CS 572: Information Retrieval

Web 2.0+: Social Media/CGC
Status

- Final project: updated proposal due by today/tomorrow
- Questions? Discuss at end of class.
• Collaboratively Generated Content: CGC
  – Creation models, quality
• Twitter/Weibo
• Searching:
  – Indexing
  – Ranking
## Estimates of daily content creation

[Ramakrishnan & Tomkins, IEEE Computer 2007]

<table>
<thead>
<tr>
<th>Content type</th>
<th>Amount produced / day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Published (books, magazines, newspapers)</td>
<td>3-4 Gb</td>
</tr>
<tr>
<td>Professional Web (paid creation, e.g., corporate Web site)</td>
<td>2 Gb</td>
</tr>
<tr>
<td><strong>User-generated (reviews, blogs, personal Web sites)</strong></td>
<td><strong>8-10 Gb</strong></td>
</tr>
<tr>
<td>Private text (instant messages, email)</td>
<td>3000 Gb (= 3Tb)</td>
</tr>
</tbody>
</table>
The value of CGC

- **Extrinsic value** – impact on AI, IR and NLP applications

- **Intrinsic value** – resources for the people
  - How to make the data more useful (e.g., increase user engagement)

- **Synergistic view of CGC**
  - AI techniques can be used to improve the quality of CGC repositories,
  - which can in turn improve the performance of other AI applications
Extrinsic value: sources of knowledge

Collaboratively Generated Content

Before

articles: 65,000
Since 1768

entries: 150,000
Since 1985

assertions: 4,600,000
Since 1984

WordNet

CYC

After

3,600,000 articles
Since 2001

2,500,000 entries
Since 2002

1,000,000,000 Q&A
Since 2005

4,000,000,000 image
Since 2004

Britannica

Wikipedia

Wiktionary

Yahoo!

flickr
Extrinsic value: what we can do with all this knowledge

• Under the hood
  – CGC as an additional (huge!) corpus
    • Better term statistics, observe the document authoring process
  – Repositories of knowledge
    • Concepts as features (BOW $\rightarrow$ Bag of Words + Concepts)
    • Concepts as word senses (way beyond WordNet)
  – More comprehensive lexicons, gazetteers
  – Extending existing knowledge repositories (e.g., WordNet)

• Higher-level tasks
  – (Concept-based) information retrieval
  – Question answering
Yes, we could read your blog. Or, you could tell us about your day.
Models for CGC generation: Overview

• Wikipedia: contribution, curation, and coordination

• Community forums: Yahoo! Answers, focused forums

• Sharing: twitter, flickr, YouTube
Models of CGC Generation: Link Creation
“Rich-Get-Richer” (for a while ...)

- Flickr and other sites also show strong evidence of preferential attachment [Mislove et al. ‘08]

Preferential attachment:
More popular Wikipedia pages are more likely to acquire links (up to a point!) [Capocci et al. ‘06]
Models of CGC Generation: Edits
“Rich-Get-Richer” (again, up to a point)

Popular Wiki pages (> 1000 edits):
- Editing activity initially increases than drops off
- Later on, arriving visitors are less able to contribute novel information

[Das and Magdon-Ismail ‘10]
Models of CGC generation: Slowing Growth [Suh et al., WikiSym 2009]

- Article growth (#) per month, extrapolated to slow down dramatically by 2013
- Increased coordination and overhead, resistance to new edits
Models for CGC generation

Wikipedia conflict and coordination

[Kittur & Kraut, 2008]

- Explicit coordination: discussion page
- Implicit coordination: leading the effort

Example: “Music of Italy” page – The bulk of edits are made by TUF-KAT and Jeffmatt

- Page quality increases with number of editors iff concentrated effort; otherwise, more editors → higher overhead, lower quality
Models of CGC Generation: Content Visibility and Evolution

• Dynamics of votes of select Digg stories over a period of 4 days [Lerman ‘07]

• Dynamics of Y! Answers posts over a period of 4 days [Aji et al. ‘10]
Models for CGC Generation: Forums
Info seekers vs. info providers

- In Yahoo! Answers, question and answer quality are strongly related: bad questions attract bad or mediocre answers: [Agichtein et al. ’08]

- Askers (Java forums) prefer help from users with just slightly more expertise [Zhang et al. ‘07]
Lots of content... Do we trust it?

