Knowledge Acquisition for Web Search

Marius Paşca

Google Inc.
mars@google.com

Overview

- Part One: Introduction
- Part Two: Acquisition of Open-Domain Knowledge
- Part Three: Role of Knowledge in Information Retrieval
Part One: Introduction

- Open-domain information extraction
- Instances, concepts, relations
- Impact on Web search

Unweaving the World Wide Web of Facts

- The Web is a repository of implicitly-encoded human knowledge
  - some text fragments contain easier-to-extract knowledge
- More knowledge leads to better answers
  - acquire facts from a fraction of the knowledge on the Web
  - exploit available facts during search
- Open-domain information extraction
  - extract knowledge (facts, relations) applicable to a wide range,
    rather than closed, pre-defined set of domains (e.g., medical,
    financial etc.)
  - no need to specify set of concepts and relations of interest in
    advance
  - rely on as little manually-created input data as possible
Instances, Concepts and Relations

- A concept (class) is a placeholder for a set of instances (objects) that share similar properties
  - set of instances
  - (matrix, kill bill, ice age, pulp fiction, inception, cidade de deus,...)
  - class label
  - movies, films
  - definition
  - a series of pictures projected on a screen in rapid succession with objects shown in successive positions slightly changed so as to produce the optical effect of a continuous picture in which the objects move (Merriam Webster)
  - a form of entertainment that enacts a story by sound and a sequence of images giving the illusion of continuous movement (WordNet)

Instances, Concepts and Relations

- Relations are assertions linking two (binary relation) or more (n-ary relation) concepts or instances
  - actors-act in-movies; cities-capital of-countries
- Facts are instantiations of relations, linking two or more instances
  - leonardo dicaprio-act in-inception; cairo-capital of-egypt
- Attributes correspond to facts capturing quantifiable properties of a concept or an instance
  - actors --> awards, birth date, height
  - movies --> producer, release date, budget

- Terminology
  - concept vs. class: used interchangeably
  - instance vs. entity: used interchangeably
Open-Domain Knowledge

- **Diseases**
  - yellow fever, influenza, bipolar disorder, rocky mountain spotted fever, anosmia, myxedema...
  - used in the treatment of
  - can reduce risk of
  - is a form of

- **Chemical Elements**
  - potassium, magnesium, gold, sulfur, palladium, argon, carbon, boron, ruthenium, zinc, lead...
  - decay product of
  - depletes the body of
  - brand name of

- **Foods**
  - fish, turkey, rice, milk, chicken, cheese, eggs, corn, beans, wheat, asparagus, grapes...
  - good sources of

- **Currencies**
  - euro, won, line, pounds, rand, us dollars, yen, pesos, pesetas, kroner, escudos, shillings...
  - worth millions of
  - currency of

- **Countries**
  - australia, south korea, kenya, greece, sudan, portugal, argentina, mexico, cuba, kuwait...
  - brand name of

- **Drugs**
  - paxil, lipitor, ibuprofen, prednisone, albuterol, effexor, azithromycin, fluconazole, advil...

Usefulness in Information Retrieval

- Open-domain knowledge resources
  - Search
Part Two: Acquisition of Knowledge

- Human-compiled knowledge resources
  - created by experts:
  - created collaboratively by non-experts:
    - [BEP+08]: K. Bollacker, C. Evans, P. Paritosh et al. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. SIGMOD-08.
Quantitative Comparison of Human-Compiled Resources

• Wikipedia
  - 5+ million articles in English
  - articles also available in 250+ other languages
• DBpedia
  - 4+ million instances in English, 250+ million relations
• Freebase
  - 40+ million instances, 3+ billion relations
• Cyc
  - ResearchCyc: 300,000+ concepts and 3+ million assertions
  - OpenCyc 2.0: add mappings from Cyc concepts to Wikipedia articles
• Open Mind
  - 800,000+ facts in English
  - facts also available in other languages
• ConceptNet
  - 1.5+ million assertions in multiple languages

Extraction of Open-Domain Knowledge

Methods for extraction of:
• concepts and instances as:
  - flat sets of unlabeled instances
  - flat sets of labeled instances, associating instances with class labels
  - conceptual hierarchies
• relations and attributes over:
  - flat concepts
  - conceptual hierarchies
Instances Within Unlabeled Concepts

- Instances: yellow fever, influenza, bipolar disorder, rocky mountain spotted fever, anosmia, myxedema,
  potassium, magnesium, gold, sulfur, palladium, argon, carbon, boron, ruthenium, zinc, lead,
  euro, won, line, pounds, rand, us dollars, yen, pesos, pesetas, kroner, escudos, shillings,
  paxil, lipitor, ibuprofen, prednisone, albuterol, effexor, azithromycin, fluconazole, advil,
  australia, south korea, kenya, greece, sudan, portugal, argentina, mexico, cuba, kuwait.

- Extract clusters of distributionally similar words from text documents:
  • [LP02]: D. Lin and P. Pantel. Concept Discovery from Text. COLING-02.
  • [WC08]: R. Wang and W. Cohen. Iterative Set Expansion of Named Entities using the Web. ICDM-08.
  • [LW09]: D. Lin and X. Wu. Phrase Clustering for Discriminative Learning. ACL-IJCNLP-09.
  • [HX11]: Y. He and D. Xin. Seisa: Set Expansion by Iterative Similarity Aggregation. WWW-11.
Instances Within Unlabeled Concepts


Extraction from Queries

- Data sources
  - anonymized search queries along with frequencies and click-through data (clicked search results)
  - Web documents
- Output
  - clusters of similar instances
    - e.g., (basic algebra, numerical analysis, discrete math, lattice theory, nonlinear physics, ...), (aaa insurance, roadside assistance, personal liability insurance, international driving permits, ...)
- Steps
  - collect set of candidate instances from queries
  - cluster instances using context in queries or click-through data or both
Similarity in Documents vs. Queries

- Contextual space of Web documents
  - an instance is represented by the contexts in which it appears in text documents
  - instances are modeled "objectively", according to descriptions of the world

- Contextual space of Web search queries
  - an instance is represented by the contexts in which it appears in a search queries
  - instances are modeled "subjectively", according to users' perception of the world

Extraction of Instances

- Identify candidate instances
  - intuition: in queries composed by copying fragments from Web documents and pasting them into queries, capitalization of instances is preserved
  - from queries containing capitalization, extract contiguous sequences of capitalized tokens as instances

- Retain set of best candidate instances
  - first criterion: promote candidate instances whose capitalization is frequent in Web documents
  - second criterion: promote candidate instances that occur as full-length queries

\[ r_w(E) = \frac{|\gamma(E)|}{\sum_{i \in O(E)} |\gamma(i)|} \]
\[ s_q(E) = \frac{|Q = E|}{|\text{queries that contain } E|} \]

- retain set of candidate instances that score highly (above some thresholds) according to both criteria

(Courtesy A. Jain)
Clustering of Instances

- Induce unlabeled classes of instances, by clustering instances using features collected from queries
  - as an alternative to collecting features from unstructured text in documents
  - for efficiency, no attempt to parse the queries
- Context features
  - vector of elements corresponding to contexts, where a context is the prefix and postfix around the instance, from queries containing the instance
- Click-through features
  - vector of elements corresponding to documents, where a document is one that is clicked by a user submitting the instance as a full-length query
- Hybrid features
  - normalized combination of context and click-through vectors

Impact of Clustering Features

- Given an instance, manually judge each co-clustered instance:
  - “If you were interested in instance I, would you also be interested in instance Ic in any intent?”
  - also, annotate with type of relation between instance and co-clustered instance

- Compute precision, over a set of evaluation instances
  - CL-CTX: context
  - CL-CLK: click-through
  - CL-HYB: hybrid
  - CL-Web: context collected from Web documents rather than queries

<table>
<thead>
<tr>
<th>Relation Type</th>
<th>Method</th>
<th>CL-Web</th>
<th>CL-CTX</th>
<th>CL-CLK</th>
<th>CL-HYB</th>
</tr>
</thead>
<tbody>
<tr>
<td>topic</td>
<td></td>
<td>0.27</td>
<td>0.46</td>
<td>0.46</td>
<td>0.40</td>
</tr>
<tr>
<td>sibling</td>
<td></td>
<td>0.72</td>
<td>0.43</td>
<td>0.29</td>
<td>0.32</td>
</tr>
<tr>
<td>parent</td>
<td></td>
<td>-</td>
<td>0.09</td>
<td>0.13</td>
<td>0.09</td>
</tr>
<tr>
<td>child</td>
<td></td>
<td>0.01</td>
<td>-</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>synonym</td>
<td></td>
<td>0.01</td>
<td>0.03</td>
<td>0.12</td>
<td>0.16</td>
</tr>
</tbody>
</table>
Methods for extraction of:
- concepts and instances as:
  - flat sets of unlabeled instances
  - flat sets of labeled instances, associating instances with class labels
  - conceptual hierarchies
- relations and attributes over:
  - flat concepts
  - conceptual hierarchies

