

Putting Word Relatedness Metrics To Work

Timothy Weale

Department of Computer Science and Engineering
The Ohio State University
<http://www.cse.ohio-state.edu/~weale>

2009.04.24



Outline

- 1 Background Information
 - Research Question
 - Graph-Based Measures
- 2 Graph Extraction
 - Vertex Extraction
 - Edge Extraction
 - Link Text
- 3 TREC Evaluation
 - Lemur Toolkit
 - TREC Evaluation
 - Query Expansion
- 4 Future Work
 - Weighted Transitions



Motivation

Primarily a forcing function for progress in my research.

Feedback on portions of my dissertation work:

- *Data Set Challenges*: Why is working with Wikipedia hard?
- *Evaluation Techniques*: How am I implementing my IR evaluations?
- *Future Questions*: What does my next year look like?



Research Question In A Slide...

Develop a function to measure the relatedness of two words.

$$f(w_1, w_2) = x \quad (1)$$

w_1 and w_2 are the input words

x is the measure of relatedness between the two input words and

$$0 \leq x \leq 1$$



Research Variables

- 1 **Data Source** – the data set mined for relatedness. Sources differ in many aspects that may affect the resulting weights, including topic coverage, labeling methodology and internal organization.
- 2 **Function Form** – the internals of the relatedness function. Different functions weigh data features differently and are applicable only for certain types of data sources.
- 3 **Target Task** – the application domain for the relatedness metric. Word relatedness has been used in fields such as Natural Language Processing (NLP) and Information Retrieval (IR). For example, is relatedness alone sufficient $\{good, bad\}$ or do they have to be semantically similar $\{excellent, great\}$?



Timothy Weale

Putting Word Relatedness Metrics To Work

Research Variables

- 1 **Data Source** – Wikipedia. 2008.01.03 version used.
- 2 **Target Task** – Word Pair Relatedness Metrics and TREC Evaluation Sets
- 3 **Function Form** – What are the current graph-based function forms? Can we improve the function form to improve our results on the target tasks?



Timothy Weale

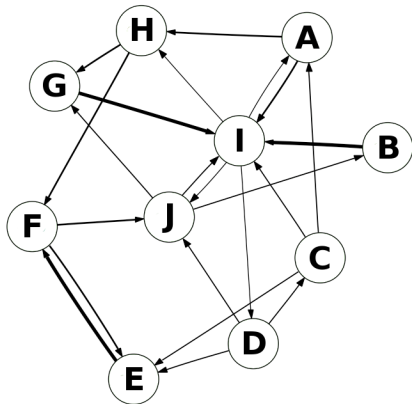
Putting Word Relatedness Metrics To Work

Overview

The Main Idea

Related words will share many common neighbors.

Wikipedia provides direct page-page links that create an overall Wikipedia graph.



- **Local:** Relationships are based on a comparison of the page-page links used by a vertex of the graph.
- **Global:** Treat the page-page graph as a searchable 'web' for related pages.



Timothy Weale

Putting Word Relatedness Metrics To Work

Local Graph Relatedness

- Related pages will have related link structures.
- “*Star Wars*” will have more neighbors in common with “*George Lucas*” than “*Potato*”.
- Derive relatedness based on two different metrics:
 - Cosine Metric [Mil07]
 - Google Normalized Distance [MW08]



Timothy Weale

Putting Word Relatedness Metrics To Work

Local Graph Relatedness [Mil07]

- Some links are more informative than others
 - Over 9000 pages link to the article on *water* (a common chemical substance)
 - Only 34 pages link to the article on *focaccia* (a type of bread)
- The less likely a link is, the more information it contains

$$w(s \rightarrow t) = \log \left(\frac{|W|}{|T|} \right) \text{ if } s \in T \quad (2)$$

- Creates a weighted vector of links: $\langle w(s, t_1), \dots, w(s, t_{|W|}) \rangle$
- $rel(a, b) = \cos(\vec{A}, \vec{B})$



Local Graph Relatedness [MW08]

