IDENTIFYING WORD RELATIONS IN SOFTWARE: A COMPARATIVE STUDY OF SEMANTIC SIMILARITY TOOLS

Giriprasad Sridhara, Emily Hill, Lori Pollock and K. Vijay-Shanker

Natural Language Program Analysis Laboratory
Computer and Information Sciences
University of Delaware
U.S.A
Motivation

- Maintenance ➔ Hard to find what you’re looking for
- Scenario: Developer maintaining code written by another programmer, in an auction bidding software
  - Searches code for “remove auction”
  - Unsuccessful search

Vocabulary Mismatch

```
/** Delete an auction from the Auctions list that it's in.
 *
 * @param ae - The auction to delete.
 */

public void deleteAuction(AuctionEntry ae) {
```
Solving Vocabulary Mismatch

- Humans: Refine the query by adding related words
  - Error prone and time consuming
- Some IR techniques can automatically expand query

- Approaches:
  - Digital thesaurus with semantic similarity
  - Latent Semantic Indexing and related approaches

What words are similar to “remove”?

Remove

Delete/Withdraw/Eliminate

Giriprasad Sridhara
Synonyms are not enough for searching

Query: “money transaction” Not successful
Query: “bank transaction” successful

But <money, bank> not synonymous

Other Semantic Similarity Types:

- **Hypernyms and Hyponyms**
  - Words with general/specific meaning
  - Car → hyponym of Vehicle

- **Topically related**
  - Words belonging to the same topic
  - Bank, Check, Money, Deposit

All these types can be identified by current semantic similarity techniques.
Focus of this Paper

*How well do the thesaurus-based semantic similarity techniques work for software?*

**Contributions:**

- Quantitative and qualitative study of existing state of the art semantic similarity techniques applied to software
- Analysis of the limitations of these techniques when applied to software
  - Discovered opportunities for customization for software
Many **Potential** uses of Word Relations

- Program Exploration
- Refactoring
- Quality Assessment
- Code Recommendation
- Software Version Differencing Tools
- Automatic Generation of Code based on Keywords
- Traceability
In Practice: Who uses Word Relations in Software?

FindConcept (Shepherd et al. 2006)

- Concern location tool
- Uses synonyms to suggest additional query terms
- Uses a digital thesaurus
- Improved number of relevant methods retrieved

iComment (Tan et al. 2007)

- Bug detection based on comments
- Uses topically related words to expand query
- Mines related words from comments
- Achieved higher accuracy in extracting comments related to a query

Documentation to Source links with LSI (Marcus and Maletic 2003)

- Infers word relations from corpus
- Higher accuracy in retrieved links and more relevant links obtained
- Other tools based on LSI E.g. Concern Location, Code Clone detection
Targeted Open Question:

How well do the thesaurus-based semantic similarity techniques work for software?

- 6 Existing Semantic Similarity Techniques
- Quantitative Evaluation
- Qualitative Evaluation
- Summary and Future Work
Overview: Semantic Similarity Techniques

- Remove
- Delete

Word 1

Word 2

Semantic Similarity Technique

Similarity Score 0.71

WordNet: Most Commonly Used English Word Relation Database
Provides relation between word sense pairs

- **Sense**: Particular meaning of a word (e.g., financial bank, river bank, blood bank)
- **Gloss**: Definition of a sense

Synonymous word senses are organized into a set

- **Sense Hierarchy**

<table>
<thead>
<tr>
<th>Sense</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Program, broadcast}</td>
<td>A radio or TV show</td>
</tr>
<tr>
<td>{Program, Computer Program}</td>
<td>A sequence of instructions that a computer can interpret and execute</td>
</tr>
</tbody>
</table>

- **entity**
  - **object**
    - **organism**
      - **person**
        - **female**
          - **woman**
      - **parasite**
        - **parasitic plant**
          - **mistletoe**
### Semantic Similarity Techniques

<table>
<thead>
<tr>
<th>Category</th>
<th>Techniques</th>
<th>Similarity Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path-based</td>
<td>LCH, WUP</td>
<td>shorter path → words more similar</td>
</tr>
<tr>
<td>Information</td>
<td>RES, LIN and JCN</td>
<td>closer common ancestor → words more similar</td>
</tr>
<tr>
<td>Content</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gloss</td>
<td>LESK</td>
<td>more overlap in glosses → words more similar</td>
</tr>
</tbody>
</table>

