Text Extraction, Similarity and WordNet

Steve Vincent
March 6, 2006
Agenda

• Extraction of term-document matrix
• WordNet and Similarity Measures
• WordNet::Similarity
• Putting it all together
The Document-term Matrix is the matrix with rows are indexed by documents and columns are indexed by terms.

\[
D = \begin{bmatrix}
tf(t_1, d_1) & \ldots & tf(t_n, d_1) \\
\vdots & \ddots & \vdots \\
tf(t_1, d_m) & \ldots & tf(t_n, d_m)
\end{bmatrix}
\]

\(D^T\) (or the transpose of \(D\)) is the Term-document matrix.
Extracting Term-Document Matrix

• There are several tools to extract a matrix which will show the number of terms in each document.

• The most user-friendly is the Text to Matrix Generator (TMG) written by D. Zeimpekis and E. Gallopolous from the University of Patras, Greece.

• The application is a MATLAB GUI which will be shown in the next slides.
TMG Application

A MATLAB Toolbox for generating term-document matrices from text collections

Dorit Bamberger and E. Gallopoulos
High Performance Systems Laboratory
Computer Engineering & Informatics Department
University of Patras
26504 Patras, Greece

Summary

MATLAB® toolbox that constructs new and updates existing term-document matrices from documents, in the form of MATLAB sparse matrices, such as removal of common words, such as articles and conjunctions, removal of very short or very long infrequent terms, and the application of stemming. TMG applies common filtering techniques (removal of common words, frequent or infrequent, removal of words that are too short or too long) to reduce the size of the term dictionary. TMG accepts an input CIF text. In most cases it also processes with maximum accuracy HTML and many PostScript and PDF files. TMG allows an option to load, save, and export data sets as well as a wide range of other features. TMG can be used as a preprocessor for a variety of information retrieval tasks based on linear algebra in vector space models (LSM), clustering, and other techniques such as Principal Component Analysis and Principal Components. The TMG also supports incremental updating of existing matrices by efficient rigid operations.

Information

Currently, TMG is currently in beta

Note: The beta version of TMG is available at the following link: http://www.cs.ucr.edu/∼gallop/
TMG GUI Screen Shot
Semantic Relatedness

• Some pairs of words are closer in meaning than others
  – E.g. *car* – *tire* are strongly related
    *car* – *tree* are not strongly related
• *Relatedness* between words can consist of
  – Synonymy [e.g. *car* – *automobile*]
  – Is-a/has-a relationships [e.g. *car* – *tire*]
  – Co-occurrence [e.g. *car* – *insurance*]
Ontologies

• Tools of information representation on a subject
• Hierarchical categorization of terms from general to most specific terms
  – object → artifact → construction → stadium
• Domain Ontologies representing knowledge of a domain
  – e.g., MeSH medical ontology
• General Ontologies representing common sense knowledge about the world
  – e.g., WordNet
WordNet: A Database of Lexical Relations

• WordNet:
  – The most well-developed and widely used lexical DB for English
  – Handcrafting from scratch, rather than mining information from existing dictionaries and thesauri
  – Consisting three separate DBs:
    • One each for nouns and verbs, and
    • A third for adjectives and adverbs
• Located at: http://wordnet.princeton.edu/
• A vocabulary and a thesaurus offering a hierarchical categorization of natural language terms
• Synsets represent terms or concepts
  – stadium, bowl, arena, sports stadium – (a large structure for open-air sports or entertainments)
WordNet

- A hierarchically organized lexical database
- On-line thesaurus + aspects of a dictionary

<table>
<thead>
<tr>
<th>Category</th>
<th>Unique Forms</th>
<th># of Senses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>117,097</td>
<td>145,104</td>
</tr>
<tr>
<td>Verb</td>
<td>11,488</td>
<td>24,890</td>
</tr>
<tr>
<td>Adjective</td>
<td>22,141</td>
<td>31,302</td>
</tr>
<tr>
<td>Adverb</td>
<td>4,601</td>
<td>5,720</td>
</tr>
</tbody>
</table>

Current for WordNet 2.1
WordNet Hierarchies

• There are 10 noun relations, 6 verb relations, 5 adjective relations, and 2 adverb relations.
• The synsets are also organized into senses
• Senses: Different meanings of the same term
• The synsets are related to other synsets higher or lower in the hierarchy by different types of relationships e.g.
  – Hyponym/Hypernym (Is-A relationships)
    • Approximately 60% of total relationships
  – Meronym/Holonym (Part-Of relationships)
# WordNet Noun Relations