Instances Within Labeled Concepts

- **diseases**
  - yellow fever, influenza, bipolar disorder, rocky mountain spotted fever, anosmia, myxedema...
- **chemical elements**
  - potassium, magnesium, gold, sulfur, palladium, argon, carbon, borium, ruthenium, zinc, lead...
- **drugs**
  - paxil, lipitor, ibuprofen, prednisone, albuterol, effexor, azithromycin, fluconazole, advil...
- **foods**
  - fish, turkey, rice, milk, chicken, cheese, eggs, corn, beans, wheat, asparagus, grapes...
- **currencies**
  - euro, won, lira, pounds, rand, us dollars, yen, pesos, pesetas, kroner, escudos, shillings...
- **countries**
  - australia, south korea, kenya, greece, sudan, portugal, argentina, mexico, cuba, kuwait...
Instances Within Labeled Concepts

• [Hea92]: M. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. COLING-92.
  - extract IsA pairs (i.e., pairs of an instance and a class label) from text documents using a set of
    lexico-syntactic patterns
• [RJ99]: E. Riloff and R. Jones. Learning Dictionaries for Information Extraction by Multi-
  - expand set of IsA pairs by iteratively identifying extraction patterns and conservatively growing the
    set of IsA pairs by a small number of new IsA pairs
• [RP04]: P. Pantel and D. Ravichandran. Automatically Labeling Semantic Classes. HLT-
  NAACL-04.
  - assign class labels to pre-extracted sets of instances
• [SJN05]: R. Snow, D. Jurafsky and A. Ng. Learning syntactic patterns for automatic
  hypernym discovery. NIPS-05.
  - learn extraction patterns for extracting IsA pairs from text documents
• [ECD+05]: O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld
  and A. Yates. Unsupervised Named-Entity Extraction from the Web: an Experimental Study.
  Journal of Artificial Intelligence Research 2005.
  - instantiate generic rule templates to extract instances within various concepts via search engines
  - expand set of IsA pairs from text documents, by exploiting pairs extracted for other classes as
    negative examples to improve the quality of the induced patterns and extracted IsA pairs

Instances Within Labeled Concepts

• [KRH08]: Z. Kozareva, E. Riloff and E. Hovy. Semantic Class Learning from the Web with
  Hyponym Pattern Linkage Graphs. ACL-08.
  - expand set of instances and associated class label (also given as input) from Web documents via
    search engines
• [YTK+09]: I. Yamada, K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. De Soegi, P. Bond
  and A. Sumida. Hypernym Discovery Based on Distributional Similarity and Hierarchical
  Structures. ACL-IJCNLP-09.
  - extract IsA pairs from Web documents, by using lexico-syntactic patterns and distributional
    similarities, and attach extracted pairs to Wikipedia categories
• [WCO09]: R. Wang and W. Cohen. Automatic Set Instance Extraction using the Web. ACL-
  IJCNLP-09.
  - extract instances of given class labels, from Web documents via search engines
• [TP10]: P. Talukdar and F. Pereira. Experiments in Graph-Based Semi-Supervised Learning
  Methods for Class-Instance Acquisition. ACL-10.
  - extract IsA pairs from manually-created or automatically-extracted repositories, via graph
    propagation, by incorporating structured data derived from Wikipedia
• [SHL10]: S. Singh, D. Hillard and C. Leggetter. Minimally-Supervised Extraction of Entities
  from Text Advertisements. ACL-10.
  - extract instances within around 30 class labels, from corpus of Web sponsored ads
Instances Within Labeled Concepts

  - extract ISA pairs from Web documents, by using lexico-syntactic patterns then propagating class labels among similar instances
  - extract labeled sets of instances from Web documents, by using lexico-syntactic patterns and clusters of instances from Web tables
  - given a class label, extract set of instances of the class from Web documents
  - given a class label and a small set of seed instances, extract larger set of instances of the class from Web tables
Extracting Instances from Documents

Input
- class label for which instances should be extracted (e.g., hiking trains near Baltimore)

Data source
- Web documents accessed via general-purpose Web search engine

Output
- instances of the class label (e.g., Avalon Super Loop)

Steps
- construct extraction predicates based on the HTML representation of documents, targeting table columns, lists, headers at the same level
- apply extraction patterns to Web documents relevant to the input class label
- extract instances of the class label

Extraction Predicates

\[ z = /html[1]/body[1]/table[2]/tr/td[1] \]

(Courtesy P. Pasupat)
Extraction Model

hiking trails near Baltimore

\(|Z| \approx 8500\)

/\text{html[1]/body[1]/table[2]/tr/td[1]}\]

[Avalon Super Loop, Patapsco Valley State Park, ...]

Extraction from Web Documents
Extraction Features

George Washington
John Adams
Thomas Jefferson
James Madison
... (39 more) ...
Barack Obama
John Adams
John Adams
John Adams
John Adams
John Adams
John Adams
... (100 more) ...
John Adams

Desired Classification (good) (bad) (bad)
Feature: Identity (diverse) (identical) (diverse)

Extraction Features

NNP NNP
NNP NNP
NNP NNP
NNP NNP
NNP NNP
... (39 more) ...
NNP NNP
NNP NNP
NNP NNP
NNP NNP
NNP NNP
... (100 more) ...
NNP NNP

Desired Classification (good) (bad) (bad)
Feature: Identity (diverse) (identical) (diverse)
Feature: Parts of speech (identical) (identical) (diverse)
Extracted Instances

Query: disney channel movies

Extracted Instances

Accuracy

<table>
<thead>
<tr>
<th>Baseline (Most frequent extraction predicates)</th>
<th>Accuracy</th>
<th>Accuracy @ 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.3</td>
<td>40.5</td>
<td>55.8</td>
</tr>
</tbody>
</table>
Methods for extraction of:

- concepts and instances as:
  - flat sets of unlabeled instances
  - flat sets of labeled instances, associating instances with class labels
  - conceptual hierarchies

- relations and attributes over:
  - flat concepts
  - conceptual hierarchies

---

Conceptual Hierarchies

- [Wid03]: D. Widdows. Unsupervised Methods for Developing Taxonomies by Combining Syntactic and Statistical Information. HLT-NAACL-03.
  - insert new phrases into an existing hierarchy
- [SJN06]: R. Snow, D. Jurafsky and A. Ng. Semantic Taxonomy Induction from Heterogeneous Evidence. ACL-06.
  - extend WordNet with ISA pairs extracted from text
- [PS07]: S. Ponzetto and M. Strube. Deriving a Large Scale Taxonomy from Wikipedia. AAAI-07.
  - apply filters to network of Wikipedia categories to extract hierarchy of categories
  - incrementally cluster set of phrases into a hierarchy, using co-occurrence, syntactic dependencies and lexico-syntactic patterns
- [PN09]: S. Ponzetto and R. Navigli. Large-Scale Taxonomy Mapping for Restructuring and Integrating Wikipedia. IJCAI-09.
  - map Wikipedia categories to WordNet synsets, and use mappings to restructure the hierarchy generated in [PS07]
  - organize concepts extracted from Web documents via search engines, into hierarchies created from scratch
Conceptual Hierarchies

  - from network of Wikipedia articles and Wikipedia categories, extract hierarchy of articles and categories
  - automatically generate informative questions to ask users, such that their answers serve in incrementally constructing and refining conceptual hierarchies
Constructing Hierarchies from Text

• Data source
  - category network in Wikipedia, containing edges from articles to other articles, from articles to their parent categories, and from categories to their own parent categories

• Output
  - an hierarchy containing IsA edges between Wikipedia articles, and another hierarchy containing IsA edges between Wikipedia categories

• Steps
  - analyze the text of each article to select a candidate hypernym phrase for the article (e.g., character for Mickey Mouse)
  - disambiguate the hypernym phrase into another article corresponding to the desired sense of the hypernym phrase (e.g., character into Character (arts))
  - organize pairs of an article and its disambiguated hypernym article into hierarchy of articles
  - starting from the hierarchy of articles (e.g., Real Madrid IsA Football club), exploit Wikipedia edges between articles and categories to iteratively infer IsA edges between categories of articles (e.g., Football clubs in Madrid IsA Football clubs), and then IsA edges between articles of categories (e.g., Atlético Madrid IsA Football club)
  - expand the hierarchy of categories to increase its coverage

Edges in Category Network

(Courtesy T. Flati)
Constructing Hierarchy of Articles

- Select main (first) sentence in article

Scrooge McDuck

- From the dependency parse of main sentence, select candidate hypernym phrases

Scrooge McDuck is a character [...] → Scrooge McDuck is a character [...] → Hypernym phrase: character

- Using heuristics, disambiguate the hypernym phrase into another article corresponding to the desired sense
  - heuristics rely on links among articles, common categories, context around links
  - Hypernym phrase: character → Character (arts)

- Retain pairs of an article and its hypernym article, as IsA edges

Scrooge McDuck → Character (arts)

Constructing Hierarchy of Categories

Start from the hierarchy of articles
Constructing Hierarchy of Categories

Exploit the article to category edges to infer hypernym relations in the category hierarchy.