![Diagram showing the relationship between person, female, girl, woman, man, and male.](image)
Our work: Comparing Semantic Similarity Techniques

- Collected word pairs from open source software
- Manually identified *related* word pairs
  - How? Mapped comment words to program identifier words
- Resulting Set is the **Gold Set** of 60 related word pairs
- **Gold Set** pairs → Input to 6 semantic similarity techniques

Example Gold Set Pairs:

<table>
<thead>
<tr>
<th>Comment Word</th>
<th>Code Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Begin</td>
</tr>
<tr>
<td>Display</td>
<td>Show</td>
</tr>
<tr>
<td>Write</td>
<td>Save</td>
</tr>
<tr>
<td>Output</td>
<td>Print</td>
</tr>
<tr>
<td>Locate</td>
<td>Find</td>
</tr>
</tbody>
</table>
Quantitative Evaluation: What is the appropriate measure?

Problem: Different techniques assign different scores to word pairs

So, what is the appropriate measure?

Number of Returned Results at x % recall:
To identify x % of the Gold Set (related word pairs), number of unrelated word pairs returned by the semantic similarity technique.

Where do we get unrelated word pairs for input?

Formed a set of 41,000 unrelated word pairs:
All possible comment word to code identifier mappings
Quantitative Evaluation Results

- Even at low thresholds (10%), 10 times the desired results (6 Gold Set Pairs) are returned.
- At 25% threshold, WUP and JCN outperform others.
- At 50% threshold, WUP is the winner.

But **100 times the number of desired results are returned!**

Implication: Not currently practical for software.

Giriprasad Sridhara
Digging deeper: Qualitative Evaluation

- Open question: If they work well on general English text, what went wrong in software?
- Hypothesis: General English Pairs scored better than software specific pairs, i.e., If software specific pairs scored higher, number of unrelated pairs would decrease
- Manually identified 5 software specific and 5 general English semantically similar pairs from the Gold Set and examined scores

<table>
<thead>
<tr>
<th>General English</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make-Create</td>
<td>Write-Save</td>
</tr>
<tr>
<td>Start-Begin</td>
<td>Assign-Set</td>
</tr>
<tr>
<td>Determine-Check</td>
<td>Find-Search</td>
</tr>
<tr>
<td>Locate-Find</td>
<td>Notify-Fire</td>
</tr>
<tr>
<td>Display-Show</td>
<td>Output-Print</td>
</tr>
</tbody>
</table>
Qualitative Evaluation Results

- General English pairs scored well
- Software specific pairs except <write, save> scored lesser than general English pairs
- Unrelated word pairs scored higher than software pairs

<table>
<thead>
<tr>
<th>General English</th>
<th>Score</th>
<th>Software</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make-Create</td>
<td>1</td>
<td>Write-Save</td>
<td>1</td>
</tr>
<tr>
<td>Start-Begin</td>
<td>1</td>
<td>Assign-Set</td>
<td>0.66</td>
</tr>
<tr>
<td>Determine-Check</td>
<td>1</td>
<td>Find-Search</td>
<td>0.51</td>
</tr>
<tr>
<td>Locate-Find</td>
<td>0.88</td>
<td>Notify-Fire</td>
<td>0</td>
</tr>
<tr>
<td>Display-Show</td>
<td>1</td>
<td>Output-Print</td>
<td>0.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unrelated</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>See-Have</td>
<td>0.87</td>
</tr>
<tr>
<td>Double-Print</td>
<td>0.72</td>
</tr>
<tr>
<td>Point-Throw</td>
<td>0.71</td>
</tr>
<tr>
<td>Execute-Close</td>
<td>0.68</td>
</tr>
<tr>
<td>Single-throw</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Why is <write, save> scored high by all techniques?  
Because they are listed as synonyms in WordNet
Summary and Future Work

- Existing semantic similarity techniques work well with general English pairs found in software
- Do not work well with software specific word pairs -- return many unrelated word pairs
  - Detrimental to software maintenance tools
- No need to develop new similarity techniques per se
- Customize similarity techniques for software
  - Augment Wordnet with software specific pairs
  - Mine pairs from software using statistical NLP techniques

Thanks!
Summary and Future Work

- Existing semantic similarity techniques work well with general English pairs found in software
- Do not work well with software specific word pairs -- return many unrelated word pairs
  - Detrimental to software maintenance tools
- No need to develop new similarity techniques per se
- Customize similarity techniques for software
  - Augment Wordnet with software specific pairs
  - Mine pairs from software using statistical NLP techniques