- Based on Google Normalized Distance

$$dist(a, b) = \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|, |B|))} \quad (3)$$

Where A, B are the sets of pages that link to page *a*, *b* respectively



Global Graph Relatedness [OS07]

Leverage PageRank for vertex relatedness determination.

$$\vec{P}_n = \alpha \times (P_{n-1} \times T) + (1 - \alpha) \times \vec{W} \quad (4)$$

- \vec{P} : **PageRank Vector**. Holds the measure of vertex 'importance'.
- T : **Transition Matrix**. Contains edge transition probabilities.
- \vec{W} : **Walk Vector**. Uniform vector used to model 'random walk' behavior.
- α : **Model Weighting**. Tradeoff between strict PageRank behavior and Random Walk behavior (usually 0.85)



Global Graph Relatedness [OS07]

- PageRank gives an a priori indication of importance.
"How important is this vertex to the graph?"
- We're really interested in:
"How important is this vertex, given a starting point?"
- To determine this, we "pour" weight into the graph at our given vertex and "drain" weight from the graph at each vertex.
- The amount to drain at each vertex is determined by its a priori importance (PageRank value)



Global Graph Relatedness [OS07]

$$\vec{R}_n = R_{n-1} \times T + (\vec{S} - \vec{P}) \quad (5)$$

- \vec{R} : **Relatedness Vector**.
- T : **Transition Matrix**. Edge transition probabilities (same as PageRank).
- \vec{S} : **Source Vector**. Contains a 1 at the source vertex and zeros in all other locations.
- \vec{P} : **Sink Vector**. Subtracts a value from the vertex values based on their PageRank value.

$$rel(a, b) = \vec{R}_a[b] \times \log \left(\frac{1}{\vec{P}[b]} \right) \quad (6)$$



Implementation

- 2008.01.03 Wikipedia Data Set
- Specific files to utilize from the data set:
 - *page.sql*: Basic Structure Information
 - *redirect.sql*: Page-Page Redirects
 - *pagelinks.sql*: Page-Page Links
 - *pages-articls.xml*: Article text



page.sql

- Wikipedia is organized as a collection of “page” objects
- Each page has the following items associated with it:
 - **ID** Internal page ID. Generally increasing based on date page was created.
 - **Title:** Page name. Does not contain spaces.
 - **Namespace:** Page type.
 - **Redirect:** Boolean value. Based on if the page is active or not.
 - **Last Edited:** Date of last edit.



Timothy Weale

Putting Word Relatedness Metrics To Work

Wikipedia Data

- Every page belongs to one of several potential types:
 - **Main:** Encyclopedia-type pages. “Content” pages. (0)
 - **Talk:** Discussion and coordination pages for updating main pages. (1)
 - **User:** Information about Wikipedia contributors. (2)
 - **Wikipedia:** Wikipedia-specific information for high-level coordination. (4)
 - **File:** Uploaded Information (images, etc.). (6)
 - **Template:** Common information structures (biography, see also, etc.). (10)
 - **Help:** General help over all Wikipedia-based projects. (12)
 - **Category:** User-defined category structure for organization/navigation. (14)
 - **Portal:** Organizing structure for topic-based information. (100)



Timothy Weale

Putting Word Relatedness Metrics To Work

redirect.sql

- 'Non-active' pages (pages based on misspellings, alternative spellings, acronyms) redirect to 'active' pages automatically through the redirect table
- Each redirect entry is a triplet of the following form:
 - **ID:** ID of the 'from' page.
 - **Namespace:** Page type.
 - **Title:** Page name of the 'to' page.



Timothy Weale

Putting Word Relatedness Metrics To Work

Implementation

2008.01.03 Wikipedia Data Set

Page Counts	
All Pages	11,372,877
Main Namespace	4,610,749
Final Graph Vertices	2,179,265



Timothy Weale

Putting Word Relatedness Metrics To Work

pagelinks.sql

- Each page entry is a triplet of the following form:
 - **ID:** ID of the 'from' page.
 - **Namespace:** Page type.
 - **Title:** Page name of the 'to' page.

