Knowledge Base Construction with Epistemological Databases

Andrew McCallum

Department of Computer Science
University of Massachusetts Amherst

Joint work with Sameer Singh, Michael Wick, Limin Yao, Sebastian Riedel, Karl Schultz, Aron Culotta.
W. Bruce Croft

Distinguished Professor
Department of Computer Science, University of Massachusetts
BRUCE CROFT, Amherst, MA, 01003-9264
Email: croftg@cs.umass.edu
URL: http://ciir.cs.umass.edu/personnel/croft.html

Publications: (1 to 40 of 233) (total 1436 citations)
Sorted by date | citations
2004
- Xiaoyong Liu, W. Bruce Croft. *Cluster-based retrieval using language models*. SIGIR, 2004 (0 citations)
- Andrés Corrada-Emmanuel, W. Bruce Croft. *Answer models for question answering passage retrieval*. SIGIR, 2004 (0 citations)
- Chirag Shah, W. Bruce Croft. *Evaluating high accuracy retrieval techniques*. SIGIR, 2004 (1 citation)

2003
- W. Bruce Croft. *Language Models for Information Retrieval*. ICDE, 2003 (0 citations)
Goal Application

A KB of all scientists in the world from papers, reports, web pages, newswire, press releases, blogs, patents,..

• Better tools → Accelerate progress of science.
• Help...
  - find papers to read, to cite
  - find reviewers, collaborators, people to hire
  - understand trends and landscape of science
• Platform for a “New Model of Publishing” [LeCun]
  - post to archive; public comments and ratings.
Attributes of our Task

A KB of all scientists in the world
from papers, reports, web pages, newswire, press releases, blogs, patents,..

• Open universe of entities (strong entity resolution essential)
  - not coref into pre-known finite set e.g. in Wikipedia

• Closed list of relation types*
  - not OpenIE *later “open” through “universal schema”

• Low tolerance for error
  - users willing to edit

• Changing world
  - e.g. new papers, people moving institutions,...
Wei Li studies at Xinghua U. Her 2008 publications include W. Li. “Scalable NLP” ACL, 2008.

Knowledge Base Construction

Information Extraction components aren’t perfect. Errors snowball.
Knowledge Base Construction

1. How to represent & inject uncertainty from IE into DB?
2. Want to use DB contents to aid IE.
3. IE isn’t “one-shot.” Add new data later; redo inference.

Want DB infrastructure to manage IE.

[POS & shallow parsing, ICML 2004]
[Entity & Relation Extraction, ACL, 2011]
**Human Edits as evidence:** [Wick, Schultz, McCallum 2012]

- Traditional: Change DB record of truth
- ✔ Mini-document “Nov 15: Scott said this was true”

- Sometimes humans are wrong, disagree, out-of-date.
- Jointly reason about truth & editors’ reliability/reputation.

“Truth is inferred, not observed.”
“Epistemological Database”

Never Ending Inference [Riedel, Wick, McCallum 2012]

✘ KB entries locked in
✔ KB entries always reconsidered with more evidence, time, ...

Never Ending Inference [Riedel, Wick, McCallum 2012]

✘ KB entries locked in
✔ KB entries always reconsidered with more evidence, time, ...
Resolution is foundational  

[KDD 2008; ACL 2012]

✗ Not just for coref of entity-mentions...

✔ Align values, ontologies, schemas, relations, events,...

Especially in Epistemological DB: entities/relations never input, only “mentions”
“Epistemological Database”

Resource-bounded Information Gathering [WSDM 2012]

✘ Full processing on whole web
✔ Focus queries and processing where needed & fruitful
“Epistemological Database”

Inference constantly bubbling in background...

Entity Extraction → Relation Extraction → Resolution (Coref) → KB

- Entity Mention Extraction
- Relation Mention Extraction
- Resolution (Coref)

p(Entity Mentions) → p(Relation Mentions) → p(Entities, Relations)

Structured Data

Human Edits

Text docs

Evidence

KB

Inference worker

Smart Parallelism  [ACL 2011; NIPS 2011]

✗ MapReduce, black-box

✔ Reason about inference & parallelism together
MCMC, parallel, distributed  [ACL 2011; submitted 2012]

✘ Unroll whole factor graph. Limited model structures.
✓ Focused sampling, conflict resolution, particle filtering
“Epistemological Database”

MCMC, parallel, distributed  [ACL 2011; submitted 2012]

✘ Unroll whole factor graph. Limited model structures.
✔ Focused sampling, conflict resolution, particle filtering
“Epistemological Database”

MCMC, parallel, distributed [ACL 2011; submitted 2012]

✔ Focused sampling, conflict resolution, particle filtering

✘ Unroll whole factor graph. Limited model structures.
“Epistemological Database”

MCMC, parallel, distributed  [ACL 2011; submitted 2012]
✖️  Unroll whole factor graph.  Limited model structures.
✔️  Focused sampling, conflict resolution, particle filtering
“Epistemological Database”

Entity Extraction → Relation Extraction → Resolution (Coref) → Structured Data

Text docs → Human Edits → query

p(Entity Mentions) → p(Relation Mentions) → p(Entities, Relations)

Samples p("truth")

Inference worker → Inference worker → Inference worker → Inference worker → Inference worker → Inference worker

Inference constantly bubbling in background...