Iteratively Refining the Hierarchies

Traverse article to category edges to infer back is-a relations in the hierarchy of articles.
Iteratively Refining the Hierarchies

Iteratively use the relations found in previous step to infer new hyponym edges.

Expanding the Hierarchy of Categories

- Expand for categories with a single parent category in the category network.

Retain the parent category as being a hyponym category.
Expanding the Hierarchy of Categories

- Infer hypernym categories of a category, from hypernym categories already extracted for child categories in the category network.
Methods for extraction of:

- concepts and instances as:
  - flat sets of unlabeled instances
  - flat sets of labeled instances, associating instances with class labels
  - conceptual hierarchies

- relations and attributes over:
  - flat concepts
  - conceptual hierarchies
Attributes and Relations

- **diseases**: yellow fever, influenza, bipolar disorder, rocky mountain spotted fever, anosmia, myxedema...
- **chemical elements**: potassium, magnesium, gold, sulfur, palladium, argon, carbon, boron, ruthenium, zinc, lead...
- **foods**: fish, turkey, rice, milk, chicken, cheese, eggs, corn, beans, wheat, asparagus, grapes...
- **currencies**: euro, won, line, pounds, rand, us dollars, yen, pesos, pesetas, kroner, escudos, shillings...
- **countries**: australia, south korea, kenya, greece, sudan, portugal, argentina, mexico, cuba, kuwait...

**Drugs**: paxil, lipitor, ibuprofen, prednisone, albuterol, effexor, azithromycin, fluconazole, advil...

**Uses in the treatment of**

- **used in the treatment of**
- **can reduce risk of**
- **is a form of**
- **depletes the body of**

**Relations over Flat Concepts**

- [AP04]: A. Almuhareb and M. Poesio. Attribute-Based and Value-Based Clustering: an Evaluation. EMNLP-04.
  - examine the role of attributes vs. values in acquiring concept descriptions via search engines
- [PE05]: A. Popescu and O. Etzioni. Extracting Product Features and Opinions from Reviews. EMNLP-05.
  - use product features (including attributes) found in text, to extract and rank opinions about products
- [TKT05]: K. Tokunaga, J. Kazama and K. Torisawa. Automatic Discovery of Attribute Words from Web Documents. IJCNLP-05.
  - apply small set of patterns to extract attributes from unstructured text in a small Web collection
  - extract open-ended facts, without specifying the concepts or relations of interest in advance
- [SS06]: Y. Shinyama and S. Sekine: Preemptive Information Extraction using Unrestricted Relation Discovery. HLT-NAACL-06.
  - extract clusters of relations from parsed text, without specifying relations of interest in advance
- [PP06]: P. Pantel and M. Pennacchiotti. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. ACL-06.
  - expand seed set of relations from text documents via iteratively induced extraction patterns
- [WW07]: F. Wu and D. Weld. Autonomously Semantifying Wikipedia. CIKM-07.
  - extend Wikipedia infoboxes with attributes and values inferred from text
  - extract attributes and associated values from semi-structured text via search engines
Relations over Flat Concepts

- [BM07]: R. Bunescu and R. Mooney. Learning to Extract Relations from the Web using Minimal Supervision. ACL-07.
  - exploit small sets of positive and negative seeds, to extract relations from text via search engines
  - extract attributes and associated values of products
  - extract relations in a single pass over collection of Web documents, without any manual input
- [DRK07]: D. Davidov, A. Rappoport and M. Koppel. Fully Unsupervised Discovery of Concept-Specific Relationships by Web Mining. ACL-07.
  - extract relevant relations for given concepts, from unstructured text via search engines
- [VQS08]: B. Van Durme, T. Qian and L. Schubert. Class-Driven Attribute Extraction. COLING-08.
  - extract attributes via more complex representations of parsed text
  - identify and exploit high-quality relational data available in tables

- [BE08]: M. Banko and O. Etzioni. The Tradeoffs Between Open and Traditional Relation Extraction. ACL-08.
  - investigate the mapping of open-ended relations into relation-independent lexico-syntactic patterns
  - extract attributes of products from semi-structured text within Web documents
- [NS08]: V. Nastase and M. Strube. Decoding Wikipedia Categories for Knowledge Acquisition. AAAI-08.
  - from categories and category network, derive relations among categories or instances, including attributes of categories
  - using tuples already available for the relation in Freebase, extract additional relations from unstructured text
  - identify relevant relations for Wikipedia categories, from parsed Wikipedia articles and from Web documents via search engines
- [LWA09]: X. Li, Y. Wang and A. Acero. Extracting Structured Information from User Queries with Semi-Supervised Conditional Random Fields. SIGIR-09.
  - detect relevant fields in product search queries, using click data and document content
  - expand seed sets provided for each target concept and relation, enhancing extractions of individual concepts/relations using extractions for other concepts/relations
Relations over Flat Concepts

  - loosely identify the relation between a query and a given instance, in the form of explanatory
    sentences collected from Wikipedia articles
  - improve quality of individually extracted facts, by global analysis of common arguments (instances)
    shared among the facts
• [DR10]: D. Davidov and A. Rappoport. Extraction and Approximation of Numerical Attributes
  from the Web. ACL-10.
  - given an instance and an attribute whose value is numerical, extract the value from Web documents
    via search engines
• [KH10b]: Z. Kozareva and E. Hovy. Learning Arguments and Supertypes of Semantic
  Relations using Recursive Patterns. ACL-10.
  - given an extraction pattern expressing a relation, and a seed instance for one argument of the
    relation, infer additional pairs of arguments for the same relation as well as the types of those
    arguments, from Web documents via search engines
• [LME10]: T. Lin, Massam and O. Etzioni. Identifying Functional Relations in Web Text.
  EMNLP-10.
  - given relations extracted from Web documents, identify relations that connect the first argument to
    a unique value
• [SEW-10]: S. Schoenmackers, O. Etzioni, D. Weld and J. Davis. Learning First-Order Horn
  Clauses from Web Text. EMNLP-10.
  - acquire inference rules and apply them to expand a set of relations extracted from Web documents
• [BM10]: D. Bollegala, Y. Matsuo and M. Ishizuka. Relation Duplity: Unsupervised Extraction
  of Semantic Relations between Entities on the Web. WWW-10.
  - given relations through combination of patterns expressing the type, and phrase pairs expressing
    the arguments
• [YTT10]: X. Yin, W. Tan and Y. Tu. Automatic Extraction of Clickable Structured Web
  Contents for Name Entity Queries. WWW-10.
  - given a query containing an instance, extract structured data from click data and contents of
    subsequently visited documents
• [WW10]: F. Wu and D. Weld. Open Information Extraction Using Wikipedia. ACL-10.
  - from unstructured text, extract relations whose types are derived from Wikipedia
• [FSE11]: A. Fader, S. Soderland and O. Etzioni. Identifying Relations for Open Information
  Extraction. EMNLP-11.
  - enforce lexical and syntactic constraints on relations extracted from text, to improve their quality
• [DG13]: L. Del Corro and R. Gemulla. ClausIE: Clause-Based Open Information Extraction.
  WWW-13.
  - apply a small set of general-purpose patterns to parse trees over unstructured text, to extract
    higher-precision relations
• [WGM-14]: R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta and D. Lin. Knowledge Base
  Completion via Search-Based Question Answering. WWW-14.
  - extract missing values of attributes of instances within an existing knowledge repository, from Web
    search result snippets returned for automatically-generated questions
• [TMW14]: N. Tandon, G. de Melo and G. Weikum. Acquiring Comparative Commonsense
  Knowledge from the Web. AAAI-14.
  - from unstructured text, extract relations among disambiguated instances, where the relations
    compare the respective pairs of arguments along relevant dimensions
• [DGK-14]: X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao and K. Murphy. Knowledge Vault:
  A Web-Scale Approach to Probabilistic Knowledge Fusion. KDD-14.
  - create a knowledge repository, based on relations extracted from Web documents and knowledge
    from available repositories
Relations over Flat Concepts