Thanks!
Summary and Future Work

- Existing semantic similarity techniques work well with general English pairs found in software
- Do not work well with software specific word pairs -- return many unrelated word pairs
  - Detrimental to software maintenance tools
- No need to develop new similarity techniques per se
- Customize similarity techniques for software
  - Customize Wordnet with software specific pairs
  - Mine pairs from software using statistical NLP techniques
Semantic Similarity Techniques

Path-based (LCH and WUP)
- shorter path $\rightarrow$ words more similar

Information Content (RES, LIN, JCN)
- closer common ancestor $\rightarrow$ words more similar

Gloss
- more overlap in glosses $\rightarrow$ words more similar
- LESK

Some studies show JCN and LESK performed better than others on English text.
Quantitative Evaluation:
What is the appropriate measure?

Different techniques assign different scores to Gold Set Pairs

LESK
- 440 unrelated word pairs
- <remove, delete>: 280
- <add, append>: 79
- <find, search>: 61
- 25%

LIN
- <add, append>: 1
- double, print 0.72
- <remove, delete>: 0.71
- execute, close 0.68
- 936 unrelated word pairs
- 63%
- <find, search>: 0.51

Giriprasad Sridhara
Summary and Future Work

- Existing semantic similarity techniques for English: May not be suitable for direct use in software
- Semantic similarity techniques need to be customized for software
- Two ways of customizing
  - Augment Wordnet with software specific word relations
  - Improve the word frequencies used by the information content based techniques
  - The above is our future work.
Software Tools using Related Words

**FindConcept (Shepherd et al. 2006)**

- Uses synonyms to suggest additional query terms
  - Uses WordNet
  - Improved recall of relevant methods retrieved

**iComment (Tan et al. 2007)**

- Bug detection based on comments
- Uses topically related words to expand query
- Mines related words from comments
- Achieved higher precision in obtaining comments related to a query

**Documentation to Source links with LSI (Marcus and Maletic 2003)**

- Infers word relations from corpus
- Higher precision and recall
- Other tools based on LSI E.g. Concern Location, Code Clone detection
Related Work

- Automatically mapping comment to code
  - Lawrie et al. (2006)
    - Assess quality by cosine similarity between comments and code
  - Fluri et al. (2007)
    - How comments and code evolve with time?

- Avoiding explicit semantic similarity by Latent Semantic Indexing
  - Marcus et al. (2003, 2004)
    - Uses co-occurrences of words in documents to discover hidden relations among words
      - Gives semantically similar words
    - But, the discovered word relations need not necessarily be semantically similar (superset)
    - Also, discovered relations are dependent on the corpus in which the co-occurrences are present.
### Qualitative Evaluation

- Identified 5 software specific and 5 general English semantically similar pairs.
- English pairs scored higher than software pairs.
- Scores from similarity technique LIN.
- Note: `<write, save>` scored high by all techniques (WordNet considers them synonyms).

<table>
<thead>
<tr>
<th>Word Pair</th>
<th>Score</th>
<th># Returned Results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>English</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make-Create</td>
<td>1</td>
<td>68</td>
</tr>
<tr>
<td>Start-BEGIN</td>
<td>1</td>
<td>68</td>
</tr>
<tr>
<td>Determine-Check</td>
<td>1</td>
<td>68</td>
</tr>
<tr>
<td>Locate-Find</td>
<td>0.88</td>
<td>149</td>
</tr>
<tr>
<td>Display-Show</td>
<td>1</td>
<td>68</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write-Save</td>
<td>1</td>
<td>68</td>
</tr>
<tr>
<td>Assign-Set</td>
<td>0.66</td>
<td>636</td>
</tr>
<tr>
<td>Find-Search</td>
<td>0.51</td>
<td>1376</td>
</tr>
<tr>
<td>Notify-Fire</td>
<td>0</td>
<td>18680</td>
</tr>
<tr>
<td>Output-Print</td>
<td>0.62</td>
<td>804</td>
</tr>
</tbody>
</table>
Conclusion (I)

- Existing semantic similarity techniques for English
  - May not be suitable for direct use in software
- At low recall levels (upto 25%),
  - JCN and WUP appear to perform better than others
- At high recall levels, none of the techniques do well.
  - No need to develop new similarity detection techniques per se
  - Only customize for software
Conclusion (II)

❖ Two ways of customizing
  ▪ Augment Wordnet with software specific word relations
  ▪ Improve the word frequencies used by the information content based techniques
  ▪ The above is our future work.