On the Internet, nobody knows you are a dog.
CGC Quality Filtering (Overview)

• Multiple dimensions of CGC content to exploit:
  – Content persistence (Wikipedia)
  – Surface quality (language-wise): all CGC
  – Metadata: Author reliability, geo-spatial edit patterns
  – Citation/link-based estimation of authority

• Most effective systems use a combination of all of the above
Content-Driven Author Reputation

- Authors of long-lived contributions gain reputation
- Authors of reverted contributions lose reputation
Automatic Vandalism Detection

- Hoaxes
- Libelous content
- Malware distribution
- Errors picked up by mainstream media!
- Wikipedia “truthiness”

Benjamin Franklin (January 17, 1706 [O.S. January 6, 1705[1]] – April 17, 1790) was one of the Founding Fathers of the United States. A noted polymath, Franklin was a leading author and printer. He was a satirist, political theorist, politician, postmaster, scientist, inventor, civic activist, statesman, and diplomat. As a scientist, he was a major figure in the American Enlightenment and the history of physics for his discoveries and theories regarding electricity. He invented the lightning rod, bifocals, the Franklin stove, a carriage odometer, and the glass 'armonica'. He formed both the first public lending library in America

Merriam-Webster's #1 Word of the Year for 2006:

1. **truthiness** (noun)
   1 : “truth that comes from the gut, not books” (Stephen Colbert, Comedy Central’s "The Colbert Report," October 2005)

2 : “the quality of preferring concepts or facts one wishes to be true, rather than concepts or facts known to be true” (American Dialect Society, January 2006)
Estimating *text trust* from edit history information.

Trust is computed as:

1) **Insertion**: trust value for newly inserted text (E)
2) **Edge effect**: the text at the edge of blocks has the same trust as newly inserted text.
3) **Revision**: text trust may increase, if author reputation gets higher
NLP-based Vandalism Detection

[Wang et al., COLING 2010]

- Use language modeling to predict edit quality
  - Lexical: vocabulary, regular expression/lists
  - Syntactic: N-gram analysis using part-of-speech tags
  - Semantic: N-gram analysis with hand-picked words

Figure adapted from Andrew G. West, 2010
Comparison of Vandalism Detection Approaches: Shared tasks [Potthast et al. ‘10]

- Shared vandalism corpus, 32,000 edits
- Best single approach: NLP-based
- Meta-detector: best
  - Task requires different classes of features.

- CLEF 2010 Competition on Vandalism Detection:
  - Best result: AUC 0.92, Precision = 100% at Recall = 20%
Finding High Quality Content in SM

- Well-written
- Interesting
- Relevant (answer)
- Factually correct
- Popular?
- Provocative?
- Useful?

As judged by professional editors


Eugene Agichtein, Emory University, IR Lab
Do I have a shot at Emory University?

I have an unweighted 3.73 GPA on a 4.0 scale, and a weighted 3.82. I've only taken a couple honors classes throughout high school (Chemistry and Math 9) and no APs, but I'm taking two APs this year (senior year) (Economics and Psychology). I've taken the ACTs twice and scored a 29 Composite with a 9 out of 12 on the

Best Answer - Chosen by Voters

Your GPA is average for Emory. However, the average Emory student has more AP classes than you do. You are on the right track taking more -- but you aren't there yet.

Your ACT score corresponds to an SAT score of about 1920-1980. Over 75% of those who are accepted at Emory have higher SAT scores.
How do Question and Answer Quality relate?

<table>
<thead>
<tr>
<th>Answer Quality</th>
<th>Question Quality</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A. High</td>
<td>A. High</td>
<td>41%</td>
<td>15%</td>
<td>8%</td>
</tr>
<tr>
<td>B. Medium</td>
<td>B. Medium</td>
<td>53%</td>
<td>76%</td>
<td>74%</td>
</tr>
<tr>
<td>C. Low</td>
<td>C. Low</td>
<td>6%</td>
<td>9%</td>
<td>18%</td>
</tr>
<tr>
<td>Total</td>
<td>Total</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Text analysis

Readability statistics

Language modeling

Punctuation density

[Example]: Help! math! histogram! asap?

[Details]: In Mathematics - Asked by MarkyMe123 - 0 answers - 3 minutes ago

Capitalization errors

[Example]: WHAT is heidi montag thinking WITH THIS MUSIC VIDEO?