• [DMS15]: A. Dutta, C. Meilicke and H. Stuckenschmidt. Enriching Structured Knowledge with Open Information. WWW-15
  - given relations extracted from Web documents, convert arguments and relations from ambiguous strings into disambiguated entries from a knowledge repository

  - infer missing relations based on relations already available in a knowledge repository

  - reduce document sentences deemed relevant to shorter clauses, then apply small set of patterns to clauses to extract relations

From Relations to Explanatory Sentences

- Data source
  - collection of Web documents
- Output
  - facts capturing comparative knowledge, e.g., <snow, less dense, rain>
- Steps
  - apply a small set of extraction patterns to Web documents to extract comparative facts
  - disambiguate arguments and predicate within the extracted facts relative to WordNet
  - organize into groups of equivalent facts

Acquisition from Web Documents
(Courtesy N. Tandon)

<table>
<thead>
<tr>
<th>Argument1</th>
<th>Relation/Adjective</th>
<th>Argument2</th>
</tr>
</thead>
<tbody>
<tr>
<td>snow-n-2</td>
<td>less dense-a-3</td>
<td>rain-n-2</td>
</tr>
<tr>
<td>marijuana-n-2</td>
<td>more dangerous-a-1</td>
<td>alcohol-n-1</td>
</tr>
<tr>
<td>little child-n-1</td>
<td>happier (happy-a-1)</td>
<td>adult-n-1</td>
</tr>
<tr>
<td>private school-n-1</td>
<td>more expensive-a-1</td>
<td>public institute-n-1</td>
</tr>
<tr>
<td>peaceful resistance-n-1</td>
<td>more effective-a-1</td>
<td>violent resistance-n-1</td>
</tr>
<tr>
<td>wet wood-n-1</td>
<td>softer (soft-a-1)</td>
<td>dry wood-n-1</td>
</tr>
</tbody>
</table>
Next Topic

Methods for extraction of:
• concepts and instances as:
  - flat sets of unlabeled instances
  - flat sets of labeled instances, associating instances with class labels
  - conceptual hierarchies
• relations and attributes over:
  - flat concepts
  - conceptual hierarchies

Relations over Conceptual Hierarchies

• [PP06b]: M. Pennacchiotti and P. Pantel. Ontologizing Semantic Relations. ACL-06.
  - attach relations extracted from text to WordNet hierarchies, by identifying the WordNet concepts to which the arguments of the relation correspond
  - map Wikipedia categories to WordNet to generate hybrid resource of concepts and relations
• [WW08]: F. Wu and D. Weld. Automatically Refining the Wikipedia Infobox Ontology. WWW-08.
  - extend Wikipedia infoboxes with additional attributes and values, by mapping templates of Wikipedia infoboxes to WordNet
  - extend existing repositories of relations like Wikipedia, with facts acquired from unstructured text
  - extract relations from unstructured text within Wikipedia articles, via dynamic lexicons acquired from semi-structured text within Web documents
  - link Wikipedia articles to WordNet concepts and apply machine translation, to create a multi-lingual repository of relations
  - given hierarchically-organized concepts associated with their sets of instances, extract relations among the concepts from unstructured text
Relations over Conceptual Hierarchies

  - combine WordNet, Wikipedia and other sources into hierarchically organized instances and their relations, where the data is anchored in time and space
  - from Web documents, extract textual descriptions of relations between entries in pairs of entries from a knowledge repository
  - from Web documents, extract hierarchy of concepts and relation types, and also extract relations filling in the hierarchy

Relations over Conceptual Hierarchies

From Relations to Explanatory Sentences

- **Input**
  - pairs of instances connected by a relation, as available in a knowledge repository (e.g., the pair Brad Pitt, Seven (1995 movie))
- **Data source**
  - Wikipedia articles corresponding to instances from the knowledge repository
- **Output**
  - ranked list of sentences extracted from documents, which explain the relations within the pairs of instances (e.g., "Brad Pitt gave critically acclaimed performances in the crime thriller Seven")
- **Steps**
  - extract surface forms for each instance in a pair of instances (e.g., Brad Pitt, Brad, Pitt)
  - extract candidate sentences from Wikipedia articles, using both surface forms and instances to which surface forms are disambiguated
  - rank candidate sentences using a variety of features

Ranking Candidate Sentences

- **Textual features**
  - length of candidate sentence
  - fractions of sentence tokens that are verbs vs. nouns vs. adjectives
  - ...
- **Instance features**
  - count of instances in candidate sentence
  - distance in tokens between last match of the two input instances in the candidate sentence
  - ...
- **Relation features**
  - whether candidate sentence contains tokens of the input relation
  - ...
- **Document features**
  - position of candidate sentence in source document
  - whether the source document of the candidate sentence is the article of one of the two input instances
  - ...
Accuracy of Explanatory Sentences

Normalized discounted cumulative gain

Prior document retrieval method developed specifically for sentence retrieval

Proposed ranking method with one combined ranking model for all relation types

Proposed ranking method with separate ranking models for each relation type

At least one of the retrieved sentences contain a sentence of the indicated relevance bucket

(Courtesy N. Voskarides)

Next Topic

- Part One: Introduction
- Part Two: Acquisition of Open-Domain Knowledge
- Part Three: Role of Knowledge in Information Retrieval
Role of Knowledge in Search

- [Voo94]: E. Voorhees. Query Expansion Using Lexical-Semantic Relations. SIGIR-94.
  - investigate the impact of manual and automatic expansion of queries on search results, using lexico-semantic relations available in WordNet
- [LZZ+06]: M. Li, M. Zhu, Y. Zhang and M. Zhou. Exploring Distributional Similarity Based Models for Query Spelling Correction. ACL-06.
  - take advantage of a repository of distributionally similar phrases acquired from search queries, in order to suggest correctly spelled queries in response to misspelled queries
- [Fan08]: H. Fang. A Re-Examination of Query Expansion Using Lexical Resources. ACL-08.
  - propose an alternative term weighting scheme for query expansion using lexico-semantic relations available in WordNet
  - model query intent domains as areas in the Wikipedia category network situated around manually-provided seed articles in Wikipedia, and map queries into those domains
  - in response to fact-seeking queries, return facts identified in tuples of an instance, attribute and value extracted from tables within Web documents
  - return synthetic query suggestions in response to long-tail queries for which few or no query suggestions would be otherwise available
  - compare the impact of algorithms for disambiguating instances mentioned in a document relative to articles in Wikipedia, using evidence available locally for each mention vs. globally for all mentions
  - compute mappings from queries into instances from a structured database, for the purpose of identifying relevant products from a product catalog and recommending them in response to queries
  - extend search results with multimedia objects, for the purpose of improving search experience and aiding users in determining the relevance of search results
  - take advantage of mappings from instances mentioned in documents to Wikipedia articles, in order to cluster search results and their result snippets into sets associated with descriptive labels
  - in response to queries with how-to intent, return relevant tips extracted from a collaborative question-answering repository containing pairs of a question and an answer
  - improve the search results returned for poorly performing queries, by expanding the queries with concepts derived from a large knowledge repository
Role of Knowledge in Search

  - compute mappings from query fragments into instances from an existing knowledge repository, to better identify patterns of Web usage
  - extract the most salient instances mentioned in Web documents
  - in response to fact-seeking questions, extract answers from unstructured text from Web documents and from relations available in a knowledge repository
  - compute mappings from query fragments into instances from an existing knowledge repository, to expand queries for better search results
  - given a query, compute and recommend related entries from a knowledge repository
  - compute mappings from query fragments into instances from an existing knowledge repository, under strong latency constraints
  - extract and convert mentions of local events within Web documents into structured, searchable calendar entries

Role of Knowledge in Search

• Document analysis and understanding
  - mapping of document terms into concepts [RRD+11]
  - clustering of search results [SMF+12]
  - extraction of salient instances [GYS+13]
• Query analysis and understanding
  - understanding intent, query categorization [HWL+09]
  - Web usage analysis [HMB13]
  - mapping of queries into concepts [BOM15], product recommendation [PF11]
  - query suggestion [JOV11], recommendation of related queries [BMH+15]
  - spell checking [LZZ+06]
• Matching of queries onto documents
  - query expansion [Voo94, Fan08, KZ12, DAD14]
• Onebox search results
  - retrieval of answers for queries with how-to intent [WUG12]
  - retrieval of answers for fact-seeking queries [YTL11, YV14]
  - retrieval of multimedia objects [HMT+11]
• ...
•...
Document Understanding


---

Disambiguation to Wikipedia

- Task
  - given a text fragment containing mentions (substrings) to be disambiguated, "wikifi" the mentions by identifying the Wikipedia article, if any, corresponding to each mention
  - mapping from mentions to Wikipedia articles relies on evidence available in the text fragment
- Scope of available evidence
  - local: separately available for each mention in the text fragment
  - global: collectively available for all mentions in the text fragment
- Goal
  - investigate impact of local vs. global evidence on accuracy of disambiguation
Mapping Mentions to Wikipedia

It's a version of Chicago – the standard classic Macintosh menu font, with that distinctive thick diagonal in the “N”.