Page IDs must be valid pages.

Pages must link to valid pages.



Timothy Weale

Putting Word Relatedness Metrics To Work

Implementation

2008.01.03 Wikipedia Data Set

Link Counts	
All Links	208,173,450
Main Namespace	136,492,176
Non-Redirected Main	80,710,797



Timothy Weale

Putting Word Relatedness Metrics To Work

Link Text

We assume that each vertex corresponds to one 'idea' in the world.

We need a way to connect vertices
to the words used to describe the idea.

Each vertex has a title. However, this is a
limited and very specific method of describing the vertex.

Wikipedia has its own HTML-like markup language. If we can
extract the way wikipedia authors refer to a vertex by extracting
the link text, we can get a list of real-world instances of words that
correspond to the target vertex. [Mih07, MW08]



articles-pages.xml

Using the 2007.05.27 version of pages-articles.xml

- 11.3 GB when extracted
- 187,228,767 lines of text



Link Text

Links are indicated by the following structure:
 [[TEXTGOESHERE]]

Wiki Formatting	Surface Words
[[America]]	America
[[Barack_Obama President Obama]]	President Obama
[[#Document Section]]	Document Section
[[Education#School grades fifth grade]]	fifth grade
[[es:Wii]]	(not displayed)
[[Category: Economies]]	Economies (Placed in Category Box)



Link Text

Works well in the small-scale. However, there are difficulties of scale with the full data set.

Can be parallelized for reduced runtime.

Current solution is to use (for all pages and redirected pages):

- 1 **Article Titles:** { "Barack Obama", "Barack H. Obama", "Senator Obama", "O'Bama" }
- 2 **Non-disambiguated Title Information:** { "bar (counter)", "bar (unit)", "bar (law)", "bar (music)" }



Graph Structure

Demo



Timothy Weale

Putting Word Relatedness Metrics To Work

Java Hints for Large-Scale Data Work

Java Strings take up a LOT of memory on their own.

- Creating a substring copies the entire string and just changes the offset.
- Use `new String()` to get a String for minimum memory usage.

Reading/Writing binary objects can save a lot of time/overhead.

- `ObjectOutputStream` keeps references to written objects.
- Use `.reset()` to have the `OutputStream` 'forget' these objects for garbage collection.



Timothy Weale

Putting Word Relatedness Metrics To Work

About Lemur

Lemur is an open-source toolkit used in Information Retrieval research. Joint work between CMU and University of Massachusetts.

Includes many programs for IR evaluation (corpus indexing, document retrieval).

Comprehensive API for Java, C#, C++ and other languages.

Currently using version 4.6



Timothy Weale

Putting Word Relatedness Metrics To Work

Indexing

Quickly and efficiently map documents to words and vice-versa.

Lemur provides the **BuildIndex** program to effectively index corpora. This program takes a xml parameter file as input:

- *index*: Location to write the index
- *datafiles*: List of all files to be indexed (directory location is not enough!)
- *stopwords*: text file containing the list of stop words*
- *stemmer*: type of stemmer used* (porter)



Timothy Weale

Putting Word Relatedness Metrics To Work

Retrieval

Quickly and efficiently retrieve documents based on (1) input query and (2) retrieval function.

Lemur provides the **RetEval** program for document retrieval. This program takes a xml parameter file as input:

- *index*: Location of the index to be searched
- *retModel*: Retrieval Model
- *textQuery*: xml file containing the set of queries to be run
- *resultCount*: top-n documents to retrieve for each query



Timothy Weale

Putting Word Relatedness Metrics To Work

Lemur's TFIDF retrieval method is based on a standard vector-inspired view of documents and queries:

$$s(\vec{d}, \vec{q}) = \sum_{i=1}^n tf_d(x_i) \times tf_q(y_i) \times idf(t_i)^2 \quad (7)$$

Lemur also includes other standard retrieval functions

- OKAPI
- KL-Divergence



Timothy Weale

Putting Word Relatedness Metrics To Work

[FZ05] derived a new retrieval function, F2EXP and demonstrated its effectiveness in IR.

$$S(Q, D) = \sum_{t \in Q \cap D} c(t, Q) \cdot \left(\frac{N}{df(t)} \right)^{0.35} \cdot \frac{c(t, D)}{c(t, D) + b + \frac{b \cdot |D|}{avdl}} \quad (8)$$

Retrieval function used in subsequent evaluations of retrieval and relatedness [FZ06, Fan08].

