Automated knowledge base construction

4. Taxonomy induction + entity disambiguation

Simon Razniewski
Summer term 2022
Outline

1. Taxonomy induction
2. Entity disambiguation
Recap: Entity types

- Einstein: Physicist, Nobel prize winner, ETH alumni
- Dudweiler: Village, municipality
- RCH$_2$OH: chemical formula, psychoactive substance

Why organize them?
- Observations are usually sparse
- Upper classes may be needed for queries:
  - German locations ending in -weiler
  - Scientists born in 1879
- Class relations needed for constraint checking
  - graduatedFrom(Person, educationalInstitution)
  - UdS -> University -> OK?
Taxonomy induction: Goal
Taxonomy induction: General approach

• Hypernymy candidates are “cheap”
  → Start with large noisy candidate graph, clean it up
Candidates #1: Hearst-patterns

- Hearst-style patterns (below: WebIsALOD for Frodo)
Candidates #2: Sub-category relations in Wiki systems

Categories:
- Featured articles
- Characters
- Cleanup
- Hobbits
- Baggins
- Ring bearers
- Elf friends
- Fellowship members
- Major characters (The Lord of the Rings)
- The Lord of the Rings Characters
- Characters that have appeared in the Hobbit and the Lord of the Rings
- The Hobbit: An Unexpected Journey Characters
- Bearers of the One Ring
- The Lord of the Rings: The Fellowship of the Ring (film) Characters
- The Lord of the Rings: The Two Towers (film) Characters
- The Lord of the Rings: The Return of the King (film) Characters

Categories:
- The Lord of the Rings characters
- Middle-earth Hobbits
- Adventure film characters
- Fictional orphans
- Bearers of the One Ring
- Fictional characters who can turn invisible
- Fictional characters introduced in 1954
- Fictional swordsmen
- Fictional amputees
- Fictional writers

Hidden categories:
- Commons category link is on Wikidata
- Categories requiring diffusion
Challenges

- Noise
  - Meta-categories
  - Ambiguous terms
- Structural oddities
  - Cycles
  - Upward branching
  - Redundancy (transitive edges)
- Imbalance in observations and scoring
  - Score-based thresholding discards entire regions
Text-based taxonomy induction challenge [Semeval 2016, Bordea et al.]

- Input: Set of domain terms
  - Tofu, pizza, garlic
  - Computer, smartphone, printer

- Task: Induce a taxonomy over these terms

- Potential evaluation measures
  - #nodes
  - #edges
  - Acyclicity
  - Recall w.r.t. gold standard
  - Precision w.r.t. gold standard
  - Connectedness (#connected components / #c.c)
  - Categorization (#intermediate nodes / #i.i)
Taxi [Panchenko et al., 2016]

1. **Crawl domain-specific text corpora in addition to WP, Commoncrawl**

2. **Candidate hypernymy extraction**
   1. Via substrings
      - “biomedical science” isA “science”
      - “microbiology” isA “biology”
      - “toast with bacon” isA “toast”
      - Lemmatization, simple modifier processing
      - Scoring proportional to relative overlap
   2. **Candidate hypernymy from 4 Hearst-Pattern extraction works**

3. **Supervised pruning**
   1. Positive examples: gold data
   2. Negative examples: inverted hypernyms + siblings
   3. Features: Substring overlap, Hearst confidence (more features did not help)
4. Taxonomy induction
   • Break cycles by random edge removal
   • Fix disconnected components by attaching each node with zero outdegree to root

- Too many hypernyms in English
Taxonomy induction using hypernym subsequences [Gupta et al., 2017]

- Looking at edges in isolation ignores important interactions
  - Hypernym candidates typically contain higher-level terms that help in predicting whole sequence
  - Crucial as abstract term hypernym extraction empirically harder (e.g., “company” → “group of friends”?)