Chicago was used by default for Mac menus through MacOS 7.6, and OS 8 was released mid-1997.

Chicago VIII was one of the early 70s-era albums to catch my ear, along with Chicago II.

(Courtesy L. Ratinov)

Disambiguation Strategy

Algorithm: Disambiguate to Wikipedia

Input: document d, Mentions M = \{m_1, \ldots, m_N\}
Output: a disambiguation \( \Gamma = (t_1, \ldots, t_N) \).

1) Let \( M' = M \cup \{\text{Other potential mentions in } d\} \)
2) For each mention \( m'_i \in M' \), construct a set of disambiguation candidates \( T_i = \{t'_i, \ldots, t'_{i(M')}\} \), \( t'_i \neq \text{null} \)
3) Ranker: Find a solution \( \Gamma = (t'_1, \ldots, t'_{M'}) \), where \( t'_i \in T_i \) is the best non-null disambiguation of \( m'_i \).
4) Linker: For each \( m'_i \), map \( t'_i \) to null in \( \Gamma \) iff doing so improves the objective function
5) Return \( \Gamma \) entries for the original mentions \( M \).

- Two stages
  - ranker: compute best Wikipedia article that potentially disambiguates the mention
  - linker: determine whether the mention should be mapped to the Wikipedia article or should not be mapped to any article
## Ranker: Local vs. Global Disambiguation

Accuracy: fraction of mentions for which ranker identifies correct disambiguation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>Baseline+ Lexical</th>
<th>Baseline+ Global Unambiguous</th>
<th>Baseline+ Global NER</th>
<th>Baseline+ Global, All Mentions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACE</td>
<td>94.05</td>
<td>94.56</td>
<td>96.21</td>
<td>96.75</td>
<td></td>
</tr>
<tr>
<td>MSNBC News</td>
<td>81.91</td>
<td>84.46</td>
<td>84.04</td>
<td>88.51</td>
<td></td>
</tr>
<tr>
<td>AQUAINT</td>
<td>93.19</td>
<td>95.40</td>
<td>94.04</td>
<td>95.91</td>
<td></td>
</tr>
<tr>
<td>Wikipedia Test</td>
<td>85.88</td>
<td>89.67</td>
<td>89.59</td>
<td>89.79</td>
<td></td>
</tr>
</tbody>
</table>

Previous methods

Over test set of Wikipedia documents, local performs better than global
Overall: Local vs. Global Evidence

Combined precision and recall (F1 score)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>Baseline+ Lexical</th>
<th>Baseline+ Lexical+ Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACE</td>
<td>94.05</td>
<td>96.21</td>
<td>97.83</td>
</tr>
<tr>
<td>MSNBC News</td>
<td>81.91</td>
<td>85.10</td>
<td>87.02</td>
</tr>
<tr>
<td>AQUAINT</td>
<td>93.19</td>
<td>95.57</td>
<td>94.38</td>
</tr>
<tr>
<td>Wikipedia Test</td>
<td>85.88</td>
<td>93.59</td>
<td>94.18</td>
</tr>
</tbody>
</table>

(Comparing set of Wikipedia articles output by algorithm for a document, with gold set of Wikipedia articles for the document)

(Result) Document Understanding

- [SMF+12]: U. Scaiella, A. Marino, P. Ferragina and M. Ciaramita. Topical Clustering of Search Results. WSDM-12.
Clustering of Search Results

- **Input**
  - search results and their snippets, returned in response to queries
- **Data source**
  - Wikipedia articles and categories, connected via the category network
- **Output**
  - decomposition of search results into topically coherent subsets associated with labels derived from Wikipedia
  - on the fly, without analysis of full content of search results
- **Steps**
  - annotate snippets with corresponding Wikipedia articles ("topics")
  - analyze graph of snippets and topics, to determine most significant topics
  - partition graph around most significant topics, and cut into ~10 clusters
  - for each cluster, select centroid topic as label for the entire cluster

Annotation of Snippets with Topics

(Courtesy P. Ferragina)
Selection of Significant Topics

- Exploit weights assigned during annotation to mappings from snippets to topics
- Process topic nodes iteratively, in order of the sum of weights of connecting edges
- As topic nodes are marked as significant, ignore corresponding snippet nodes and their outgoing mappings in subsequent iterations

Snippets Topics

Jaguar International - Market selector page
Official worldwide web site of Jaguar Cars ...

Jaguar India - Jaguar Cars - Home
Jaguar - XF - XK. The information and specifications shown on this website may vary and are subject to change ...

Jaguar Facts - Defenders of Wildlife - Defenders ...
Jaguars are listed as endangered under Endangered Species Act ...

Jaguars, Jaguar Pictures, Jaguar Facts - National Geographic
Learn all you wanted to know about jaguars with pictures, videos, photos, facts, and news from National Geographic ...

National Center for Computational Sciences - Jaguar
The Jaguar system consists of an 84-cabinet quad-core Cray X14 system and 200 upgraded Cray XTS cabinets ...

Apple - Support - Mac OS X 10.2 Jaguar and earlier
The Mac OS X 10.2 (Jaguar) and earlier Support Page helps with troubleshooting, tutorials, service, and information for new users.

Jaguar Technologies LLC
JaguarRC is the original leader in VPS Hosting, providing the best web hosting services ...
Partition into Clusters

Snippets
- Jaguar International - Market selector page
  Official worldwide web site of Jaguar Cars ...
- Jaguar India - Jaguar Cars - Home
  Jaguar - XF - XK. The information and specifications shown on this website may vary and are subject to change ...
- Jaguar Facts - Defenders of Wildlife - Defenders
  Jaguars are listed as endangered under Endangered Species Act ...
- Jaguar Pictures, Jaguar Facts - National Geographic
  Learn all you wanted to know about Jaguars with pictures, videos, facts, and news from National Geographic ...
- National Center for Computational Sciences - Jaguar
  The Jaguar system consists of an 84-cabinet quad-core Sun Fire X44 system and 200 upgraded Sun Fire X54 cabinets ...
- Apple - Support - Mac OS X 10.2 Jaguar and earlier
  The Mac OS X 10.2 (Jaguar) and earlier Support Page helps with troubleshooting, tutorials, service, and information for new users ...
- Jaguar Technologies LLC
  JaguarTech is the original leader in VPS hosting, providing the best web-hosting services ...

Topics
- Cluster of topics
- Jaguar XK
- Jaguar XP
- Jaguar Cars
- Panthera Onca
- Endangered Species Act
- National Center for Computational Sciences
- Mac OS X
- Jaguar
- Web hosting
- Vendor

Selection of Cluster Labels

Snippets
- Jaguar International - Market selector page
  Official worldwide web site of Jaguar Cars ...
- Jaguar India - Jaguar Cars - Home
  Jaguar - XF - XK. The information and specifications shown on this website may vary and are subject to change ...
- Jaguar Facts - Defenders of Wildlife - Defenders
  Jaguars are listed as endangered under Endangered Species Act ...
- Jaguar Pictures, Jaguar Facts - National Geographic
  Learn all you wanted to know about Jaguars with pictures, videos, facts, and news from National Geographic ...
- National Center for Computational Sciences - Jaguar
  The Jaguar system consists of an 84-cabinet quad-core Sun Fire X44 system and 200 upgraded Sun Fire X54 cabinets ...
- Apple - Support - Mac OS X 10.2 Jaguar and earlier
  The Mac OS X 10.2 (Jaguar) and earlier Support Page helps with troubleshooting, tutorials, service, and information for new users ...
- Jaguar Technologies LLC
  JaguarTech is the original leader in VPS hosting, providing the best web-hosting services ...

