Exploiting Semantic Analysis of Documents for the Domain User

Evangelos Milios
Dalhousie University

Closing Conference:
Statistical and Computational Analytics for Big Data

Dalhousie University
June 12, 2015
Outline

• Visual text analytics point of view
• Interactive document clustering
  – Term-supervised
  – Concept-supervised
  – Dual-supervised
• Entity and concept based document representation
  – Sunflower: concept representation
  – Tulip: Entity recognition and disambiguation
• Google n-gram based relatedness
• Domain-specific semantic relatedness from Wikipedia structure
  – Case of biomedical text
Text Visualization

Visualization

Text corpus

Text Visualization

Text Mining
Interactive Text Visualization

Visualization

User

Text Mining

Text corpus

Interaction

Text Visualization
Visual text analytics

Text Mining

Text corpus

Visualization

Text Visualization

Interaction

User

Manipulation

Feedback
Hamid Nourashraf, Yeming Hu

INTERACTIVE DOCUMENT CLUSTERING
Motivation:
Organizing research literature

- Writing a proposal
- Writing a thesis
- Organizing a lab library
- Program chairing a conference
Problem Characteristics

• Often in practice, no metadata is available
• Automatically generated clusters are often unsatisfactory
• Each user has own point of view in organizing documents
  – Topics for document clusters
  – Number of clusters
• No ground truth is available in practice for evaluation
  – The user evaluates the quality of clusterings
Modes of User Supervision

• Document Supervision:
  – Labeling a set of training documents
  – Specifying *must-link* and *cannot-link* constraints

• Term Supervision:
  – Term Selection (Specifying a feature space)
  – Term Labeling (Specifying clusters’ subspaces)

• Dual Supervision:
  – Document Supervision + Term Supervision
Problem Definition

• Clustering a collection of text documents
• No metadata is available
  o No assumption is considered for documents
  o Each document is a set of terms (Bag of Words model)
• User-desired document clusters
  ➢ Number of document clusters
  ➢ Topic of document clusters
• Challenge
  ➢ Useful clusters with minimum user effort
Road Map

Evolutionary Clustering (FSDC) → Lexical Double Clustering (LDC) → Ensemble Clustering (ELSDC)

- Doc-supervised LDC
- Term-supervised LDC
- Dual-supervised LDC

User Interface → User Study
Base Algorithm: Lexical Double Clustering (LDC)

Unsupervised

Document-term Matrix

Term Clustering

Term cluster 1

Seed Extraction

Seed documents 1

....

Term cluster k

Seed Extraction

Seed documents k

Document Clustering

Bag of Words

Fuzzy c-means
Cosine Similarity

Feature Selection is performed on term clusters to keep only discriminative terms.

Euclidean Distance

Computational Complexity: $O(NMK)$
LDC vs. LDA model

LDA[10000]: LDA is run 10,000 iterations
LDA[RelTime]: LDA is run for max(runtime(LDC))

Each algorithm is run 50 times on eight text corpora
Quality is measured by Normalized Mutual Information
LDC vs. LDA model

LDA[10000]: LDA is run 10,000 iterations
LDA[RelTime]: LDA is run for max(runtime(LDC))

Each algorithm is run 50 times on eight text corpora
Quality is measured by Normalized Mutual Information
Issue of different vocabularies

• What if related documents use different terms?
  – Medical research paper
  – Clinical practice guideline
  – Patient information pamphlet

• Bag of words (BOW) model:
  – Each document is a set of terms
  – No semantic relation among terms
  – It is limited to term frequency
  – Related documents might not have common terms
Enriching BOW

• Enriching document representation using external knowledge resource
  – WordNet
  – Wikipedia.