[Details]: In Celebrities - Asked by chris bann88 - 0 answers - 3 minutes ago

Number of words

[Example]: Help!!!!!!!!!!!!

[Details]: In General - Asked by *So Confused* - 1 answer - 6 minutes ago

+ spacing density, syllables per word,...
Text analysis

Readability statistics

Language modeling

Language model disagreement
Distributions of word n-grams and part-of-speech sequences

when | how | why -- “to” -- verb

“how to identify …”

when | how | why -- verb -- verb -- pronoun -- verb

“how do I remove …”

Do I have a shot at Emory University?

I have an unweighted 3.73 GPA on a 4.0 scale, and a weighted 3.82. I've only taken a couple honors classes throughout high school (Chemistry and Math 9) and no APs, but I'm taking two APs this year (senior year) (Economics and Psychology). I've taken the ACTs twice and scored a 29 Composite with a 9 out of 12 on the

Best Answer - Chosen by Voters

Your GPA is average for Emory. However, the average Emory student has more AP classes than you do. You are on the right track taking more -- but you aren't there yet.

Your ACT score corresponds to an SAT score of about 1920-1980. Over 75% of those who are accepted at Emory have higher SAT scores.
Link Analysis for Authority Estimation

\[ A(j) = \sum_{i=0}^{M} H(i) \]

\[ H(i) = \sum_{j=0}^{K} A(j) \]

[Jurzcyk & Agichtein, CIKM 2007]
Qualitative Observations

HITS effective ➔

← HITS ineffective

Eugene Agichtein, Emory University, IR Lab
Do I have a shot at Emory University?

I have an unweighted 3.73 GPA on a 4.0 scale, and a weighted 3.82. I've only taken a couple honors classes throughout high school (Chemistry and Math 9) and no APs, but I'm taking two APs this year (senior year) (Economics and Psychology). I've taken the ACTs twice and scored a 29 Composite with a 9 out of 12 on the

Best Answer - Chosen by Voters

Your GPA is average for Emory. However, the average Emory student has more AP classes than you do. You are on the right track taking more -- but you aren't there yet.

Your ACT score corresponds to an SAT score of about 1920-1980. Over 75% of those who are accepted at Emory have higher SAT scores.
## Result 1: Identifying High Quality Questions

<table>
<thead>
<tr>
<th>Method</th>
<th>High qual.</th>
<th></th>
<th>Normal/low qual.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Text (Baseline)</td>
<td>0.654</td>
<td>0.481</td>
<td>0.762</td>
<td>0.867</td>
</tr>
<tr>
<td>Usage</td>
<td>0.594</td>
<td>0.470</td>
<td>0.755</td>
<td>0.836</td>
</tr>
<tr>
<td>Relation</td>
<td>0.694</td>
<td>0.603</td>
<td>0.806</td>
<td>0.861</td>
</tr>
<tr>
<td>Intrinsic</td>
<td>0.746</td>
<td>0.650</td>
<td>0.829</td>
<td>0.885</td>
</tr>
<tr>
<td>T+Usage</td>
<td>0.683</td>
<td>0.571</td>
<td>0.798</td>
<td>0.865</td>
</tr>
<tr>
<td>T+Relation</td>
<td>0.739</td>
<td>0.647</td>
<td>0.828</td>
<td>0.881</td>
</tr>
<tr>
<td>T+Intrinsic</td>
<td>0.757</td>
<td>0.650</td>
<td>0.830</td>
<td>0.891</td>
</tr>
<tr>
<td>T+Intr.+Usage</td>
<td>0.717</td>
<td>0.690</td>
<td>0.845</td>
<td>0.861</td>
</tr>
<tr>
<td>T+Relation+Usage</td>
<td>0.722</td>
<td>0.690</td>
<td>0.845</td>
<td>0.865</td>
</tr>
<tr>
<td>T+Intr.+Relation</td>
<td><strong>0.798</strong></td>
<td><strong>0.752</strong></td>
<td><strong>0.874</strong></td>
<td><strong>0.901</strong></td>
</tr>
<tr>
<td>All</td>
<td>0.794</td>
<td><strong>0.771</strong></td>
<td>0.885</td>
<td>0.898</td>
</tr>
</tbody>
</table>
Top Features for Question Classification