Topics
- Label selected for entire cluster
- Jaguar XK
- Jaguar XP
- Jaguar Cars
- Panthera Onca
- Endangered Species Act
- National Center for Computational Sciences
- Mac OS X
- Jaguar
- Web hosting
- Vendor
Next Topic

- Document analysis and understanding
- Query analysis and understanding
- Matching of queries onto documents
- Onebox search results

Query Understanding

Modeling Query Intent with Wikipedia

- **Input**
  - queries
- **Data source**
  - Wikipedia articles and categories, connected via the category network
- **Output**
  - intent domains identified for queries, modeled as areas in the Wikipedia category network situated around manually-provided seed articles in Wikipedia
- **Steps**
  - independently from input queries, manually identify a small set of seed queries for each domain of interest
  - given set of seed queries, manually identify seed Wikipedia articles that correspond to the domain of interest
  - for each domain, expand seed Wikipedia articles into more Wikipedia articles, using connections between articles (article links, category network)
  - map queries into intent domains, taking into consideration manually-provided mappings from sets of seed queries

System Architecture

- Wikipedia
  - Labeled seed articles
  - Wikipedia link graph construction
  - Wikipedia articles associated with scores for each intent domain

- Query
  - Query can be mapped directly to some Wikipedia article?
  - Map query to Wikipedia articles
  - Set of Wikipedia articles corresponding to query terms
  - Query intent
Modeling of Intent Domains

• Construct link graph for Wikipedia articles
  - nodes: Wikipedia articles, Wikipedia categories
  - edges: links between articles, links in Wikipedia category network between articles and categories; edges added between two nodes only when bi-directional links exist between the two nodes
  - edge weights: counts of links between the two nodes

• Associate Wikipedia articles with score for each intent domain
  - manually select seed Wikipedia articles deemed to belong to intent domain
  - iteratively propagate intent from seed articles to their neighbors articles in the link graph, assigning gradually lower intent scores

Determining Query Intent

• Case 1: query can be mapped directly to a Wikipedia article
  - retrieve intent domain whose intent score associated with the Wikipedia article is highest

• Case 2: query cannot be mapped directly to a Wikipedia article
  - map query into its more related Wikipedia articles, by disambiguating (“wikifying”) mentions (substrings) from query to corresponding Wikipedia articles
  - retrieve intent domain for which the combination of intent scores, associated with the related Wikipedia articles, is highest
Query Understanding


Mapping Queries to Structured Entities

- **Input**
  - queries
  - click data for search results returned in response to queries
  - click data for structured instances returned in response to queries

- **Data source**
  - collection of instances available within a structured knowledge repository (e.g., Freebase, IMDB, product catalog)

- **Output**
  - list of instances from knowledge repository deemed relevant to the query
  - similar to query suggestion, but suggestions are instances not strings

- **Steps**
  - create click graph connecting queries, instances and clicked documents
  - exploit edges between queries and clicked documents, and similarity edges between queries capturing the overlap of their sets of clicked documents, to extend graph with new edges between queries and instances
  - transfer weights from existing edges to newly added edges
  - apply resulting graph to suggest relevant instances for each query
Construction of Click Graph

(Courtesy P. Pantel)

Observed clicks on instances from knowledge repository

Edges capturing query similarity computed via clicked documents

Inferred edges

Luretech Hot Hooks
Hi-Tech Fish 'N' Bucket
Eskimo Mako Auger
Strike-Lite II Auger

User

fish jigs

ice jigs

Ice fishing tackle

Fishing bucket

Ice fishing

d rush

Inferred edges

Construction of Click Graph
### Associations from Instances to Queries

<table>
<thead>
<tr>
<th>Query</th>
<th>$P_{\text{ml}}$</th>
<th>$P_{\text{nml}}$</th>
<th>Query</th>
<th>$P_{\text{ml}}$</th>
<th>$P_{\text{nml}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Garmin GTM 20 GPS</td>
<td>0.44</td>
<td>0.45</td>
<td>Canon PowerShot SX110 IS</td>
<td>0.57</td>
<td>0.57</td>
</tr>
<tr>
<td>garmin gtm 20</td>
<td>0.30</td>
<td>0.27</td>
<td>powershot sx110</td>
<td>0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>garmin traffic receiver</td>
<td>0.02</td>
<td>0.02</td>
<td>powershot sx110 is</td>
<td>0.38</td>
<td>0.36</td>
</tr>
<tr>
<td>garmin nuvi 885t</td>
<td>0.00</td>
<td>0.01</td>
<td>powershot sx130 is</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>gtm 20</td>
<td>0.00</td>
<td>0.00</td>
<td>canon power shot sx110</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>garmin gtm20</td>
<td>0.00</td>
<td>0.00</td>
<td>canon dig camera review</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>nuvi 885t</td>
<td>0.00</td>
<td>0.00</td>
<td>Devil May Cry: 5th Anniversary Col.</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>Samsung PN50A450 50&quot; TV</td>
<td>0.75</td>
<td>0.83</td>
<td>devil may cry</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>samsung 50 plasma hdtv</td>
<td>0.33</td>
<td>0.32</td>
<td>devil mansary</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>50&quot; hdtv</td>
<td>0.17</td>
<td>0.12</td>
<td>High Island Hammock/Stand Combo</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>samsung plasma tv review</td>
<td>0.00</td>
<td>0.42</td>
<td>high island hammocks</td>
<td>0.00</td>
<td>0.10</td>
</tr>
<tr>
<td>50&quot; samsung plasma hdtv</td>
<td>0.00</td>
<td>0.35</td>
<td>hammocks and stands</td>
<td>0.00</td>
<td>0.10</td>
</tr>
</tbody>
</table>

- Associations from Queries to Instances
- Instances suggested via observed clicks and inferred "clicks" on instances

<table>
<thead>
<tr>
<th>Query</th>
<th>Product Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>wedding gowns</td>
<td>27 Dresses (Movie Soundtrack)</td>
</tr>
<tr>
<td>wedding gowns</td>
<td>Bridal Gown: The Basics of Designing, [...] (Book)</td>
</tr>
<tr>
<td>wedding gowns</td>
<td>The Perfect Wedding Dress (Magazine)</td>
</tr>
<tr>
<td>wedding gowns</td>
<td>Imagine Wedding Designer (Video Game)</td>
</tr>
<tr>
<td>low blood pressure</td>
<td>Omron Blood Pressure Monitor</td>
</tr>
<tr>
<td>low blood pressure</td>
<td>Healthcare Automatic Blood Pressure Monitor</td>
</tr>
<tr>
<td>low blood pressure</td>
<td>Ridgecrest Blood Pressure Formula - 60 Capsules</td>
</tr>
<tr>
<td>low blood pressure</td>
<td>Omron Portable Wrist Blood Pressure Monitor</td>
</tr>
<tr>
<td>‘hello cupcake’ cookbook</td>
<td>Giant Cupcake Caddy</td>
</tr>
<tr>
<td>‘hello cupcake’ cookbook</td>
<td>Ultimate 3-In-1 Storage Caddy</td>
</tr>
<tr>
<td>‘hello cupcake’ cookbook</td>
<td>13 Cup Cupcakes and More Dessert Stand</td>
</tr>
<tr>
<td>‘hello cupcake’ cookbook</td>
<td>Cupcake Stand Set (Toy)</td>
</tr>
<tr>
<td>1,800 flowers</td>
<td>Todd Oldham Party Perfect Bouquet</td>
</tr>
<tr>
<td>1,800 flowers</td>
<td>Hugs and Kisses Flower Bouquet with Vase</td>
</tr>
</tbody>
</table>
Query Understanding


Recommending Related Instances

- Input
  - query submitted by a user
  - main structured instance, if any, returned by search engine as a side result for the query
- Data source
  - click data for search results returned in response to queries
  - click data for main instances, if any, returned as side results in response to queries
  - collection of instances available within a structured knowledge repository (e.g., Freebase)
- Output
  - list of instances from knowledge repository deemed relevant to the query and the main instance, based on click data available for the user
  - similar to query suggestion, but suggestions are instances not strings, and suggestions are for instances rather than queries
- Steps
  - exploit three types of evidence, namely edges between instances within knowledge repository; edges between main instances in side results and clicked documents; and edges between main instances in side results and clicked related instances in side results
  - given a main instance, recommend a list of related entities based on the user's interests
Main Instance and Related Instances

Evidence Towards Related Instances
Recommended Instances

Mean reciprocal rank of relevant instance in computed ranked list of related instances

Co-click: evidence from user clicks on both main instance and related instance
CTR-model: evidence from click-through rate for related instances being returned
TEM: all sources of evidence

Query Understanding

Web Usage Understanding

- **Data source**
  - query sessions including queries and clicked documents
  - collection of instances available within a structured knowledge repository (Freebase and DBpedia)
- **Output**
  - semantic rather than lexical patterns of Web usage mining
- **Steps**
  - annotate queries, by linking query fragments to corresponding instances from knowledge repository
  - use properties available for instances within knowledge repository to generalize and categorize queries

Linking Queries to Instances

- **Link queries to instances**
  - using types available for instances in the knowledge repository, compute types of queries that lead to Web sites

(Courtesy L. Hollink)

- **Generalize from linked queries to query types**
  - using instance types from knowledge repository, generalize queries into query types that lead to Web sites
Patterns of Web Usage

• Estimate the differences in the content of two Web sites, by comparing the top query types that lead to clicks on the sites

The first Web site specializes in movies, whereas the second is broader as it covers content related to movies, TV and people.