• Coverage of WordNet is limited.
  – Terminology
  – Named entities

• Wikipedia
  – Concepts (entries), incl. named entities
  – Category structure
  – Multilingual versions
### Wikification

- Extracting related concepts from Wikipedia
- Concept: title of a Wikipedia article

<table>
<thead>
<tr>
<th>Document 1</th>
<th>Document 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bag of Terms</strong></td>
<td><strong>Bag of Concepts</strong></td>
</tr>
<tr>
<td>people social network family connections relations communication computer friends</td>
<td>packet wireless internet topology network hub switch communication computer connections people</td>
</tr>
<tr>
<td><strong>Bag of Concepts</strong></td>
<td><strong>Bag of Concepts</strong></td>
</tr>
<tr>
<td>Computer Social-network</td>
<td>Ethernet-hub Topology Wireless Wi-Fi Communication Computer-Network Internet Network-Packet</td>
</tr>
</tbody>
</table>

Ensemble Lexical-Semantic Document Clustering

Semantic seed documents share common concepts with Term Clusters.
LDC vs. ELSDC

- LDC is based on bag of terms.
- ELSDC is an ensemble algorithm based on bag of terms and concepts.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LDC</th>
<th>ELSDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>20newsgroups</td>
<td>0.6328 ± 0.0105</td>
<td>0.6695 ± 0.0223*</td>
</tr>
<tr>
<td>Reuters</td>
<td>0.5381 ± 0.0289</td>
<td>0.5861 ± 0.0222*</td>
</tr>
<tr>
<td>Classic4</td>
<td>0.8145 ± 0.0302</td>
<td>0.8370 ± 0.0184*</td>
</tr>
</tbody>
</table>

mean(NMI) ± standard deviation

*Significant difference according to paired-sample T-test with p≤ 0.05
User-Supervised Algorithms
LDC

- Unsupervised
- Random initialization
Doc-supervised LDC

- Document supervision in the form of document labeling
- Keyterm extraction is performed using $\chi^2$ statistic.
Term-supervised LDC

- Term supervision in the form of term labeling
- Keyterm extraction is performed using $\chi^2$ statistic.
Dual-supervised LDC

- Dual Supervision: Document labeling + Term Labeling
User Study

• 30 Computer Science students
• Each participant uploads her own papers in PDF format
• Individual Participation (21)
  o Each participant has her own collection
• Group Participation (9)
  o A document collection of 300 papers
• Analysis based on participants’ opinions and system log
Future Work

1. Term-supervised LDC in clustering Big data (scalability)
2. Term-supervised LDC in collaborative multi-user document clustering
3. Improved Visualization
   1. Wikipedia concepts
   2. Multi-word terms
4. Extending term-supervised LDC to hierarchical clustering and creating a mind map
TULIP: LIGHTWEIGHT ENTITY RECOGNITION AND DISAMBIGUATION USING WIKIPEDIA-BASED TOPIC CENTROIDS
Problem Definition

- The goal of Entity Recognition and Disambiguation (ERD)
  - Identify mentions of entities
  - Link the mentions to a relevant entry in an external knowledge base
  - The knowledge base is typically a large subset of Wikipedia articles

- Example:
  - The selling offsets decent earnings from **Cisco Systems** and **Home Depot**. Techs fall, led by **Microsoft** and **Intel**. **Michael Kors** rises. Gold and oil slip.
Recognition and Disambiguation


- **Recognition**
  - Is this a valid mention of an entity present in the knowledge base?

- **Disambiguation**
  - Which of the potential entities (senses) is correct?
Recognition and Disambiguation

• The selling offsets decent earnings from Cisco Systems and Home Depot. Techs fall, led by Microsoft and Intel. Michael Kors rises. Gold and oil slip.