❖ Semantic similarity detection in software
  ▪ JCN with customized wordnet and word frequencies from programs may be the best technique.
Quantitative Evaluation

- What is the appropriate measure?
- Different techniques assign different scores to Gold Set Pairs

Number of Returned Results at x % recall: To identify x % of the GoldSet (related pairs), how many unrelated pairs must be returned by the

- LESK
  - 10%: Map, Find 338
  - 50%: Remove, Delete 280
  - Find, Search 61

- LIN
  - 25%: Remove, Delete 0.71
  - 63%: Execute, Close 0.68
  - Find, Search 0.51
Path-based Semantic Similarity Detection

- Path-based technique (WUP, LCH)
  - Shorter the path between words $w_1$ and $w_2$ in the wordnet is-a hierarchy
    - more similar they are
  - Note: Can add an animation for path based in the wordnet hierarchy.

- Information Content based (LIN, RES, JCN)
  - Overcomes drawbacks of path based
    - Edge lengths are treated uniformly by path based,
    - BUT,
    - Edge length of 1 between root and child, different from,
    - Edge length of 1 between leaf and its parent.
  - Use frequency of occurrence from an English corpus + wordnet hierarchy
    - English corpus can be newspaper articles from WallStreet Journal and so on
Six English Semantic Similarity Techniques (continued)

❖ Gloss Based (LESK)

- Based on the intuition,
  - Related word senses are described using the same words
- Both ‘car’ and ‘automobile’ have ‘vehicle’ in their glosses.
- Note: Overlay the glosses and show that both glosses have ‘vehicle’
- Existing studies on English text have shown
  - JCN and LESK to be the best.
Motivation

- Scenario: Developer maintaining code written by another programmer:
- Wants to modify/understand feature “remove an auction”
  - Searches code for “remove auction”
    - Unsuccessful

Vocabulary Mismatch

How is “remove auction” implemented?
To identify $x\%$ of the Gold Set (related pairs), how many unrelated pairs were returned by the semantic similarity technique.

Quantitative Evaluation:

What is the appropriate measure?

Different techniques assign different scores to Gold Set Pairs.

Number of Returned Results at $x\%$ recall:

- To identify $x\%$ of the Gold Set (related pairs), how many unrelated pairs were returned by the semantic similarity technique.

- Number of unrelated word pairs returned:
  - LESK: 440
  - LIN: 936

- Scores for different techniques:
  - LESK: find, search 0.51
  - LIN: find, search 0.72

Giriprasad Sridhara
### Types of Semantic Similarity

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synonyms</strong></td>
<td>- Words that have the same meaning</td>
</tr>
<tr>
<td></td>
<td>- Display and Show</td>
</tr>
<tr>
<td><strong>Hypernyms/Hyponyms</strong></td>
<td>- Words with a more general / specific meaning</td>
</tr>
<tr>
<td></td>
<td>- Vehicle (\rightarrow) hypernym of Car</td>
</tr>
<tr>
<td></td>
<td>- Car (\rightarrow) hyponym of Vehicle</td>
</tr>
<tr>
<td><strong>Meronyms/Holonyms</strong></td>
<td>- Words denoting part / whole relationship</td>
</tr>
<tr>
<td></td>
<td>- Wheel (\rightarrow) meronym of Car</td>
</tr>
<tr>
<td></td>
<td>- Car (\rightarrow) holonym of Wheel</td>
</tr>
<tr>
<td><strong>Topically related</strong></td>
<td>- Words belonging to the same topic</td>
</tr>
<tr>
<td></td>
<td>- Bank, Check, Money, Deposit</td>
</tr>
<tr>
<td><strong>Antonyms</strong></td>
<td>- Words having opposite meaning</td>
</tr>
<tr>
<td></td>
<td>- Hot and Cold</td>
</tr>
</tbody>
</table>
**WordNet**

- Provides relation between word sense pairs
- Sense: Particular meaning of a word, e.g., river bank, blood bank
- Gloss: Definition of a sense
- Synonymous word senses are organized into a set
  - E.g.,

**Sense Hierarchy**

<table>
<thead>
<tr>
<th>Sense</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>A radio or TV show</td>
</tr>
<tr>
<td>Computer Program</td>
<td>A sequence of instructions that a computer can interpret and execute</td>
</tr>
</tbody>
</table>

Giriprasad Sridhara