<table>
<thead>
<tr>
<th>Relation</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypernym</td>
<td>From concepts to superordinates</td>
<td>breakfast → meal</td>
</tr>
<tr>
<td>Hyponym</td>
<td>From concepts to subtypes</td>
<td>meal → lunch</td>
</tr>
<tr>
<td>Has-Member</td>
<td>From groups to their members</td>
<td>faculty → professor</td>
</tr>
<tr>
<td>Member-Of</td>
<td>From members to their groups</td>
<td>copilot → crew</td>
</tr>
<tr>
<td>Has-Part</td>
<td>From wholes to parts</td>
<td>table → leg</td>
</tr>
<tr>
<td>Part-Of</td>
<td>From parts to wholes</td>
<td>course → meal</td>
</tr>
<tr>
<td>Antonym</td>
<td>Opposites</td>
<td>leader → follower</td>
</tr>
</tbody>
</table>
Verb Relations in WordNet

<table>
<thead>
<tr>
<th>Relation</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyponym</td>
<td>From events to superordinate events</td>
<td>fly (\rightarrow) travel</td>
</tr>
<tr>
<td>Troponym</td>
<td>From events to their subtypes</td>
<td>walk (\rightarrow) stroll</td>
</tr>
<tr>
<td>Entails</td>
<td>From events to the events they entail</td>
<td>snore (\rightarrow) sleep</td>
</tr>
<tr>
<td>Antonym</td>
<td>Opposites</td>
<td>increase (\leftrightarrow) decrease</td>
</tr>
</tbody>
</table>

Adjective & Adverb Relations in WordNet

<table>
<thead>
<tr>
<th>Relation</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antonym</td>
<td>Opposite</td>
<td>heavy (\leftrightarrow) light</td>
</tr>
<tr>
<td>Adverb</td>
<td>Opposite</td>
<td>quickly (\leftrightarrow) slowly</td>
</tr>
</tbody>
</table>
Semantic Similarity Methods

- Map terms to an ontology and compute their relationship in that ontology
- Four main categories of methods:
  - **Edge counting**: path length between terms
  - **Information content**: as a function of their probability of occurrence in corpus
  - **Feature based**: similarity between their properties (e.g., definitions) or based on their relationships to other similar terms
  - **Hybrid**: combine the above ideas
Semantic Similarity Measures

• Edge Counting
  – Leacock-Chodorow
  – Wu & Palmer
  – Hirst & St-Onge
  – Liu
• Information Content
  – Resnik
  – Lord
  – Lin
  – Jiang-Conrath

• Feature Based
  – Adapted Lesk
  – Tversky
• Hybrid
  – Rodriquez
Edge Counting Methods

• Measures based only on taxonomic (IS-A) links between concepts.

• **Edge-counting metric** measures the distance between a pair of categories by the length of the shortest path connecting them in a hierarchy
  – Problem with edge counting is that the path lengths in an ontology are irregular across the hierarchies. In addition, some related terms are not in the same hierarchies.
  – For example: “tennis problem”
  – Another example: The distance between “plant” and “animal” is 2 in WordNet while the distance between “zebra” and “horse” is also 2.
Leacock & Chodorow (1998)

- $\text{Len}(c_1, c_2)$ is the shortest path between two synsets.
- Deals with nouns only
- Limit attention to IS-A links and scale the path length by the overall depth of the taxonomy.

$$\text{sim}_{LC}(c_1, c_2) = -\log \frac{\text{len}(c_1, c_2)}{2L}$$

where $L$ is the overall depth of the taxonomy
Wu & Palmer (1994)

- Based on IS-A links between two concepts to their most specific common superclass
- Similarity given by:

\[
sim_{Wu&Palmer}(c_1, c_2) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3}
\]

where \(N_1\) and \(N_2\) are the number of IS-A links from \(c_1\) to \(c_2\) to their most specific common superclass and \(N_3\) is the number of IS-A links form the most specific common superclass to the root of the ontology.
Hirst & St-Onge (1998)

• Assumptions that two concepts are semantically close if:
  – The length of the path between them is not too long
  – The path between them does not change direction too often

• Represented by

\[ \text{Sim}_{HS}(c_1,c_2) = C - \text{PathLength}(c_1,c_2) - k \times d \]

where \( d \) is the number of changes in direction and \( C \) and \( k \) are constants.