Questions:

- **Recognition**: Is this a valid mention of an entity present in the knowledge base?
- **Disambiguation**: Which of the potential entities (senses) is correct?
  - **Default sense** – the entity with a largest number of wiki-links with the mention as the anchor text
    - Tulip focuses on default sense entities
    - Main goal is to recognize whether the default sense is consistent with the document
A key technical challenge

- Key issue with state-of-the-art systems: obvious false positive mistakes

- Visualize Prof. Smith's research interests:
  - Data Mining
  - Machine Learning
  - 50 cent

- Our goal: minimize the number of false positives
Tulip – system overview

- **Spotter**
  - Find all mentions of entities in the text (Solr Text Tagger)
  - Special handling for personal names

- **Recognizer**
  - Retrieve profiles of spotted entities (from Sunflower)
  - Generate a topic centroid representing the document
  - Select entities consistent with the document
Spotter

- Find all mentions of entities in the text (Solr Text Tagger)
- Special handling for personal names

Recognizer

- Retrieve profiles of spotted entities (from Sunflower)
- Generate a topic centroid representing the document
- Select entities consistent with the document
Solr Text Tagger

- Solr (Lucene) is a text search engine
  - Indexes textual documents
  - Retrieve documents for keyword-based queries

- Solr Text Tagger
  - Indexes entity surface forms stored in a lexicon
  - E.g., Baltimore Ravens, Ravens, Baltimore (…)
  - Uses full text documents as queries
  - Finds all entity mentions in the document
  - Retrieves the mentioned entities (candidate selection)
  - Implemented based on Solr's Finite State Transducers
  - By David Smiley and Rupert Westenthaler (thanks!)
Building the lexicon

- **Three sources of entity surface forms** (external datasets)
  - Entity names (from *Freebase*)
  - Wiki-links anchor text (from *Wikipedia*)
  - Web anchor text (from *Google’s Wikilinks corpus*)

- **Special handling of personal names**
  - “Jack” and “London” are not allowed as surface forms for Jack London
  - Instead they are indexed as “generic” personal names and will be matched only if Jack London is mentioned by his full name

- **Flagging suspicious surface forms** (e.g., “It” - Stephen King's novel)
  - **stop-word filter** marks all stop-words or phrases composed of stop-words (e.g., *This is*)
  - **Wiktionary filter** marks all common nouns, verbs, adjectives, etc. found in Wiktionary
  - **lower-case filter** marks all lower-case words or phrases
Spotter – example


- Default sense for all mentions (Freebase only)
Spotter – example


- Default sense for all mentions (Freebase only)
- Default sense for all mentions (Freebase + Wikipedia)
Spotter – example


- Default sense for all mentions (Freebase only)
- Default sense for all mentions (Freebase + Wikipedia)
- Suspicious mentions removed
Spotter – example


- Default sense for all mentions (Freebase only)
- Default sense for all mentions (Freebase + Wikipedia)
- Suspicious mentions removed
- How can we remove Michael Kors and bring back Home Depot?
  - Relatedness of entities to the document
Recognizer

- **Spotter**
  - Find all mentions of entities in the text (Solr Text Tagger)
  - Special handling for personal names

- **Recognizer**
  - Retrieve profiles of spotted entities (from Sunflower)
  - Generate a topic centroid representing the document
  - Select entities consistent with the document
Relatedness score

The selling offsets decent earnings from **Cisco Systems** and **Home Depot**. Techs fall, led by **Microsoft** and **Intel**. **Michael Kors** rises. Gold and oil slip.

How strongly or are related to the document?

- **Our solution**
  - Retrieve a profile of every entity mentioned in the text
  - Agglomerate the profiles in a centroid representing the document
  - Check which entities are coherent with the topics (relatedness score)
  - **How do we create the entity profiles?**
Sunflower for relatedness

- A **concept graph** based on unified **category graph** from 120 Wikipedia language versions
  - Each language version acts like a witness for the importance of stored relation
- Compact and accurate category profiles for all Wikipedia articles
  - Removal of unimportant categories
  - Inference of more general categories
Sunflower – from graph to term profile

- Sunflower graph is:
  - Directed
  - Weighted (importance score)
  - Sparse (only $k$ most important links per node)

- Category-based profile is a sparse, weighted term vector
  - All categories at distance $< d$
  - Term weights based on edge weights
    - E.g., $k = 3$, $d = 2$
  - Path weight is the product of edge weights