<table>
<thead>
<tr>
<th>Candidate hypernym</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>5536</td>
</tr>
<tr>
<td>fruit</td>
<td>3898</td>
</tr>
<tr>
<td>apple</td>
<td>2119</td>
</tr>
<tr>
<td>vegetable</td>
<td>928</td>
</tr>
<tr>
<td>orange</td>
<td>797</td>
</tr>
<tr>
<td>tech company</td>
<td>619</td>
</tr>
<tr>
<td>brand</td>
<td>463</td>
</tr>
<tr>
<td>hardware company</td>
<td>460</td>
</tr>
<tr>
<td>technology company</td>
<td>427</td>
</tr>
<tr>
<td>food</td>
<td>370</td>
</tr>
</tbody>
</table>

Candidate hypernyms for the term apple.
Taxonomy induction using hypernym subsequences [Gupta et al., 2017]

• **Joint probabilistic model** that estimates true hypernymy relations from skewed observations

• Break cycles by removing edges with minimal weight

• Induce tree from DAG by a **min-cost-flow model**
Taxonomy induction using hypernym subsequence\[Gupta et al., 2017\]

- **Method**: Find cheapest way to send flow from leaves to root
- **Cost inversely proportional to edge weight**
Wiki[pedia|a]-based taxonomy induction: TiFi [Chu et al., WWW 2019]

Observations:
- Wikia category systems are noisy
- Wikia category systems lack abstractions

Approach: Supervised filtering + WordNet reuse
TiFi: Category cleaning

**Challenge:**
- Meta-categories (Meta, Administration, Article_Templates)
- Contextual categories (actors, awards, inspirations)
- Instances (Arda, Mordor)
- Extensions (Fan fiction)

**Approach: Supervised classification**
- “Featurizes” earlier rule-based category cleaning works, e.g., Marius Pasca at Google

**Features:**
- **Lexical**
  - Meta string dictionary (manual)
  - Headword in plural? Dark Orcs, Ring of Power
  - Capitalization? Quenya words, Ring bearers
- **Graph-based**
  - #instances
  - Supercategory/subcategory count
  - Average depth
  - Connected subgraph size
TiFi: Category cleaning - results

- Most important feature: Plural
  - Occasional errors (Food)
TiFi: Edge cleaning

• Challenge:
  • Type mismatches
    • Frodo → The Shire
    • Boromir → Death in Battle
    • Chieftains of the Dúnedain → Dúnedain of the North

• Approach: Supervised classification
  • Combination of lexical, semantic and graph-based features
TiFi: Edge cleaning - features

- **Lexical**
  - Head word generalization (c \textit{subclassOf} d?)
  - \(\text{head}(c) + \text{post}(c) = \text{head}(d) + \text{post}(d)\) and \text{pre}(d) in \text{pre}(c)
  - \(\text{pre}(c) + \text{head}(c) = \text{pre}(d) + \text{head}(d)\) and \text{post}(d) in \text{post}(c)
  - Only plural parents?

- **Semantic**
  - WordNet hypernym relations
  - Wikidata hypernym relations
  - Text matches
    - Wikia first sentence Hearst
      - \textit{Haradrim}: The Haradrim, known in Westron as the Southrons, were a \textit{race} of Men from Harad in the region of Middle-earth.
    - WordNet synset headword
      - Ex: Werewolves: a \textit{monster} able to change appearance from human to wolf and back again
  - Distributional similarity
    - WordNet graph distance (Wu-Palmer score)
    - Directional embedding scores (HyperVec — directional interpretation of embeddings)
      - Distributional inclusion hypothesis: flap is more similar to bird than to animal
      - Hypernyms occur in more general contexts

- **Graph-based**
  - \#common children
  - Parent.\#children/parent.avg-depth
TiFi - WordNet synset headword

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: castle

Display Options: 
Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Noun

- **S: (n) palace, castle** (a large and stately mansion)
- **S: (n) castle** (a large building formerly occupied by a ruler and fortified against attack)
- **S: (n) castle, rook** ((chess) the piece that can move any number of unoccupied squares in a direction parallel to the sides of the chessboard)
- **S: (n) castle, castling** (interchanging the positions of the king and a rook)
Algorithm 1: WordNet Synset Linking

Data: A category \( c \)

Result: WordNet synset \( s \) of \( c \)

\( c = \text{pre} + \text{head} + \text{pos}, l = \text{null}; \)