• No restriction to nouns
Liu (2003)

• Combines the shortest path length between two concepts \(c_1\) and \(c_2\), \(L\), and the depth in the taxonomy of the most specific common concept \(c\), \(H\), in a non-linear function.

\[
sim(c_1, c_2) = e^{-\alpha L} \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}}
\]

where \(\alpha \geq 0\) and \(\beta > 0\) are parameters scaling the contribution of shortest path length and depth respectively.
Information-theoretic approaches

- **Information content (IC):** nodes high in the hierarchy have a small IC
- The information shared by two nodes can be represented by their common ancestors (*least common subsumer*)
- The more information two terms share, the more similar they are
Resnik (1995)

• Deals with nouns only and is based on the is a hierarchy of WordNet

• Information Content of a concept indicates the specificity of the concept.

\[
IC(\text{concept}) = -\log(P(\text{concept}))
\]

• Probability of occurrence of concept in a corpus is calculated using its frequency in the corpus.

\[
P(\text{concept}) = \frac{\text{freq}(\text{concept})}{\text{freq}(\text{root})}
\]
Resnik (1995)

• If $c_1$ and $c_2$ are two classes and $c_0$ is the most specific class that subsumes both $c_1$ and $c_2$, then:

\[
sim_R(c_1, c_2) = \frac{2 \times \log p(c_0)}{\log p(c_1) + \log p(c_2)} \quad \text{or} \quad \log p(c_1) + \log p(c_2)
\]

\[
\sim_R(c_1, c_2) = -\log(lcs(c_1, c_2))
\]

where $lcs(c_1, c_2)$ is the lowest common subsumer or the closest concept that is the hypernym of both $c_1$ and $c_2$ of the two concepts.

• The Resnik measure has a lower bound of 0 with no upper bound.
Lord (2003)

• Compare two terms is by using a measure that simply uses the probability of the most specific shared parent.

\[
\text{Sim}(c_1, c_2) = 1 - P(\text{concept})
\]

• The probability-based similarity score takes values between 1 (for the very similar concepts) and 0.
Lin (1998)

- Used information content. Similar to Jaing-Conrath, but uses different formula.

\[ sim_L(c_1, c_2) = \frac{2 \times \log(p(lso(c_1, c_2)))}{\log(p(c_1)) + \log(p(c_2))} \]

- The Lin measure has a lower bound of zero and an upper bound of one.
- This measure can be seen as taking the information content of the intersection of the two concepts and then dividing it by their sum then multiplying by 2.
Jaing-Conrath (1997)

• Jiang and Conrath use the same concept as Resnik, but measure the semantic distance, which is the inverse of similarity.

$$\text{dist}_{jc}(c_1,c_2) = 2\log(p(lcs(c_1,c_2))) - \log(p(c_1)) - \log(p(c_2))$$

• The Jiang and Conrath measure uses the information content measure of Resnik and augments it with measure of the path length between concepts.

• It also assumes that all words are nouns and has a lower bound of zero with no upper limit.
Adapted Lesk (2002)

• Lesk’s (1986) idea: Related word senses are (often) defined using the same words. E.g:
  – bank(1): “a financial institution”
  – bank(2): “sloping land beside a body of water”
  – lake: “a body of water surrounded by land”

• Gloss overlaps = # content words common to two glosses \(\approx\) relatedness
  – Thus, relatedness (bank(2), lake) = 3
  – And, relatedness (bank(1), lake) = 0
Adapted Lesk (2002)

• In the original Lesk Algorithm, the similarity of two words is determined by the largest number of words that overlap in their glosses or definitions.
• In the Adapted Lesk Algorithm, a word vector $w$ is created corresponding to every content word in the WordNet glosses. Concatenating glosses of related concepts in WordNet can be used to augment this vector.
• The vector contains the co-occurrence counts of words co-occurring with $w$ in a large corpus. Adding all the word vectors for all the content words in its gloss creates the Gloss vector $g$ for a concept.
• Relatedness is determined by comparing the gloss vector using the Cosine similarity measure.
Tversky (1977)

- Model addresses problems of geometric models of similarity
- Represent stimuli with sets of discrete features
- Similarity is a flexible function of the number of common and distinctive features

\[
\text{Similarity}(X,Y) = a(\text{shared}) - b(\text{X but not Y}) - c(\text{Y but not X})
\]

\(a, b,\) and \(c\) are weighting parameters

- Different identification of distinguishing features by classifying them into functions, parts, and attributes:
  - Functions represent what is done to or with instances of a class
  - Parts are structural elements of a class
  - Attributes correspond to additional characteristics

\[
sim(c_1, c_2) = w_p \cdot S_p(c_1, c_2) + w_f \cdot S_f(c_1, c_2) + w_a \cdot S_a(c_1, c_2)
\]

where \( w_p, w_f \) and \( w_a \geq 0 \) and \( w_p + w_f + w_a = 1 \). For each type of distinguishing features, \( S_p, S_f, S_a \) a similarity function based on the Tversky feature-matching model is used.
Semantic Similarity in WordNet

Testing by Varelas et al*:
• The most popular methods are evaluated
• All methods applied on a set of 38 term pairs
• Their similarity values are correlated with scores obtained by humans