- Example:
  $w(\text{Intel} \rightarrow \text{Comp. of US} \rightarrow \text{Ec. of US}) = 0.42 \times 0.27 = 0.11$
  - Category weight is the sum of path weights
  - $w(\text{Ec. of US}) = 0.11 + 0.19 = 0.3$
## ERD (SIGIR 2014) Challenge results

<table>
<thead>
<tr>
<th>rank</th>
<th>team name</th>
<th>F1</th>
<th>prec./recall</th>
<th>latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MS_MLI</td>
<td>0.76</td>
<td>0.83/0.70</td>
<td>1.49</td>
</tr>
<tr>
<td>2</td>
<td>MLNS <em>(Tulip)</em></td>
<td>0.74</td>
<td>0.76/0.71</td>
<td>0.29</td>
</tr>
<tr>
<td>3</td>
<td>Seznam Research</td>
<td>0.72</td>
<td>0.79/0.66</td>
<td>2.33</td>
</tr>
<tr>
<td>4</td>
<td>NTUNLP</td>
<td>0.71</td>
<td>0.76/0.67</td>
<td>7.66</td>
</tr>
<tr>
<td>5</td>
<td>HITS</td>
<td>0.70</td>
<td>0.77/0.65</td>
<td>4.97</td>
</tr>
<tr>
<td>6</td>
<td>Neofonie</td>
<td>0.70</td>
<td>0.76/0.65</td>
<td>0.53</td>
</tr>
<tr>
<td>7</td>
<td>WebSAIL</td>
<td>0.69</td>
<td>0.72/0.65</td>
<td>0.70</td>
</tr>
<tr>
<td>8</td>
<td>Acube Lab</td>
<td>0.67</td>
<td>0.87/0.54</td>
<td>0.86</td>
</tr>
<tr>
<td>9</td>
<td>ExPoSe</td>
<td>0.63</td>
<td>0.74/0.55</td>
<td>0.71</td>
</tr>
<tr>
<td>10</td>
<td>UBC</td>
<td>0.63</td>
<td>0.74/0.55</td>
<td>37.29</td>
</tr>
</tbody>
</table>

- Tulip got second place in the long track
  - Category-based topic centroids – promising solution for relatedness
  - Top recall among all submitted systems (?!)
  - Lowest latency among all submitted systems
SEMANTIC RELATEDNESS THROUGH GOOGLE N-GRAMS
Google n-gram corpus: information

- Google Web 1T n-gram Corpus

<table>
<thead>
<tr>
<th>Number of</th>
<th>Number</th>
<th>Size on disk (in KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>1,024,908,267,229</td>
<td>N/A</td>
</tr>
<tr>
<td>Sentences</td>
<td>95,119,665,584</td>
<td>N/A</td>
</tr>
<tr>
<td>Unigrams</td>
<td>13,588,391</td>
<td>185,569</td>
</tr>
<tr>
<td>Bigrams</td>
<td>314,843,401</td>
<td>5,213,440</td>
</tr>
<tr>
<td>Trigrams</td>
<td>977,069,902</td>
<td>19,978,540</td>
</tr>
<tr>
<td>4-grams</td>
<td>1,313,818,354</td>
<td>32,040,884</td>
</tr>
<tr>
<td>5-grams</td>
<td>1,176,470,663</td>
<td>33,678,504</td>
</tr>
</tbody>
</table>
Google n-gram corpus: example