• Asker popularity ("stars")

• Punctuation density

• Topical category

• Page views

• KL Divergence from reference corpus LM
## Identifying High Quality Answers

<table>
<thead>
<tr>
<th>Method</th>
<th>High qual.</th>
<th>Normal/low qual.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Text (Baseline)</td>
<td>0.668</td>
<td>0.862</td>
</tr>
<tr>
<td>Relation</td>
<td>0.552</td>
<td>0.617</td>
</tr>
<tr>
<td>Intrinsic</td>
<td>0.712</td>
<td>0.918</td>
</tr>
<tr>
<td>T+Relation</td>
<td>0.688</td>
<td>0.851</td>
</tr>
<tr>
<td>T+Intrinsic</td>
<td>0.711</td>
<td><strong>0.926</strong></td>
</tr>
<tr>
<td>All</td>
<td><strong>0.730</strong></td>
<td>0.911</td>
</tr>
</tbody>
</table>
Top Features for Answer Classification

- Answer length
- Community ratings
  - Answerer reputation
- Word overlap
- Kincaid readability score
CGC as a corpus

- Better statistics on a larger or cleaner text collection
  - Compute IDF on the entire Wikipedia corpus
    - Beware of different style / topic distribution
  - Cf. Web as a corpus (Mihalcea’04, Gonzalo et al.’03)
    - Web corpus statistics, such as counts of search results
  - Cf. using very large corpora in NLP (Banko and Brill ‘01a, ‘01b)
CGC as a corpus (cont’d)

• Extend existing knowledge repositories (e.g., WordNet)

• Create new datasets, labeled by construction
  – WSD datasets using OpenMind – Chklovski & Mihalcea ‘02, ‘03
  – WSD datasets based on Wikipedia – Mihalcea’07; Bunescu & Pasca ‘06; Cucerzan ‘07
  – Text categorization datasets based on the ODP (Davidov et al. ‘04)

• Study the authoring process by observing multiple revisions of a document
Using corpus statistics for term weighting

- J. Elsas and S. Dumais. Leveraging temporal dynamics of document content in relevance ranking. *WSDM 2010*
  - Periodic crawls (starting from an unknown point in document’s life) vs. formally tracked revision history
  - LM-specific vs. general term frequency model, which can be plugged into any retrieval model
  - Aji et al.’10 also model bursts – “significant” events in the document’s life

- M. Efron. Linear time series models for term weighting in information retrieval. *JASIST 2010*
  - Temporal behavior of terms as the *entire collection* changes over time
Extending term weighting models with Revision History Analysis (RHA) [Aji et al. ’10]

- **RHA** captures term importance from the document authoring process.
- Natural integration with state of the art retrieval models (BM25, LM)
- Consistent improvement over existing retrieval models
- Beyond Wikipedia
  - Tracking changes in MS Word documents; version control
  - Past snapshots of Web pages
    (Elsas & Dumais ’10 constructed a new corpus)
Observable document generation process
Example: Revisions of “Topology” in Wikipedia

Topology
From Wikipedia, the free encyclopedia

1st revision: Topology, in mathematics, is both a structure used to capture the notions of continuity, connectedness and convergence, and the name of the branch of

250th revision: Topology is the study or science of places. It derives its name from the Greek words τόπος meaning place and λόγος meaning study, talk. See also: earth science, physical geography, human geography, geomorphology

In architecture, topology is a term used to describe spatial effects which cannot be described by topography, i.e., social, economical, spatial or phenomenological

In mathematics, topology is a branch concerned with the study of topological spaces. (The term topology is also used for a set of open sets used to define topological network topologies are discussed in network topology.)

Current revision: Topology (from the Greek τόπος, “place”, and λόγος, “study”) is a major area of mathematics concerned with spatial properties that are preserved under deformations that involve stretching, but no tearing or gluing. It emerged through the development of concepts from geometry and set theory that are now classified as topological. Ideas that are now classified as topological were expressed as early as 1736. Toward the end of the 19th century, a distinct discipline called “geometry of places” or analysis situs (Greek-Latin for “picking apart of place”). This later acquired the modern name of topology. By the study within mathematics.

The word topology is used both for the mathematical discipline and for a family of sets with certain properties that are used to define a notion of homeomorphisms, which can be defined as continuous functions with a continuous inverse. For instance, the function $y = x^2$ is a homeomorphism.