MCMC, parallel, distributed [ACL 2011; submitted 2012]

✗ Unroll whole factor graph. Limited model structures.
✔ Focused sampling, conflict resolution, particle filtering
Research Ingredients

1. Learning
2. Entity Resolution
3. Human Edits
4. Relations with “Universal Schema”
5. Probabilistic Programming
Entity Resolution

Parallel / Distributed
Interplay between modeling & efficiency
Entity resolution by CRF with pairwise factors
Entity Resolution

Entity resolution by CRF with pairwise factors
Entity resolution by CRF with pairwise factors
Entity Resolution

Entity resolution by CRF with pairwise factors
Entity Resolution

Entity resolution by CRF with pairwise factors
Entity resolution by CRF with pairwise factors
Entity resolution by CRF with pairwise factors.

These two proposals can be evaluated (and accepted) in parallel.
Entity Resolution in Parallel by Map-Reduce

[Singh, Subramanian, Pereira, McCallum, ACL, 2011]
Parallelism = faster
Distributed Entity Resolution
with hierarchical structure

Entity resolution by CRF with pairwise factors

Super-entities infer good “data distribution”

Sub-entities infer good “block moves”

Inference used not only for “truth discovery”, but also simultaneously for “strategizing about data distribution”
Smart Parallelism = much faster

[Singh, Subramanian, Pereira, McCallum, ACL, 2011]
Pair-based Coref
Entity-based Coref

Super-Entity
Entity
Sub-Entity
Mention
Entity-based Coref

Super-Entity
Entity
Sub-Entity
Mention
More efficient. Fewer factors; avoid $N^2$.
Joint inference on all attributes of entity. Pair-wise couldn’t
50k mentions “Bill Clinton” hidden under one sub-entity.
Avoid CRF problems with “changes in network cardinality”
Better supports human edits

[Wick, Schultz, McCallum, ACL, 2012]
Hierarchical vs Pairwise Evaluation

145k mentions

Accuracy versus Time

1.3m mentions

Accuracy versus Time

Currently: 80m mentions
papers, authors, institutions, venues
Entity-based Coref for Wikipedia & Newswire

- Combine structured data...
  Freebase & Wikipedia infobboxes

- ...with unstructured text.
  NYTimes articles
Mr. [Moyo |PER] had shut down most of the nation's private newspapers and amassed wide influence within the government before being implicated last month in a scheme to prevent [Joyce Mujuru |PER], a regional politician, from taking a vacant post as [Zimbabwe |LOC]'s vice president. Ms. [Mujuru |PER] was the choice of President [Robert G. Mugabe |PER], and she is currently running the country while he is on a vacation in [Malaysia |LOC].

Currently: 100k Wikipedia entities, 20 years NYTimes
4m anchor texts, 300k unique mention strings
Entity Resolution

Parallel / Distributed
Interplay between modeling & efficiency

Open Questions

*Lots of juicy research at ML+systems intersect*

- Formalize asynchronous distributed MCMC.
- How to select subset of variables for worker.
- Get coref working for 10 billion mentions...
#3

Probabilistic Reasoning about Human Edits

Humans will want to correct DB, add to DB
Entity-based Coref

Pereira
SRI
Google
Entity-based Coref
 Entity-based Coref

[Wick, Schultz, McCallum, AKBC, 2012]
Benefits of Probabilistic Reasoning about Human Edits

Database quality versus the number of correct human edits

Edit incorporation strategy
- Epistemological (probabilistic)
- Overwrite
- Maximally satisfy

Our probabilistic reasoning

Local Transitive Closure

Traditional Overwrite
Robustness to Errorful Human Edits

![Graph showing robustness to errorful human edits]

**Edit incorporation strategy**
- Epistemological (probabilistic)
- Complete trust in users

**Our probabilistic reasoning**

**Traditional Overwrite**
Benefits of Probabilistic Reasoning about Streaming Evidence

Quality of original DB as new structured evidence arrives

- Knowledge Base
  - Epistemological database
  - Traditional KB

Our probabilistic reasoning

Traditional Overwrite

F1 accuracy of original database mentions

Amount of evidence (no. of additional BibTeX mentions)
#3
Probabilistic Reasoning about Human Edits

Humans will want to correct DB, add to DB

Open Questions

- Edits: efficient forward chaining; robust to noise
- Streaming inputs: what to keep, toss, summarize
Relations with “Universal Schema”

Relation extraction
without labeled data
without pre-fixed schema
Styles of Relation Extraction

- **Supervised**

Schema: \{ advised, affiliated, authored, ... \}

Labeled Data:

- Jane Smith attends MIT.