<table>
<thead>
<tr>
<th>n=</th>
<th>n-grams</th>
<th>Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>he was a</td>
<td>3,683,417</td>
</tr>
<tr>
<td></td>
<td>hehe was a</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>he was an</td>
<td>563,471</td>
</tr>
<tr>
<td></td>
<td>he was am</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>he was awesome</td>
<td>7,520</td>
</tr>
<tr>
<td></td>
<td>he was awsome</td>
<td>548</td>
</tr>
<tr>
<td>4</td>
<td>he was a vegetarian</td>
<td>1,357</td>
</tr>
<tr>
<td></td>
<td>he was a veritable</td>
<td>454</td>
</tr>
<tr>
<td></td>
<td>he was a very</td>
<td>65,325</td>
</tr>
<tr>
<td></td>
<td>he was a veteran</td>
<td>2,979</td>
</tr>
<tr>
<td>5</td>
<td>he was a very generous</td>
<td>276</td>
</tr>
<tr>
<td></td>
<td>he was a very genuine</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>he was a very gifted</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>he was a very good</td>
<td>7,447</td>
</tr>
</tbody>
</table>
Word (pair) similarity

• Based on Google tri-grams
• Consider all tri-grams that start and end with the given word pair
• Normalize their mean frequency using uni-gram frequency of the two words
Word (pair) similarity

- Word pair $w_a \ w_b$
- $n_1$ tri-grams starting with $w_a$ and ending with $w_b$
- Frequency of one such tri-gram
  \[ c(w_a w_i w_b) \]
- Sum the frequencies of all tri-grams
  \[ \sum_{i=1}^{n_1} c(w_a w_i w_b) \]
Word (pair) similarity

• Consider both orders of words, and average the frequency sums

\[
\mu(w_a, n_1, w_b, n_2) = \frac{1}{2} \left( \sum_{i=1}^{n_1} c(w_a w_i w_b) + \sum_{i=1}^{n_2} c(w_b w_i w_a) \right)
\]

• Normalize

\[
\text{Sim}(w_a, w_b) = \begin{cases} 
\frac{\log \frac{\mu(w_a, n_1, w_b, n_2) C^2}{c(w_a)c(w_b) \min(c(w_a), c(w_b))}}{-2 \log \frac{\min(c(w_a), c(w_b))}{C}} & \text{if } \frac{\mu(w_a, n_1, w_b, n_2) C^2}{c(w_a)c(w_b) \min(c(w_a), c(w_b))} > 1 \\
\frac{\log 1.01}{-2 \log \frac{\min(c(w_a), c(w_b))}{C}} & \text{if } \frac{\mu(w_a, n_1, w_b, n_2) C^2}{c(w_a)c(w_b) \min(c(w_a), c(w_b))} \leq 1 \\
0 & \text{if } \mu(w_a, n_1, w_b, n_2) = 0
\end{cases}
\]
Text similarity

• P = “An autograph is the signature of someone famous which is specially written for a fan to keep”

• R = “Your signature is your name, written in your own characteristic way, often at the end of a document to indicate that you wrote the document or that you agree with what it says”
Graphical representation
Proposed method (1)

• **Step 1**: After preprocessing,
  – $P = \{\text{autograph, signature, famous, specially, written, fan}\}$ and
  – $R = \{\text{signature, written, characteristic, end, document, wrote, document, agree}\}$,
  – where $m = 6$ and $n = 8$.

• **Step 2**: Only two tokens (i.e., signature and written) in $P$ exactly match with $R$, therefore,
  – We set $\delta$ to 2.
  – We remove “signature” and “written” from both $P$ and $R$. As $m - \delta \neq 0$, we proceed to next step.
Proposed method (2)

• Step 3: We construct a $4 \times 6$ similarity matrix, $M$, excluding common words

\[
M = \begin{pmatrix}
\text{characteristic} & \text{end} & \text{document} & \text{wrote} & \text{document} & \text{agree} \\
\text{autograph} & 0 & 0 & 0.259 & 0.282 & 0.259 & 0 \\
\text{famous} & 0.257 & 0.055 & 0.051 & 0.374 & 0.051 & 0.001 \\
\text{specially} & 0 & 0.168 & 0.258 & 0.137 & 0.258 & 0 \\
\text{fan} & 0 & 0.012 & 0 & 0.203 & 0 & 0.174
\end{pmatrix}
\]
Proposed method (3)