Topology includes many subfields. The most basic and traditional division within topology is point-set topology, which establishes the basic concepts of topological spaces (basic examples include compactness and connectedness); algebraic topology, which generally tries to measure groups and homology; and geometric topology, which primarily studies manifolds and their embeddings (placements) in other manifolds, and graph theory, do not fit neatly in this division.
Revision History Analysis
redefines Term Frequency (TF)

- TF is a key indicator of document relevance
- TF can be naturally integrated into ranking models

\[
S(Q,D) = \sum_{t \in Q} IDF(t) \cdot \frac{TF(t,D) \cdot (k_1 + 1)}{TF(t,D) + k_1 \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}
\]

BM25

\[
S(Q,D) = D(Q||D) = \sum_{t \in V_Q} P(t|Q) \cdot \log \frac{P(t|Q)}{P(t|D)}
\]

Language Model
Define the term frequency based on the whole document generation process

- a document grows steadily over time
- a term is relatively important if it appears in the early revisions

\[ TF_{global}(t, d) = \sum_{j=1}^{n} \frac{c(t, v_j)}{j^\alpha} \]

- Frequency of term \( t \) in revision \( v_j \)
- Decay factor

Iteration over revisions
RHA **global model**: example

**Decayed term importance**

**Decay factor**

CGC as corpus / Document revisions
But some pages are different:
“Avatar (2009 film)” – very bursty editing process
RHA burst model

- A burst resets the decay clock for a term
- The weight will decrease after a burst

\[ TF_{burst}(t, d) = \sum_{j=1}^{m} \sum_{k=b_j}^{n} c(t, v_k) \frac{(k - b_j + 1)^\beta}{\beta} \]

Frequency of term \( t \) in revision \( v_k \)

Decay factor for \( j^{th} \) burst

Iteration over bursts

Iteration over revisions within each burst

Burst detection

- Content-based (relative content change)
- Activity-based (intensive edit activity)
- Both models combined
RHA **burst** model: example

**pandora**

**Decay factor**

**Term importance**
Combining the **global** model with the **burst** model:

\[
TF_{rha}(t, D) = \lambda_1 \cdot TF_g(t, D) + \lambda_2 \cdot TF_b(t, D) + \lambda_3 \cdot TF(t, D)
\]

\[
\lambda_1 + \lambda_2 + \lambda_3 = 1
\]
Integrating RHA into retrieval models

**BM25 + RHA**

\[
S(Q, D) = \sum_{t \in Q} IDF(t) \cdot \frac{TF_{rha}(t, D)}{TF_{rha}(t, D) + k_1 \left( 1 - b + b \cdot \frac{|D|}{\text{avgdl}} \right)} \cdot (k_1 + 1)
\]

**Statistical Language Models + RHA**

\[
S(Q, D) = D(Q||D) = \sum_{t \in V} P(t|Q) \cdot \log \frac{P(t|Q)}{P_{rha}(t, D)}
\]

RHA Term Probability:

\[
P_{rha}(t, D) = \lambda_1 \cdot P_g(t, D) + \lambda_2 \cdot P_b(t, D) + \lambda_3 \cdot P(t, D)
\]

Performance improvements: +2-7% (INEX), +1-4% (TREC)
Overview of Part 3

✓ Use CGC as an additional (huge!) corpus
  • Better term statistics
  • Observing the authoring process

➢ CGC as repositories of world knowledge
  • Distilling knowledge from CGC structure and content
  • Concepts as features
    (BOW ➔ Bag of Words + Concepts)
    – Semantic relatedness, and later IR
  • Concepts as word senses
    – WSD
CGC as repositories of senses and concepts

- Articles / concepts as representation features
  - From the bag of words to the bag of words + concepts
  - Semantic relatedness of words (and texts)
  - Later: concept-based IR

- Articles / concepts as word senses
  - Word sense disambiguation
Semantic relatedness of words and texts

• How related are

  cat ⇔ mouse

  Augean stables ⇔ golden apples of the Hesperides

  慕田峪 (Mutianyu) ⇔ 司 马 台 (Simatai)

• Used in many applications
  – Information retrieval
  – Word-sense disambiguation
  – Error correction (“Web site” or “Web sight”)

