Exploiting Graph Structure for Accelerating the Calculation of Shortest Paths in Wordnets

Holger Wunsch

University of Tübingen, Germany

Manchester, 22 August 2008
Motivation

- Frequent criticism: The relations in wordnets are too sparse
- Extend GermaNet (the German wordnet) with new relations (Lemnitzer, Wunsch, Gupta; 2008)
  - Extraction of verb-object and verb-subject pairs from the automatically parsed German newspaper corpus TüPP-D/Z (≈ 11.5 million sentences)
  - Ranking of the pairs according to mutual information and log-likelihood
  - Manual filtering (removal of nonsense pairs, support verb constructions, and words not present in GermaNet)
  - For each of the top 100 remaining pairs, a new relation is added to GermaNet (arg1 and arg2)

- Hypothesis: The better of both measures brings semantic fields closer together
Many approaches for determining *semantic similarity* between two concepts depend on the *shortest path* connecting them.
Motivation

- Many approaches for determining *semantic similarity* between two concepts depend on the *shortest path* connecting them.
- Calculating all shortest paths takes a lot of time.
  - $\Rightarrow$ 120 hours for GermaNet (approx. 53000 synsets), with Floyd-Warshall algorithm.
Motivation

Many approaches for determining *semantic similarity* between two concepts depend on the *shortest path* connecting them.

Calculating all shortest paths takes a lot of time:

⇒ 120 hours for GermaNet (approx. 53000 synsets), with Floyd-Warshall algorithm.

No problem for one-time offline calculation.
Motivation

- Many approaches for determining *semantic similarity* between two concepts depend on the *shortest path* connecting them.
- Calculating all shortest paths takes a lot of time. ⇒ 120 hours for GermaNet (approx. 53000 synsets), with Floyd-Warshall algorithm.
- No problem for one-time offline calculation.

But: How about repeatedly (semi-)automatically extending and evaluating the wordnet — with help of semantic similarity?
Many ("on-line") recalculations of shortest paths are a huge problem.
For GermaNet: 120 hours \( \times n \Rightarrow \text{infeasible} \)

**How to bring down processing time?**
Motivation

- Many ("on-line") recalculations of shortest paths are a huge problem.
- For GermaNet: 120 hours $\times n \Rightarrow$ infeasible

- How to bring down processing time?

- Use **Structure Adapted Shortest Path Search**
Wordnets and Graphs

<table>
<thead>
<tr>
<th>Wordnets</th>
<th>Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>synset</td>
<td>node</td>
</tr>
<tr>
<td>(directed) relation</td>
<td>(directed) edge</td>
</tr>
</tbody>
</table>

- A **synset** is a set of words that are synonymous.
- Two types of relations in wordnets:
  - **directed relations**
    - specific terms vs. more general terms (hyponymy – hyperonymy)
  - **undirected relations**
    - opposites (antonymy)
In general graphs, there are *multiple* paths connecting two nodes.

A general algorithm for finding a shortest path must consider all possible alternatives.

### Algorithms for finding all shortest paths

- Dijkstra’s algorithm \((n^3)\)
- Floyd-Warshall algorithm \((n^3)\)
  - Matrix-based (dynamic programming) approach
  - If there exists a shortest path between \(x\) and \(z\), and one between \(z\) and \(y\), then the shortest path between \(x\) and \(y\) is \(x - z - y\).
Are Wordnets Graphs?

They are for sure, but...

- Wordnets are (still) sparse
- Relatively few nodes in a dense central graph
- Numerous and large tree structures (biological and medical taxonomies, ...) on the fringe
Are Wordnets Graphs?

They are for sure, but...

- Wordnets are (still) sparse
- Relatively few nodes in a dense central graph
- Numerous and large tree structures (biological and medical taxonomies, ...) on the fringe

⇒ Wordnets are stars
Two-step Approach to Calculating Shortest Paths

- First pre-classify nodes
- Then use specialized algorithms for calculating the shortest path between nodes depending on their type
- Within trees: the path connecting two nodes is unique
- Within the graph part: use general path search algorithm
Node Classification

- Inner nodes
- Root nodes
- Tree nodes
- Leaf nodes
Node Classification

- Inner nodes
- Root nodes
- Tree nodes
- Leaf nodes
Node Classification

- Inner nodes
- Root nodes
- Tree nodes
- Leaf nodes
Node Classification

- Inner nodes
- Root nodes
- Tree nodes
- Leaf nodes
Node Classification

- Inner nodes
- Root nodes
- Tree nodes
- Leaf nodes
Node Classification

- Inner nodes
- Root nodes
- Tree nodes
- Leaf nodes
Path Calculation Proper

Path splitting

- From the start node...
- ...through the first tree...
- ...through the core graph...
- ...through the second tree...
- ...to the target node
Path splitting

- From the start node...
- ...through the first tree...
- ...through the core graph...
- ...through the second tree...
- ...to the target node
Path Calculation Proper

Path splitting

- From the start node...
- ...through the first tree...
- ...through the core graph...
- ...through the second tree...
- ...to the target node
Path Calculation Proper

Path splitting

- From the start node...
- ...through the first tree...
- ...through the core graph...
- ...through the second tree...
- ...to the target node
Path Calculation Proper

Path splitting

- From the start node...
- ...through the first tree...
- ...through the core graph...
- ...through the second tree...
- ...to the target node
**Structure Adapted Shortest Path Search**

\[
l_{xy} = l_{xr_x} + l_{rx_i_x} + l_{i_xi_y} + l_{i_yr_y} + l_{r_yy}
\]

with \(l_{rx_i_x} = l_{i_yr_y} = 1\)
\[ l_{xy} = l_{xr_x} + 1 + l_{ix_iy} + 1 \]
Structure Adapted Shortest Path Search

\[ l_{xy} = l_{xr_x} + 1 + l_{ix_iy} + 1 \]
## Results

<table>
<thead>
<tr>
<th></th>
<th>Wordnet</th>
<th>GermaNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synsets</td>
<td>117659</td>
<td>53312</td>
</tr>
<tr>
<td>Inner nodes</td>
<td>4250</td>
<td>8728</td>
</tr>
<tr>
<td>Root nodes</td>
<td>7174</td>
<td>4641</td>
</tr>
<tr>
<td>Tree nodes</td>
<td>56532</td>
<td>18949</td>
</tr>
<tr>
<td>Leaf nodes</td>
<td>49704</td>
<td>20683</td>
</tr>
<tr>
<td>Classification time</td>
<td>≈ 1 sec</td>
<td>1.2 sec</td>
</tr>
<tr>
<td>plain Floyd-Warshall</td>
<td>&gt; 35 days</td>
<td>120 hrs</td>
</tr>
<tr>
<td>Structure-adapted</td>
<td>9 min</td>
<td>40 min</td>
</tr>
<tr>
<td>shortest path search</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Holger Wunsch (University of Tübingen)
Discussion / Future Work

- Exploitation of wordnet-specific structure substantially reduces processing time
- Reduced memory overhead: Less housekeeping effort due to smaller graphs
- Replace greedy path search algorithm with heuristic ones
Thank You

Web:  http://www.sfs.uni-tuebingen.de/~wunsch
E-Mail:  wunsch@sfs.uni-tuebingen.de