Patterns of Web Usage

• Compare query patterns obtained by generalizing from queries linked to instances that are relatively recent movies; vs. instances that are relatively older movies

<table>
<thead>
<tr>
<th>Level</th>
<th>Pattern</th>
<th>Support</th>
<th>Level</th>
<th>Pattern</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>movie</td>
<td>0.396</td>
<td></td>
<td>movie</td>
<td>0.381</td>
</tr>
<tr>
<td></td>
<td>2011</td>
<td>0.15</td>
<td></td>
<td>cast</td>
<td>0.096</td>
</tr>
<tr>
<td></td>
<td>trailer</td>
<td>0.103</td>
<td></td>
<td>movies</td>
<td>0.091</td>
</tr>
<tr>
<td></td>
<td>movies</td>
<td>0.095</td>
<td></td>
<td>quotes</td>
<td>0.038</td>
</tr>
<tr>
<td></td>
<td>movies!</td>
<td>0.063</td>
<td></td>
<td>trailer</td>
<td>0.034</td>
</tr>
<tr>
<td></td>
<td>cast</td>
<td>0.053</td>
<td></td>
<td>new</td>
<td>0.027</td>
</tr>
<tr>
<td></td>
<td>new</td>
<td>0.049</td>
<td></td>
<td>free</td>
<td>0.023</td>
</tr>
<tr>
<td></td>
<td>cast</td>
<td>0.035</td>
<td></td>
<td>soundtrack</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>release</td>
<td>0.022</td>
<td></td>
<td>watch</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>reviews</td>
<td>0.028</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>movie → movie</td>
<td>0.165</td>
<td></td>
<td>movie → movie</td>
<td>0.169</td>
</tr>
<tr>
<td></td>
<td>2011 → 2011</td>
<td>0.042</td>
<td></td>
<td>movies → movies</td>
<td>0.036</td>
</tr>
<tr>
<td></td>
<td>2011 → movie</td>
<td>0.04</td>
<td></td>
<td>cast → cast</td>
<td>0.028</td>
</tr>
<tr>
<td></td>
<td>movie → 2011</td>
<td>0.038</td>
<td></td>
<td>movies → movie</td>
<td>0.025</td>
</tr>
<tr>
<td></td>
<td>movie → movie</td>
<td>0.038</td>
<td></td>
<td>quotes → quotes</td>
<td>0.019</td>
</tr>
<tr>
<td></td>
<td>trailer → trailer</td>
<td>0.027</td>
<td></td>
<td>movie → cast</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td>movie → movie 2011</td>
<td>0.026</td>
<td></td>
<td>movie → trailer</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>movie → trailer</td>
<td>0.025</td>
<td></td>
<td>movie → moviecast</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>movie 2011 → movie</td>
<td>0.024</td>
<td></td>
<td>new → new</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Next Topic

- Document analysis and understanding
- Query analysis and understanding
- Matching of queries onto documents
- Onebox search results

Matching of Queries onto Documents

- [Voo94], E. Voorhees. Query Expansion Using Lexical-Semantic Relations. SIGIR-94.
Query Expansion Using Lexical Resources

- **Goal**
  - investigate the role of concepts and relations available in WordNet in the expansion of queries, for the purpose of improving the quality of retrieved documents
- **Procedure**
  - manually or automatically identify WordNet concepts corresponding to query terms
  - collect expansion terms from among the synonym, more general and more specific concepts of the identified concepts
  - expand queries using the expansion terms
- **Findings**
  - with manual identification of WordNet concepts, the expansion of queries improves results for underspecified queries, and does not improve results for well-specified queries
  - with automatic identification of WordNet concepts, the expansion of queries degrades results

Matching of Queries onto Documents

- [Fan08]: H. Fang. A Re-Examination of Query Expansion Using Lexical Resources. ACL-08.
Query Expansion Using Lexical Resources

• Goal
  – revisit the task of query expansion using concepts and relations available in WordNet
• Procedure
  – focus on the assignment of appropriate weights to expansion terms, such that terms selected for expansion are strongly related to query terms
  – weights capture similarity among query terms and expansion terms
  – term similarity functions use synonym vs. more general vs. more specific concepts vs. overlap of concept definitions
• Findings
  – the expansion of queries with terms from WordNet improves results
  – improvement is largest when similarity between terms is computed as the overlap of their definitions in WordNet
  – combining multiple similarity functions gives no additional improvement
  – query expansion using WordNet is not better than query expansion using expansion terms that co-occur with query terms in the document collection (pseudo-relevance feedback using global analysis)

Matching of Queries onto Documents

Query Expansion Using Semantic Sources

- **Goal**
  - investigate the role of concepts and relations available in ConceptNet in the expansion of queries, for the purpose of improving the quality of retrieved documents
  - focus on difficult (i.e., poorly performing) queries
- **Procedure**
  - manually or automatically identify ConceptNet concepts related to query terms
  - collect expansion terms from among concepts available in the ConceptNet graph of concepts and relations, within a certain distance away from the identified concepts
  - expand queries using the expansion terms
- **Findings**
  - with manual identification of ConceptNet concepts, there is some possible expansion of queries that improves results, for all difficult queries
  - expansion terms manually selected from ConceptNet give better results than expansion terms automatically selected from top results (pseudo-relevance feedback using local analysis)

Impact of Query Expansion

- **Manual selection of ConceptNet concepts for expansion**

  (Courtesy A. Kotov)

<table>
<thead>
<tr>
<th></th>
<th>KL</th>
<th>KL-PF</th>
<th>CF-1</th>
<th>CF-2</th>
<th>CF-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AQUAINT</td>
<td>0.0521</td>
<td>0.0429</td>
<td>0.1247</td>
<td>0.1622</td>
<td>0.1880</td>
</tr>
<tr>
<td>ROBUST04</td>
<td>0.0509</td>
<td>0.0788</td>
<td>0.1061</td>
<td>0.1539</td>
<td>0.1823</td>
</tr>
<tr>
<td>GOV</td>
<td>0.0748</td>
<td>0.0447</td>
<td>0.1830</td>
<td>0.3481</td>
<td>0.4826</td>
</tr>
</tbody>
</table>

  Concepts from top retrieved documents (pseudo-relevance feedback)  
  Concepts in ConceptNet within radius 2 of the identified concepts
Impact of Query Expansion

- Automatic selection of ConceptNet concepts for expansion

<table>
<thead>
<tr>
<th></th>
<th>KL</th>
<th>KL-PF</th>
<th>QPATH</th>
<th>RWALK</th>
<th>LR-2</th>
<th>LR-PF-2</th>
<th>LR-3</th>
<th>LR-PF-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AQUAINT</td>
<td>0.0521</td>
<td>0.0429</td>
<td>0.0538</td>
<td>0.0534</td>
<td>0.0535</td>
<td>0.0662</td>
<td>0.0571</td>
<td>0.0776</td>
</tr>
<tr>
<td>ROBUST04</td>
<td>0.0509</td>
<td>0.0788</td>
<td>0.0542</td>
<td>0.0559</td>
<td>0.0604</td>
<td>0.0844</td>
<td>0.0588</td>
<td>0.0837</td>
</tr>
<tr>
<td>GOV</td>
<td>0.0748</td>
<td>0.0447</td>
<td>0.1034</td>
<td>0.1179</td>
<td>0.1293</td>
<td>0.1119</td>
<td>0.1236</td>
<td>0.0914</td>
</tr>
</tbody>
</table>

Heuristic-based selection of concepts from ConceptNet
Learning-based selection of concepts from ConceptNet
Combination of learning-based selection of concepts from ConceptNet and pseudo-relevance feedback

Next Topic

- Document analysis and understanding
- Query analysis and understanding
- Matching of queries onto documents
- Onebox search results
Retrieval of OneBox Results


Extracting and Retrieving How-To Tips

• Input
  – queries
• Data source
  – collaboratively-created collection of pairs of a question and an answer
• Output
  – a tip (to round a decimal in c: add 0.5 and then floor the value) in the format (tip goal: tip suggestion), selected from a set of tips extracted in advance from the question-answer pairs
  – returned only for queries deemed to have how-to intent (how to round a decimal in c, how do you fix keys on a laptop, clean iphone screen)
• Steps
  – construct tips, from pairs of a "how to" question and its answer
  – for queries with how-to intent, retrieve tip whose goal best matches the queries
Tips from Question-Answer Pairs

**Resolved Question**

How do I round a decimal in C++?