**Evaluation**

<table>
<thead>
<tr>
<th><strong>Method</strong></th>
<th><strong>Type</strong></th>
<th><strong>Correlation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Wu &amp; Palmer 1994</td>
<td>Edge Counting</td>
<td>0.74</td>
</tr>
<tr>
<td>Li 2003</td>
<td>Edge Counting</td>
<td>0.82</td>
</tr>
<tr>
<td>Leacock &amp; Chodorow 1998</td>
<td>Edge Counting</td>
<td>0.82</td>
</tr>
<tr>
<td>Resnik 1999</td>
<td>Info. Content</td>
<td>0.79</td>
</tr>
<tr>
<td>Lin 1998</td>
<td>Info. Content</td>
<td>0.82</td>
</tr>
<tr>
<td>Lord 2003</td>
<td>Info. Content</td>
<td>0.79</td>
</tr>
<tr>
<td>Jiang &amp; Conrath 1998</td>
<td>Info. Content</td>
<td>0.83</td>
</tr>
<tr>
<td>Tversky 1977</td>
<td>Feature Based</td>
<td>0.73</td>
</tr>
<tr>
<td>Adapted Lesk 2002</td>
<td>Feature Based</td>
<td>0.37*</td>
</tr>
<tr>
<td>Rodriguez 2003</td>
<td>Hybrid</td>
<td>0.71</td>
</tr>
</tbody>
</table>

*Not reported in Verelas et al paper, but included in other comparison studies. Provided on this list for completeness.*
Semantic Similarity Web

Intelligent Systems Laboratory

Semantic Similarity Measures In WordNet

First Term

Second Term

By default, all senses of each term are compared and the maximum similarity is returned. To specify a sense, use sense_number after the term (e.g. cat#1)

Methods Implemented

- Edge Counting Methods
  
  
  
  
  Leacock and Chodorow: Claudia Leacock and Martin Chodorow. Combining local context and WordNet similarity for word sense identification, in
WordNet::Similarity

This is a Perl module that implements a variety of semantic similarity measures based on information found in the lexical database WordNet. In particular, it supports the measures of Renart, Lin, Jiang-Coetz, and Patwardhan.

be upgraded to 1.02)

released 02/07/06) from CPAN or Sourceforge

Try the Web Interface (v0.15), needs to
Download the Current Version (v1.02,

Documentation

See the README and ChangeLog files, as well as our To
This diagram shows the major modules and functions of W.
Browsbe the current Psi version.

Version Dependencies of WordNet-Similarity

- WordNet-Similarity 1.02 supports WordNet 2.1, as
  - WordNet (version 2.1)
  - WordNet::QueryData (version 1.39 or better)
  - Text-Similarity (version 0.02)
- WordNet-Similarity 0.16 supports WordNet 2.0, as
  - WordNet (version 2.0)
  - WordNet::QueryData (version 1.38)
  - Text-Similarity (version 0.02)
- WordNet-Similarity supports WordNet 1.7 and 1.7
- We don't think WordNet-Similarity supports Word

Ted Pedersen - UC, Berkeley; Microsoft, Inc. - #1 thanks
WordNet::Similarity Web Interface

You may enter any two words in one of three formats:

1. word
2. word\_part\_of\_speech (where part\_of\_speech is one of s, v, a, or r)
3. word\_part\_of\_speech\_sense (where sense is a positive integer)

If words are entered in format 1 or 2, then the relatedness of all valid forms of the words will be computed (e.g., if 'dogs' is entered, then 'dog' will be used to compute relatedness).

More instructions.

- Use all senses
- Pick a sense by gloss
- Pick a sense by sense

Word 1:  
Word 2:  
Measure: Path length

Show version info

Created by Ingrid Mihalik and Ted Pedersen.
E-mail: mihalik\_1@\_col\_usa\_edu or pedersen\_1@\_col\_usa\_edu
WordNet::Similarity Information

- WordNet::Similarity is implemented with Perl’s object oriented features. It uses the WordNet::QueryData package to create an object representing WordNet.
- WordNet::Similarity can also be utilized via a command line interface provided by the utility program similarity.pl. This allows a user to run the measures interactively for specific pairs of concepts when given in word#pos#sense form.
  - For example, car#n#3 refers to the third WordNet noun sense of car.
- Regardless of how it is run, WordNet::Similarity supports detailed tracing that shows a variety of diagnostic information specific to each of the different kinds of measures.
Putting it all together

• When TMG is run for any good sized corpus, a term list of over 10,000 words is possible.
• To get semantic similarity of the 10,000 words, it will be necessary to run approximately 50,000,000 tests.
• There is a better way:
  – Select Nouns and/or Verbs since most of the similarity measures only use this part of speech.
  – Increase the size of your “Stop List” to remove common terms and remove more of the rare terms.