Test Data:

- Ted Jones studies at Harvard.

Prediction:

- affiliated(Ted Jones, Harvard)
Styles of Relation Extraction

• Supervised
• Distantly Supervised

\{ advised, affiliated, authored, … \}
Styles of Relation Extraction

- Supervised
- Distantly Supervised
- Unsupervised (no schema)  OpenIE

Ted Jones attends Harvard.

dependency parse (or approximation)

attends(Ted Jones, Harvard) ≠ affiliated
Styles of Relation Extraction

- Supervised
- Distantly Supervised
- Unsupervised (no schema) OpenIE
- Unsupervised (schema discovery) clustering

Relation #1
- affiliated
- attends
- studies at
- professor at
- employed by

Relation #2
- advised
- is the advisor of
- supervised
- chaired thesis of
- is the mentor of

Relation #3
- authored
- wrote
- published
- was co-author of
- 's paper

Arbitrary
Hard to evaluate
Incomplete
Many boundary cases
Styles of Relation Extraction

- Supervised
- Distantly Supervised
- Unsupervised (no schema) OpenIE
- Unsupervised (schema discovery) clustering

Vanderwende to Hovy: Where do the relation types come from?

Freebase: No relation for “criticized”

ANY SCHEMA
Incomplete
Many boundary cases
Styles of Relation Extraction

- Supervised
- Distantly Supervised
- Unsupervised (no schema) OpenIE
- Unsupervised (schema discovery)
- Unsupervised ("universal schema")

[Yao, Riedel, McCallum, AKBC 2012]
Prob DB of “Universal Schema”

• Schema = union of all inputs: NL & DBs
  - embrace diversity and ambiguity of original inputs
  - don’t try to force it into pre-defined boxes

• Learn implicature among entity-relations
  - “fill in” unobserved relations
Text documents: relations from dependency parses

<table>
<thead>
<tr>
<th>president of</th>
<th>prime minister of</th>
<th>chancellor of</th>
<th>chief executive</th>
<th>leader of</th>
<th>head of state</th>
<th>headOf</th>
<th>Top Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obama, U.S.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merkel, Germany</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S Harper Canada</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V Putin Russia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Larry Page Google</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V. Rometty IBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tim Cook Apple</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E Grimson MIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

350k+ rows

23k+ columns
## Prob DB of “Universal Schema”

Text documents: relations from dependency parses

<table>
<thead>
<tr>
<th>president of</th>
<th>prime minister of</th>
<th>chancellor of</th>
<th>chief executive</th>
<th>leader of</th>
<th>head of state</th>
<th>headOf</th>
<th>Top Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obama, U.S.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Merkel, Germany</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>S Harper Canada</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>V Putin Russia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Larry Page Google</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>V. Rometty IBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Tim Cook Apple</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>E Grimson MIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

350k+ rows

Model & fill in with Generalized Principle Components Analysis (ala NetFlix)

23k+ columns

Combination of structured and OpenIE
# Prob DB of “Universal Schema”

## Text documents: relations from dependency parses

<table>
<thead>
<tr>
<th>president of</th>
<th>prime minister of</th>
<th>chancellor of</th>
<th>chief executive</th>
<th>leader of</th>
<th>head of state</th>
<th>headOf</th>
<th>Top Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obama, U.S.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Merkel, Germany</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>S Harper Canada</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>V Putin Russia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Larry Page Google</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>V. Rometty IBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Tim Cook Apple</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>E Grimson MIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

350k+ rows

Model & fill in matrix with **Generalized Principle Components Analysis** (ala NetFlix)

[Yao, Riedel, McCallum, AKBC 2012]
Successfully predicts
“Forbes criticized George Bush.”
Successfully predicts
“Volvo owns percentage of Scania A.B.”
from “Volvo bought a stake in Scania A.B.”
Prob DB of “Universal Schema”

Learns asymmetric entailment:

PER historian at UNIV → PER professor at UNIV

but

PER professor at UNIV ⊬ PER historian at UNIV
Experimental Results

• 20 years NYTimes
  - extract entity mentions, perform entity resolution
  - 350k entity pairs, 23k unique relation surface forms

• Freebase
  - 6k entity pairs resolved with NYTimes pairs
  - 116 relations

<table>
<thead>
<tr>
<th>Relation Prediction</th>
<th>w/out Freebase</th>
<th>with Freebase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.687</td>
<td>0.666</td>
</tr>
<tr>
<td>Recall</td>
<td>0.491</td>
<td>0.520</td>
</tr>
</tbody>
</table>
Prob DB of “Universal Schema”