I'm using a float and it just drops the decimal. Say I have 1.1 I need that to go to 1 and if I have 2.5 I need that to go to 3 thanks.

**Best Answer - Chosen by Voters**

To round a decimal in C++ add 0.5 and then floor the value – what you call dropping the decimal. Looks like this: int x = (my_float + 0.5);

Tip goal: 100% 1 Vote

Tip suggestion: 1 person rated this as good

(Courtesy I. Weber)

Answering How-To Queries

Yahoo! Answers uses rule-based extraction to obtain 250k candidate tips. Machine learning is used to obtain 27k high quality tips. For a query like 'zest lime without zester', the system checks if the query has a how-to intent. If not, it shows normal search results. If yes, it checks if there are relevant high quality tips. If not, it shows normal search results. If yes, it ranks the matching tips and displays the highest ranking one.

TIP: To zest a lime if you don’t have a zester: use a cheese grater.
Retrieval of OneBox Results


Extracting and Retrieving Facts

**Input**
- queries

**Data source**
- collection of tables identified within Web documents

**Output**
- one-box search result containing the fact (paris: bay city), if any, deemed to most confidently answer the user’s query (france capital; where was madonna born)
- selected from a set of facts (tuples of an instance, attribute and value) extracted in advance from Web tables

**Steps**
- identify subset of Web tables containing attribute-value pairs
- from attribute-value pairs in a table, and the instance identified to be the main topic of the document containing the table, extract instance-attribute-value tuples
- if query is deemed to be a fact lookup query, retrieve the value with the highest confidence among values, if any, present in the tuples for the instance and attribute specified in the query
System Architecture

Data extraction

Query answering

Extraction of Factual Tuples

- **Table classifier**
  - distinguish attribute-value tables from other types of tables

<table>
<thead>
<tr>
<th></th>
<th>Attribute-Value Tables</th>
<th>Relational Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>% among all tables</td>
<td>6.6%</td>
<td>1.6%</td>
</tr>
<tr>
<td>avg # of instances</td>
<td>1</td>
<td>10.3</td>
</tr>
<tr>
<td>avg # of attributes</td>
<td>14.3</td>
<td>3.6</td>
</tr>
<tr>
<td>avg. # of data elements</td>
<td>14.3</td>
<td>38.8</td>
</tr>
<tr>
<td>% of numerical data elements</td>
<td>8.8%</td>
<td>62.8%</td>
</tr>
</tbody>
</table>

- **Pattern summarizer**
  - analyzing sets of documents with the same format, identify and discard spurious attribute-value tables

<table>
<thead>
<tr>
<th></th>
<th>Log in</th>
<th>Contact us</th>
<th>Britney Spears</th>
<th>Paris Hilton</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td>Customer services</td>
<td>Jennifer Lopez</td>
<td>Jessica Simpson</td>
<td></td>
</tr>
<tr>
<td>About us</td>
<td>Store locations</td>
<td>Madonna</td>
<td>Jessica Alba</td>
<td></td>
</tr>
</tbody>
</table>
Extraction of Factual Tuples

- **Entity extractor**
  - extract the main instances about which the source Web documents, and the attribute-value tables that they contain, are about
  ... → repository of instance-attribute-value tuples

- **Attribute equivalence detector**
  - attributes that have the same value for the same instance tend to be equivalent

<table>
<thead>
<tr>
<th>address</th>
<th>location</th>
<th>phone</th>
<th>price</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>addresse</td>
<td>direccion</td>
<td>telephone</td>
<td>list price</td>
<td>gewicht</td>
</tr>
<tr>
<td>street address</td>
<td>tel</td>
<td>regular price</td>
<td>poids</td>
<td></td>
</tr>
<tr>
<td>dirección</td>
<td>admissions</td>
<td>our price</td>
<td>peso</td>
<td></td>
</tr>
<tr>
<td>dirección</td>
<td>admissions</td>
<td>your price</td>
<td>waga</td>
<td></td>
</tr>
</tbody>
</table>

Query Answering

- **Query parser**
  - match queries against small set of manually-written rules ("E A", "E's A", "who was the A of E", "when was E born")

- **Instance equivalence detector**
  - instances whose vectors of search-result click counts are very similar to one another are deemed equivalent
  - considered very similar, when vectors have cosine ≥ 0.5:

<table>
<thead>
<tr>
<th>Equivalent?</th>
<th>Cause of Error</th>
<th>Pct</th>
<th>Example of Instance Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>N/A</td>
<td>87%</td>
<td>australian job vs. job in australia</td>
</tr>
<tr>
<td>No</td>
<td>One is more specific than the other</td>
<td>7%</td>
<td>flightless bird vs. large flightless bird</td>
</tr>
<tr>
<td>No</td>
<td>One is an aspect of the other</td>
<td>5%</td>
<td>will county vs. map of will county</td>
</tr>
<tr>
<td>No</td>
<td>Different</td>
<td>1%</td>
<td>1972 chevrolet suburban vs. 1968 chevrolet suburban</td>
</tr>
</tbody>
</table>
Query Answering

- Structured query engine
  - given a query, generate instance-attribute pairs by replacing entity with equivalent entity or attribute with equivalent attribute
  - lookup instance-attribute pairs in instance-attribute-value tuples
- Result aggregator
  - single or multiple lookups may result in retrieval of multiple values
  - select value if extracted from more Web domains, and if similar to more of the other values

<table>
<thead>
<tr>
<th>Type of Answering Error</th>
<th>Example of Query - Erroneous Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer is wrong</td>
<td>turkey language - english</td>
</tr>
<tr>
<td>answer is incomplete</td>
<td>george bush date of birth - 1946</td>
</tr>
<tr>
<td>answer is relevant for another query</td>
<td>how santa monica college was founded - 1929</td>
</tr>
<tr>
<td>query is an instance</td>
<td>microsoft publisher - (any)</td>
</tr>
<tr>
<td>query is navigational</td>
<td>lil wayne myspace - (any)</td>
</tr>
<tr>
<td>query should not trigger an answer</td>
<td>watch free movies - (any)</td>
</tr>
</tbody>
</table>

Retrieval of OneBox Results

Question Answering as Binary Classification

- **Input**
  - natural-language questions (What is the name of Justin Bieber’s brother?)
- **Data source**
  - knowledge repository of inter-connected topics (Freebase)
  - collection of Web documents
- **Output**
  - topics that answer the questions (Jaxon Bieber)
- **Steps**
  - convert the question into a question graph
  - based on the node from the question graph corresponding to the question topic (Justin Bieber), assemble a topic graph of inter-connected topics up to a few hops away from the question topic
  - using individual and combination features from question graph and topic graph, determine whether each node from the topic graph is or is not an answer to the question

Answers from Knowledge Repositories

(Courtesy X. Yao)

Question: What is the name of Justin Bieber’s brother?
Question: What is the name of Justin Bieber's brother?

Answer: Jaxon Bieber
Features for Classification

- Features from dependency parse trees of questions
  - qword=what
  - qfocus=name
  - qverb=be
  - qtopic=person
  - qword=what|cop|qverb=be
  - qword=what|nsbj|qfocus=name
  - brother|nn|qtopic=person
  - ...

- Features from knowledge repository

  - Has awards
  - Has place of birth
  - Has sibling
  - Type person
  - ...

  - Justin Bieber
  - Gender: female
  - Type: person
  - ...

  - Jazmyn Bieber
  - Gender: female
  - Type: person
  - ...

  - Jaxon Bieber
  - Gender: male
  - Type: person
  - ...

Combined features from knowledge repository and questions
Estimated Utility of Features

- Align relations from knowledge repository to phrases that may express the relations in document sentences
  - film/starring (Gravity, Sandra Bullock) vs.
  - Sandra then was cast in Gravity, a two actor spotlight film
  - Sandra Bullock plays an astronaut hurtling through space in new blockbuster “Gravity”
  - Sandra Bullock stars/acts in Gravity
  - Sandra Bullock conquered her fears to play the lead in Gravity

Mapping Relations to Phrases

- Use alignments to predict relevant relations when answering questions
Summary

- Knowledge and its acquisition from textual data have the potential to enhance Web search
  - sources of textual data: documents, queries
  - impact on content understanding: query and document analysis, query-document matching
  - impact on alternative search interfaces: structured search, answer retrieval