CGC as knowledge / Semantic relatedness
Evaluation criterion: correlation with human judgments

<table>
<thead>
<tr>
<th></th>
<th>Human score [0 .. 10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>journey</td>
<td>voyage</td>
</tr>
<tr>
<td>Jerusalem</td>
<td>Israel</td>
</tr>
<tr>
<td>money</td>
<td>property</td>
</tr>
<tr>
<td>football</td>
<td>tennis</td>
</tr>
<tr>
<td>alcohol</td>
<td>chemistry</td>
</tr>
<tr>
<td>coast</td>
<td>hill</td>
</tr>
<tr>
<td>professor</td>
<td>cucumber</td>
</tr>
</tbody>
</table>
Previous approaches

- **WordNet-based measures**
  - Using graph structure (path length, random walks), information content, glosses
  - Comprehensive review: Budanitsky & Hirst ‘06
  - Patwardhan et al. ‘03 – using semantic relatedness measures for WSD
    - Jiang-Conrath (WN structure + info content) performs on par with Adapted Lesk (gloss overlap)
    - All other WN-based measures are inferior to Adapted Lesk
- **Co-occurrence based measures**
  - Latent Semantic Analysis (Deerwester et al. ‘90)
When human judge semantic relatedness they use massive amounts of background knowledge.

Can we endow computers with such capability of knowledge-based reasoning?
Using Wikipedia for computing semantic relatedness

- **Structure (links)**
  - Strube & Ponzetto ‘06 (*WikiRelate*)
  - Milne & Witten ‘08 (*WLM*)

- **Content (text)**
  - Gabrilovich & Markovitch ‘07 (*ESA*)

- **Structure + content**
  - Yeh et al. ‘09 (*WikiWalk*)
Explicit Semantic Analysis (ESA)  
[Gabrilovich & Markovitch ‘06–’09]

• Wikipedia-based semantic representation

• Some ESA applications
  – Computing semantic relatedness of words and texts
  – Text categorization
  – Information retrieval
    • Cross-lingual retrieval utilizing cross-language links in Wikipedia
The AI Dream

Problem solver

Problem solver

Problem

Solution

CGC concepts as features
The circular dependency problem

Problem

Natural Language Understanding

Problem solver

Solution
Circumventing the need for deep Natural Language Understanding

Wikipedia-based semantics

Statistical text processing

Problem solver

Problem

Solution

CGC concepts as features
Every Wikipedia article represents a concept
Wikipedia can be viewed as an ontology – a collection of concepts
The semantics of a word is a vector of its associations with Wikipedia concepts.

Cf. Bengio’s talk tomorrow on learning text and image representations.
The semantics of a word is a point in the multi-dimensional space defined by the set of Wikipedia concepts.
The semantics of a text fragment is a **centroid** of the vectors of the individual words.
Panthera

From Wikipedia, the free encyclopedia

Not to be confused with Panther.

For other uses, see Panthera (disambiguation).

Panthera is a genus of the family Felidae (cats), which contains four well-known living species: the tiger, the lion, the jaguar, and the leopard. The genus comprises about half of the Pantherinae subfamily, the big cats. The word panther, while technically referring to all members of the genus, is commonly used to specifically designate the black panther.

Only the four Panthera cat species have the anatomical structure that enables them to roar. The primary reason for this was formerly assumed to be the incomplete ossification of the hyoid bone. However, new studies show that the ability to roar is due to other morphological features, especially of the larynx. The snow leopard, Uncia uncia, which is sometimes included within Panthera, does not roar. Although it has an incomplete ossification of the hyoid bone, it lacks the special morphology of the larynx.

Contents [show]

Name

See also: Panther (legendary creature)

According to the American Heritage Dictionary, the origin of the word is unknown. A folk etymology derives the word from the Greek πάνα ("all") and θήρ ("beast of prey") because they can hunt and kill almost everything. The Greek word πάνθηρ, pánther, referred to all spotted Felidae generically. Although it came into English through the classical
Computing semantic relatedness as cosine between two vectors.