- **Summary**
  - Embrace diversity and ambiguity of original inputs; don’t try to force it into pre-defined boxes.
  - Reason about entities & relations together; not an abstract relation-relation mapping.
  - User can query without understanding a limited schema; ask and we probably have a column for that.
  - Model to predict original expressions (well defined task); do not try to create models of semantic equivalence (illusive).
Prob DB of “Universal Schema”

• Summary
  - Embrace diversity and ambiguity of original inputs; don’t try to force it into pre-defined boxes.
  - Reason about entities & relations together; not an abstract relation-relation mapping.
  - User can query without understanding a limited schema; ask and we probably have a column for that.
  - Model to predict original expressions (well defined task); do not try to create models of semantic equivalence (illusive).

• Related Work
  - OpenIE [Etzioni…], but we also “fill in” unobserved relations
  - Clustering [Pantel; Yates; Yao], but we learn asymmetric
  - Rules between textual patterns [Schoenmackers et al. 2008], similar goals, but we avoid limited tree-width & batch-mode learning
#4 Relations with “Universal Schema”

Relation extraction without labeled data; without pre-fixed schema

**Future Work**

- Incorporate relations with different arities
- Integrate background knowledge
- Scale up further in both pairs and relations
#5

Prob-Programming, its Integration with Prob-DB

Need way to easily specify models.
Schema Matching

\[ P(Y \mid X) = \frac{1}{Z_X} \prod_{y_i \in Y} \psi_w(y_i, x_i) \prod_{y_i, y_j \in Y} \psi_b(y_{ij}, x_{ij}) \]

\[ \psi(y_i, x_i) = \exp \left( \sum_k \lambda_k f_k(y_i, x_i) \right) \]

Coreference and Canonicalization
Schema Matching

\[ P(Y \mid X) = \frac{1}{Z_X} \prod_{y \in Y} \psi_w(y_i, x_i) \prod_{y_i, y_j \in Y} \psi_b(y_i, x_j) \]

Really Hairy Models!

How to do

- parameter estimation
- inference

Coreference and Canonicalization

\[ \psi(y_i, x_i) = \exp \left( \sum_k \lambda f_k(y_i, x_i) \right) \]
Schema Matching

$$P(Y \mid X) = \frac{1}{Z} \prod_{y_i \in Y} \psi_w(y_i, x_i) \prod_{y_i, y_j \in Y} \psi_b(y_i, x_i)$$

$$\psi(y_i, x_i) = \exp \sum_k \lambda f_k(y_i, x_i)$$

Really Hairy Models!

How to do

• parameter estimation
• inference
• software engineering

Coreference and Canonicalization
Probabilistic Programming Languages

- Make it easy to specify rich, complex models, using the full power of programming languages
  - data structures
  - control mechanisms
  - abstraction

- Inference implementation comes for free

Provides language to easily create new models
Our Approach to Probabilistic Programming

**FACTORIE**
http://factorie.cs.umass.edu

- **Object-oriented**: Variables, factors, inference & learning methods are objects,.. inheritance…
- **Imperative** definition of construction & operation
- **Embedded** in a general-purpose prog. language.
- **Scalable** to billions of variables and factors. Tightly integrates into DB back-end, providing PrDB.

Implemented in Scala

Replacement for MALLET
Prob-Programming & its Integration with Prob-DB

Need way to easily specify models. Tight coupling $\rightarrow$ efficiency, scalability.

Open Questions

- Tools for prob programming, e.g. debuggers, profilers
- *Automatically* pick good inference for model/query, e.g. like DB query planners.
- Storing uncertainty. Samples? Particles? Marginals?
"Epistemological Database"

Text docs → p(Entity Mentions) → Relation Extraction → p(Relation Mentions) → Resolution (Coref) → p(Entities, Relations) → Structured Data → evidence → Human Edits → evidence → query → KB

Inference constantly bubbling in background...

Evidence

Samples p("truth")

Inference worker  Inference worker  Inference worker  Inference worker  Inference worker  Inference worker  Inference worker

answer
Summary

• Epistemological DBs
  - “entities & relations inferred from evidence”

• Research ingredients
  - SampleRank
  - Hierarchical coref, parallel/distributed
  - Human edits
  - PrDB of “universal schema”
  - Probabilistic programming

BTW: I’m currently looking for a post-doc.
Ingredients of our Approach

1. **Epistemological Database**
   - evidence from outside; truth discovery inside

2. **Human Edits as Evidence**
   - joint interpretation of *edits* with text & tables

3. **Never Ending Inference**
   - effects of new evidence propagate always

4. **Coreference as the Foundation**
   - all semantics as similarity including to ontologies; no fixed ontology

5. **Resource-bounded Information Gathering**
   - decision-theoretic approach to focussed KB filling

6. **Smart parallelism**
   - integrated with inference, asynchronous