• Step 4:
  – compute mean and stdev of each row
  – Select elements that represent high similarities (>mean+stdev)

\[
A_i = \{ \alpha_{ij} : \alpha_{ij} \in \{\alpha_{i1}, \ldots, \alpha_{ij}, \ldots, \alpha_{i(n-\delta)}\}, \quad \alpha_{ij} > \\
\mu(\{\alpha_{i1}, \ldots, \alpha_{ij}, \ldots, \alpha_{i(n-\delta)}\}) + \sigma(\{\alpha_{i1}, \ldots, \alpha_{ij}, \ldots, \alpha_{i(n-\delta)}\}) \}
\]
Proposed method (4)

• Step 5:
  – Add the means of high similarity elements
  – Compute overall similarity $S(P, R)$

$$S(P, R) = \frac{(\delta + \sum_{i=1}^{m-\delta} \mu(A_i)) \times (m + n)}{2mn}$$
Evaluation
Work completed

• Efficient implementation (6 orders of magnitude faster than naïve implementation), with Andrew Rau-Chaplin
• Phrase similarity using 4-grams (Master’s thesis by M. Rakib)
• Summarization using the most “central” words/phrases in the self-similarity matrix of a document
SEMANTICRELATEDNESSFROMWIKIPEDIASTRUCTURE

A. Sajadi, V. Keselj, J. Janssen
How good is Wikipedia for domain-specific semantic relatedness?

• Domain:
  – Biomedical text

• Novel Graph-based similarity based on Wikipedia

• Comparison with
  – Ontology-based methods
  – Distributional methods
Why biomedical domain?

• Availability of high-quality ontologies (MeSH, SNOMED-CT, …)
• Rich literature on semantic relatedness
• Availability of reliable datasets with ground truth
Key idea

• Given two concepts
• Extract neighbourhood graph for each concept in Wikipedia graph
• Transform graph to a list using HITS algorithm
• Calculate Kendall’s tau distance between the two lists
Relatedness Calculation
Formulation

• HITS Ranking Algorithm
  – Input: A graph with adjacency matrix M
  – Output: Authority / Hub scores on vertices

• Kendall’s tau distance
  – Number of pairwise disagreements between two lists

• Given two concepts
  – Lists of neighbours sorted by HITS (hub or auth)
  – Lists compared by Kendall’s tau
## Results

### Table 1. Comparison with Ontology-based methods. $o_1$: set-umls; $o_2$: mesh-umls; $o_3$: umls

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$o_1$</td>
<td>$o_2$</td>
<td>$o_3$</td>
<td>$o_1$</td>
</tr>
<tr>
<td>LCH</td>
<td>.44</td>
<td>.42</td>
<td>.61</td>
<td>.03</td>
</tr>
<tr>
<td>IIC-LCH</td>
<td>.38</td>
<td>.43</td>
<td>.7</td>
<td>.3</td>
</tr>
<tr>
<td>PPR</td>
<td>.63</td>
<td>.31</td>
<td>.69</td>
<td>.17</td>
</tr>
<tr>
<td><strong>HITS-sim</strong></td>
<td>.71</td>
<td>* .52</td>
<td>* .58</td>
<td>* .51</td>
</tr>
</tbody>
</table>

### Table 2. Comparison with distributional methods. * Mayo Corpus of Clinical Notes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Resources</th>
<th>Pedersen</th>
<th>Mayo</th>
<th>UMN sim.</th>
<th>UMN rel.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector</td>
<td>Mayo Corpus*+UMLS</td>
<td>.76</td>
<td>†.02</td>
<td>†.02</td>
<td>†-.13</td>
</tr>
<tr>
<td>Tensor</td>
<td>OHSUMED+UMLS</td>
<td>.76</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word2Vec</td>
<td>OHSUMED</td>
<td>†.34</td>
<td>†.26</td>
<td>†.36</td>
<td>†.29</td>
</tr>
<tr>
<td></td>
<td>OHSUMED+UMLS</td>
<td>.80</td>
<td>.63</td>
<td>.39</td>
<td>.39</td>
</tr>
<tr>
<td><strong>HITS-sim</strong></td>
<td>Wikipedia</td>
<td>.71</td>
<td>.52</td>
<td>.58</td>
<td>.51</td>
</tr>
</tbody>
</table>