CGC concepts as features.
### Experimental results (individual words)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Correlation with human judgments</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordNet-based</td>
<td>0.33–0.35</td>
</tr>
<tr>
<td>Roget’s Thesaurus</td>
<td>0.55</td>
</tr>
<tr>
<td>LSA</td>
<td>0.56</td>
</tr>
<tr>
<td>WikiRelate</td>
<td>0.49</td>
</tr>
<tr>
<td>WLM</td>
<td>0.69</td>
</tr>
<tr>
<td>ESA (ODP)</td>
<td>0.65</td>
</tr>
<tr>
<td><strong>ESA (Wikipedia)</strong></td>
<td><strong>0.75</strong></td>
</tr>
<tr>
<td>TSA [Radinsky et al. ‘11]</td>
<td>0.8</td>
</tr>
<tr>
<td>EWC (ESA/Wikipedia + WordNet + collocations)</td>
<td>0.79–0.87</td>
</tr>
<tr>
<td>[Haralambous &amp; Klyuev, forthcoming at IJCNLP’11]</td>
<td></td>
</tr>
</tbody>
</table>

CGC concepts as features
Using Wikipedia for Named Entity Disambiguation [Bunescu & Pasca ’06]

- Both the method **and** the dataset are based on Wikipedia
- Dataset: 1.7M labeled examples
  - "The [[Vatican City | Vatican]] is now an independent enclave ..."

The basic approach: compute the **cosine similarity** between the **word context** and the **target articles**
Problems with the basic approach

1. Short articles and stubs

2. Synonymy (lexical mismatch between the word context and the article)

Solution: use categories for generalization
Word-Category Correlations

When a Wikipedia article is too short, get **additional context from its categories**

```
John Williams and the Boston Pops conducted a summer Star Wars concert at Tanglewood.
```
The full solution [Bunescu & Pasca ’06]

• ML-based approach (SVM) to rank candidate senses

• Features:
  – Cosine(word context, article text)
  – Word-category affinities (binary features indicating if a word has occurred in any article of a given category)

• Dimensionality (|V| x |C|) can be reduced by thresholding word frequencies

• Accuracy improvement vs. cosine-only: 3%-25.5%
Results [Bunescu & Pasca ’06]

- The classification model can be trained on Wikipedia, then applied to any input text
  - Senses can only be resolved to those defined in Wikipedia

<table>
<thead>
<tr>
<th></th>
<th># Cat.</th>
<th>TRAINING DATASET</th>
<th>TEST DATASET</th>
<th># SV</th>
<th>TK(A)</th>
<th>Cos(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>QUERIES</td>
<td>Pairs $\langle q, e_k \rangle$</td>
<td># CONSTR.</td>
<td>QUERIES</td>
<td>Pairs $\langle q, e_k \rangle$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>110</td>
<td>12,288</td>
<td>39,880</td>
<td>27,592</td>
<td>48,661</td>
<td>147,165</td>
</tr>
<tr>
<td>$S_2$</td>
<td>540</td>
<td>17,970</td>
<td>55,452</td>
<td>37,482</td>
<td>70,468</td>
<td>235,290</td>
</tr>
<tr>
<td>$S_3$</td>
<td>2,847</td>
<td>21,185</td>
<td>64,560</td>
<td>43,375</td>
<td>75,190</td>
<td>261,723</td>
</tr>
<tr>
<td>$S_4$</td>
<td>540</td>
<td>38,726</td>
<td>102,553</td>
<td>63,827</td>
<td>80,386</td>
<td>191,227</td>
</tr>
</tbody>
</table>

- Another idea to overcome the lexical mismatch: augment Wikipedia articles with contexts of corresponding word senses
  - Cf. anchor text aggregation [Metzler et al. ’09] and [Cucerzan ‘07]
An alternative approach to named entity disambiguation [Cucerzan ‘07]

• Applicable to any text, not solely that from Wikipedia
  – Sense inventory still limited to Wikipedia

• The system performed both NE identification and disambiguation

• World knowledge used
  – Known entities (Wikipedia articles)
  – Their class (Person, Location, Organization)
  – Known surface forms (redirection pages + anchor text)
  – Words that describe or co-occur with the entity
  – Categories, lists, enumeration, tables
Restricted document representation: aggregation of features of all identified surface forms for all entity names
Key idea: joint disambiguation

- Maximize the agreement
  - between the entity vectors and the document, and
  - between the categories of the entities

- Example: \{Columbia, Discovery\}
  - Space Shuttle Columbia & Space Shuttle Discovery
  - Columbia Pictures & Discovery Investigations
  - \ LIST\_astronomical\_topics, CAT\_manned\_spacecraft, CAT\_space\_shuttles\

