78 nodes explored)
perl bfs2.perl chair#n#1 table#n#2
Found at distance 2 (78 nodes explored)

holonym
The name of the whole of which the meronym names a part.
Y is a holonym of X if X is a part of Y.

meronym
The name of a constituent part of, the substance of, or a member of something.
X is a meronym of Y if X is a part of Y.
Programmed search

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

• More complete search:
  • add more relations and **queryWord**

```perl
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:
  • *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*

<table>
<thead>
<tr>
<th>bfs3.perl mend#v#1 tear#v#1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found at distance 6 (58492 nodes explored)</td>
</tr>
<tr>
<td>tear#v#1 hypo separate#v#2 hype break_up#v#10 also break#v#4 ants</td>
</tr>
<tr>
<td>repair#v#1 hypo better#v#2 hype mend#v#1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>perl bfs3.perl tear#v#1 mend#v#1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found at distance 6 (33606 nodes explored)</td>
</tr>
<tr>
<td>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</td>
</tr>
</tbody>
</table>
Programmed search

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

```
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#v1 deri rupture#n3 hypo breakage#n3 deri
break#v4 ants repair#v1 hype cobble#v2 hype mend#v1

Found at distance 6 (75039 nodes explored)
tear#v1 hypo separate#v2 hype break_up#v10 also break#v4
ants repair#v1 hype point#v14 hypo mend#v1

Found at distance 6 (75099 nodes explored)
rupture#v1 deri rupture#n3 hypo breakage#n3 deri
break#v4 ants repair#v1 hype point#v14 hypo mend#v1

Found at distance 6 (75321 nodes explored)
tear#v1 hypo separate#v2 hype break_up#v10 also break#v4
ants repair#v1 hype trouble-shoot#v1 hypo mend#v1

Found at distance 6 (75381 nodes explored)
rupture#v1 deri rupture#n3 hypo breakage#n3 deri
break#v4 ants repair#v1 hype trouble-shoot#v1 hypo
mend#v1
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
Programmed search

perl bfs4.perl chair#n#1 table#n#2
Found at distance 2 (82 nodes explored)
table#n#2 holo leg#n#3 mero chair#n#1
All minimal solutions found

• Take out holo and mero from @relations

perl bfs4.perl table#n#2 chair#n#1
Found at distance 3 (139 nodes explored)
chair#n#1 hypo seat#n#3 hypo furniture#n#1 hype
table#n#2
All minimal solutions found

Noun

• S: (n) furniture#1, piece of furniture#1, article of furniture#1 (furnishings that make a room or other area ready for occupancy) "they had too much furniture for the small apartment", "there was only one piece of furniture in the room"
Homework 8

• Question 1:
  • Find the shortest distance links between *star* and *telescope*
  • (Make sure you have the right word sense)
  • How many are there?

• Question 2:
  • Draw the graph of semantic relations found

• Question 3:
  • Are any of the chains of semantic relations what you expect?

• Question 4:
  • Is the chain useful? Why or why not?

• Question 5:
  • What do you think the shortest connection linking star and telescope should look like?