F2EXP has been re-implemented for Slate's copy of Lemur as part of this work.



TREC: Text REtrieval Conference

TREC 1 – 8 have included various “bake-offs” of IR techniques using a variety of corpora and queries.

TREC	Documents	Topics
TREC-1	Disks 1 & 2	51 – 100
TREC-2	Disks 1 & 2	101 – 150
TREC-3	Disks 1 & 2	151 – 200
TREC-4	Disks 2 & 3	201 – 250
TREC-5	Disks 2 & 4	251 – 300
TREC-6	Disks 4 & 5	301 – 350
TREC-7	Disks 4 & 5	351 – 400
TREC-8	Disks 4 & 5	401 – 450
TREC-8	WT2G	401 – 450



TREC Topic Files

Each TREC evaluation included a set of standardized queries to be run over the corpus

- **TREC 1 & 2:** This topic format includes many different fields not included in subsequent revisions, including information on query concepts, definitions and domains.
- **TREC 4:** The shortest of all topic formats, this only includes the topic number and the topic description.
- **TREC 3 & 5–8:** Format includes number, title, description and narrative. Later topic formats tend to have a shorter average text length than TREC-3 formatting.



Timothy Weale

Putting Word Relatedness Metrics To Work

<top>

<num> Number: 401

<title> foreign minorities, Germany

<desc> Description: What language and cultural differences impede the integration of foreign minorities in Germany?

<narr> Narrative: A relevant document will focus on the causes of the lack of integration in a significant way; that is, the mere mention of immigration difficulties is not relevant. Documents that discuss immigration problems unrelated to Germany are also not relevant.

</top>



Timothy Weale

Putting Word Relatedness Metrics To Work

QREL Files

Relatedness is compared to the given QREL file for the TREC data set.

The QREL file contains a list of documents determined to be 'relevant' to a given query by the experts.

Binary relevancy determination and no overall ordering of relevancy.

```
401 0 WT02-B12-218 0
401 0 WT02-B12-219 1
401 0 WT02-B12-220 1
```



Type	Source	Metric	TREC7	TREC8	WT2G
TFIDF	T	MAP	0.1765	0.2356	0.2683
		GMAP	0.0637	0.1330	0.0366
OKAPI	T	MAP	0.1669	0.2345	0.2488
		GMAP	0.0664	0.1308	0.1578
KL	T	MAP	0.1775	0.2536	0.2920
		GMAP	0.0705	0.1496	0.1885
F2EXP ¹	T	MAP	0.186	0.250	0.282
		GMAP	0.083	0.147	0.188
F2EXP ²	T	MAP	0.1792	0.2474	0.2898
		GMAP	0.0686	0.1462	0.1944
HUI QE_{def}	T	MAP	0.254	0.266	0.301
		GMAP	0.088	0.164	0.210

¹Reported Results

²Our Implementation



Query Expansion

$$Q = \{q_1, q_2, \dots, q_i\}$$

Can we augment the query with set of words that will improve the returned results of the query?

$$Q' = \{q_1 : v_1, q_2 : v_2, \dots, q_i : v_i, q_{i+1} : v_{i+1}, \dots, q_{i+n} : v_{i+n}\}$$

Non-standard term weighting has been added to Slate's implementation of the Lemur Toolkit.