- If the “one sense per discourse” assumption is violated – shrink the context iteratively
Results [Cucerzan ‘07]

- **Baseline:** most frequent Wikipedia sense
- **Wikipedia dataset (labeled by construction)**
  - 86.2% ➞ 88.3%
- **MSNBC news stories (human labeling)**
  - 51.7% ➞ 91.4%
  - Surface forms common in news are highly ambiguous
  - Popularity in Wikipedia ≠ popularity in news

- Blackberry
- China
Case Study: Searching Tweets

Search @twitter

Michael Busch
@michibusch
michael@twitter.com
buschmi@apache.org
Twitter Statistics (2014)

- **284 million** monthly active users
- **500 million** tweets are sent per day
- More than **300 billion** tweets have been sent since company founding in 2006
- Tweets-per-second record: one-second peak of **143,199 TPS**.
- More than **2 billion** search queries per day.
Twitter Search Architecture

RT stream
raw tweets → Analyzer/Partitioner → analyzed tweets → RT index (Earlybird)

Tweet archive
HDFS
raw tweets → Mapreduce Analyzer → analyzed tweets → Archive index

Blender
Search requests

writes
searches
Search Architecture

- Blender is our Thrift service aggregator
- Queries multiple Earlybirds, merges results
Fig. 2. Organization of the active index segment where tweets are ingested. Increasingly larger slices are allocated in the pools to hold postings. Except for slices in pool 1 (the bottom pool), the first 32 bits are used for storing the pointer that links the slices together. Pool 4 (the top pool) can hold multiple slices for a term. The green rectangles illustrate the the “current” postings list that is being written into.
Twitter Search Architecture

- New Lucene extension package
- This package is truly generic and has no dependency on an actual product/index
- It contains Twitter’s extensions for real-time search, a thin segment management layer and other features
Twitter Lucene Extensions

- API layer for Lucene segments
  - *IndexSegmentWriter
  - *IndexSegmentAtomicReader
- Two implementations
  - In-memory: RealtimeIndexSegmentWriter (and reader)
  - On-disk: LuceneIndexSegmentWriter (and reader)
Twitter Lucene Extensions (cont’d)

- IndexSegments can be built ...
  - in realtime
  - on Mesos or Hadoop (Mapreduce)
  - locally on serving machines
- Cluster-management code that deals with IndexSegments
  - Share segments across serving machines using HDFS
  - Can rebuild segments (e.g. to upgrade Lucene version, change data schema, etc.)
Lucene Extensions: Storage

- Mesos
- Hadoop (MR)
- RT pipeline
- HDFS
- Earlybird
In-Memory Real Time Index

• RT term dictionary

• Term lookups using a lock-free hashtable in O(1)

• v2: Additional probabilistic, lock-free skip list maintains ordering on terms

• Perfect skip list not an option: out-of-order inserts would require rebalancing, which is impractical with our lock-free index

• In a probabilistic skip list the tower height of a new (out-of-order) item can be determined without knowing its insert position by simply rolling a dice
Revisit Old Idea: Probabilistic Skip Lists

- Probabilistic skip list

  Tower height determined by rolling a dice BEFORE knowing the insert location; tower height never has to change for an element, simplifying memory allocation and concurrency.
Searching for top entities within Tweets

- Query
- Term id
  - Inverted index
  - Doc ids
  - Term label

- Doc id
  - Forward index
  - Document Metadata

- Doc id
  - Facet index
  - Term ids
Top Entities (cont’d)

Top-k heap

<table>
<thead>
<tr>
<th>Id</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>48239</td>
<td>15</td>
</tr>
<tr>
<td>31241</td>
<td>12</td>
</tr>
<tr>
<td>85932</td>
<td>8</td>
</tr>
<tr>
<td>6748</td>
<td>3</td>
</tr>
</tbody>
</table>

Matching doc id

Facet index

Term ids
Resources

• CGC tutorial at WSDM 2012:
  http://www.mathcs.emory.edu/~eugene/wsdm2012-cgc-tutorial/

• Real-time Search at Twitter:
  https://www.youtube.com/watch?v=cbksU0qkFm0
  http://www.slideshare.net/lucidworks/search-at-twitter-presented-by-michael-busch-twitter