### Table 3. Comparison between Wikipedia based methods

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>.73</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPRel</td>
<td>.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WLM†</td>
<td>.86</td>
<td>.67</td>
<td>.63</td>
<td>.69</td>
<td>.67</td>
<td>.49</td>
<td>.58</td>
<td>.49</td>
</tr>
<tr>
<td>Co-Citation†</td>
<td>.86</td>
<td>.67</td>
<td>.62</td>
<td>.68</td>
<td>.66</td>
<td>.47</td>
<td>.57</td>
<td>.49</td>
</tr>
<tr>
<td>Coupling†</td>
<td>.90</td>
<td>* .65</td>
<td>.61</td>
<td>.66</td>
<td>.64</td>
<td>* .44</td>
<td>* .49</td>
<td>* .4</td>
</tr>
<tr>
<td>Amsler†</td>
<td>.86</td>
<td>.68</td>
<td>.58</td>
<td>.66</td>
<td>.64</td>
<td>* .45</td>
<td>* .53</td>
<td>* .43</td>
</tr>
<tr>
<td>SimRank†</td>
<td>.79</td>
<td>* .51</td>
<td>* .56</td>
<td>* .55</td>
<td>* .55</td>
<td>* .39</td>
<td>* .45</td>
<td>* .39</td>
</tr>
<tr>
<td>EHITS-sim†</td>
<td>.84</td>
<td>* .62</td>
<td>.6</td>
<td>.67</td>
<td>.64</td>
<td>* .46</td>
<td>* .54</td>
<td>* .45</td>
</tr>
<tr>
<td>HITS-sim</td>
<td>.88</td>
<td>.70</td>
<td>.67</td>
<td>.72</td>
<td>.71</td>
<td>.52</td>
<td>.58</td>
<td>.51</td>
</tr>
</tbody>
</table>
Conclusion

• Wikipedia based method
  – Comparable to the available specialized resources
  – Often significantly improves upon them

• New HITS based algorithm
  – Best correlation with human judgement
LESSONS LEARNED
Discussion

• Text mining for the end user requires rethinking of:
  – The algorithms (for interactivity)
  – The document representations (concept-centric)

• Exploit knowledge bases for semantics
  – Wikipedia and derivatives

• Computational aspects
  – Complex computation in the inner loop of text mining algorithms
  – Speed to support interactivity

• Evaluation needs ethnographic / longitudinal methods
Future work

• Use the techniques for
  – Interactive clustering
  – Visualization

• Applications to:
  – News recommendation (work with Halifax Herald)
  – Support of reuse of technical documentation in authoring (work with Innovatia)
  – Filtering resumes against job advertisements (work with Interviewrocket)
  – OCR error correction of old printed books (Mining Biodiversity project, with Biodiversity Heritage Library (USA) and NaCTeM (U. of Manchester)
Publications

  First Prize in the long document category.
  – Marek’s ERD page: https://web.cs.dal.ca/~lipczak/erd/


Publications


• Aminul Islam, Evangelos Milios, Vlado Keselj. (2012). _Comparing word Relatedness based on Google n-grams_. 24th Int. Conf. on Computational Linguistics (COLING), Mumbai, India, 2012-12-08 (8pp)


• Armin Sajadi, Evangelos E. Milios, Vlado Keselj,and Jeannette C.M. Janssen: _``Domain-Specific Semantic Relatedness from Wikipedia Structure: A Case Study in Biomedical Text'',_16th International Conference, Computational Linguistics and Intelligent Text Processing, CICLing 2015,Cairo, Egypt, April 14–20, 2015, Springer LNCS 9041, part I, pp. 347-360, Received the Verifiability, Reproducibility, and Working Description Award (1st Place).