Timothy Weale

Putting Word Relatedness Metrics To Work

Query Expansion

For each query word find top k related words

$$K_1 = \text{topN}(q_1, W, f(\cdot)); \dots$$

Add the top- k words to a candidate list

$$C = \{K_1, K_2, \dots, K_n\}$$

Re-rank the candidate list

$$\text{rerank}(C)$$

Pull out the top n words for expansion



Timothy Weale

Putting Word Relatedness Metrics To Work

Vertex Relatedness

Demo



Timothy Weale

Putting Word Relatedness Metrics To Work

Vertex Relatedness

Raw vertex relatedness values may not be the best method of determining query expansion values.

- *Rank values*: Function of the relative ranking of each expanded word
- *Raw values*: Function of the raw values from the graph relatedness



Timothy Weale

Putting Word Relatedness Metrics To Work

Right now, all outbound/inbound transitions are based on uniform transition probability.

This is not the case: *Harrison Ford* is more connected to *Star Wars* or *Indiana Jones* than *arachnologist*.

Is there a way to leverage additional information in Wikipedia to get non-uniform transition weights?



Category Weighting

There's a large body of work on ontology-based relatedness. Can we leverage this by using the Wikipedia category structure?

One problem: Wikipedia is not a tree.

Potential solution: Seed multiple trees, each starting from a different "fundamental" category: { Information, Nature, Society, Structure, Thought }

Now, use path- or information-based relatedness to get a new weighting for the links.



How do we define category probability?

Definition	<i>GetCount()</i>	Category Probability
Vertex Count	1	$p_{count}(c_n) = \frac{ V(c_n) }{ V(G) }$
Inbound Count	$ E^-(v_i) $	$p_{in}(c_n) = \frac{ E^-(c_n) }{ E(G) }$
Outbound Count	$ E^+(v_i) $	$p_{out}(c_n) = \frac{ E^+(c_n) }{ E(G) }$



Text Overlap Weighting

Can we use traditional text overlap metrics (Jaccard, Overlap, etc.) to get an effective vertex relatedness metric?

$$w(v_1 \rightarrow v_2) = w(v_2 \rightarrow v_1) = \text{sim}(v_1.\text{text}, v_2.\text{text}) \quad (9)$$



Temporal Weighting

We have multiple copies of the Wikipedia link structure capturing several different times over the past year or so.

Can we use these copies to get a measure of how confident we are in this relationship?

Use existing graphs to get a new graph / transition probabilities:

$$f(G_0, \dots, G_i) \Rightarrow G' \quad (10)$$

Use a combination of existing relatedness values to determine an overall relatedness value:

$$f(\text{rel}(v_1, v_2, G_0), \dots, \text{rel}(v_1, v_2, G_i)) \Rightarrow \text{rel}(v_1, v_2) \quad (11)$$



Conclusions

Wikipedia as a data source has many challenges but many untapped areas of exploration for improvement and refinement.

Lemur toolkit is a useful tool for IR evaluation and the TREC data sets provide a standardized test set for evaluation.

Much work left to be done in this domain, but much of the implementation structure has been developed and is available for investigation.





Hui Fang.

A Re-examination of Query Expansion Using Lexical Resources.

In *Proceedings of ACL-08: HLT*, Columbus, Ohio, June 2008. Association for Computational Linguistics.



Hui Fang and ChengXiang Zhai.

An Exporation of Axiomatic Approaches to Information Retrieval.

In *Proceedings of SIGIR 2005*, 2005.



Hui Fang and ChengXiang Zhai.

Semantic Term Matching in Axiomatic Approaches to Information Retrieval.

In *Proceedings of SIGIR 2006*, pages 115 – 122, 2006.



Rada Mihalcea.

Using Wikipedia for Automatic Word Sense Disambiguation.

In *Proceedings of NAACL HLT 2007*, pages 196–203, 2007.



David Milne.

Computing Semantic Relatedness using Wikipedia Link Structure.

In *New Zealand Computer Science Research Student Conference (NZCSRSC'2007)*, 2007.



David Milne and Ian H. Witten.

An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links.

In *Proceedings of AAAI 2008*, 2008.



Yann Ollivier and Pierre Senellart.

Finding Related Pages Using Green Measures: An Illustration with Wikipedia.

In *Proceedings of AAAI 2007*, 2007.