\( l = \text{list of WordNet synset candidate for} \ c; \)

if \( l = \text{null} \) then

\( l = \text{list of WordNet synset candidates for} \ \text{pre} + \text{head}; \)

if \( l = \text{null} \) then

\( l = \text{list of WordNet synset candidates for} \ \text{head}; \)

if \( l = \text{null} \) then

\( \text{return} \ \text{null}; \)

\( \text{max} = 0, s = \text{null}; \)

for all WordNet synset \( s_i \) in \( l \) do

\( \text{sim}(s_i, c) = \cosine(V_{s_i}, V_c) \) with \( V \): context vector;

\( \text{sim}(s_i, c) = \text{sim}(s_i, c) + 1/(2R_{s_i}) \) where \( R \): rank in WordNet;

if \( \text{sim}(s_i, c) > \text{max} \) then

\( \text{max} = \text{sim}(s_i, c); \)

\( s = s_i; \)

return \( s; \)
TiFi: Edge cleaning - results

<table>
<thead>
<tr>
<th>Method</th>
<th>Universe</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>HyperVec [31]</td>
<td>LoTR</td>
<td>0.82</td>
<td>0.8</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>GoT</td>
<td>0.83</td>
<td>0.81</td>
<td>0.82</td>
</tr>
<tr>
<td>HEAD [16]</td>
<td>LoTR</td>
<td>0.85</td>
<td>0.83</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>GoT</td>
<td>0.81</td>
<td>0.78</td>
<td>0.79</td>
</tr>
<tr>
<td>TiFi</td>
<td>LoTR</td>
<td>0.83</td>
<td>0.98</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>GoT</td>
<td>0.83</td>
<td>0.91</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Table 4: Step 2 - In-domain edge cleaning.

- Most important features:
  - Only plural parent
  - Lexical generalization
  - Common child support
  - Page type matching
TiFi: Top-level construction

- Problem: Wikia categories represent many disconnected components
- Solution: Link sinks to WordNet taxonomy and import further top level
TiFi — Top-level construction

• Using same algorithm as for linking in edge cleaning
  • *Birds* is mapped to *bird*%1:05:00::
    Subsequent hypernyms: *wn_vertebrate* → *wn_chordate* → *wn_animal* → *wn_organism* → *wn_living_thing* → *wn_whole* → *wn_object* → *wn_physical_entity* → *wn_entity*
  • **Removal of long paths** (nodes with only one child and one parent)
  • **Dictionary-based filtering** of ~100 too abstract classes (whole, sphere, imagination, …)
TiFi: Top-level construction - results

<table>
<thead>
<tr>
<th>Universe</th>
<th>#New Types</th>
<th>#New Edges</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoTR</td>
<td>43</td>
<td>171</td>
<td>0.84</td>
</tr>
<tr>
<td>GoT</td>
<td>39</td>
<td>179</td>
<td>0.84</td>
</tr>
<tr>
<td>Starwars</td>
<td>373</td>
<td>3387</td>
<td>0.84</td>
</tr>
<tr>
<td>Simpsons</td>
<td>115</td>
<td>439</td>
<td>0.92</td>
</tr>
<tr>
<td>World of Warcraft</td>
<td>257</td>
<td>2248</td>
<td>0.84</td>
</tr>
<tr>
<td>Greek Mythology</td>
<td>22</td>
<td>76</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table 7: Step 3 - WordNet integration.
TiFi – Relevance for entity search

Table 12. Example queries and results for the entity search evaluation.

<table>
<thead>
<tr>
<th>Query</th>
<th>Text</th>
<th>Structured Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dragons in LOTR</td>
<td>Glaurung, Túrin, Turambar, Eärendil, Smaug, Ancalagon</td>
<td>Urgost, Long-worms, Gosti, Dragoth the Dragon Lord, Cave-Drake, War of the Dwarves and Dragons, Dragon-spell, Stone Dragons, Fire-drake of Gondolin, Spark-dragons, Were-worms, Summoned Dragon, Fire-drakes, Glaurung, Ancalagon, Dragons, Cold-dragons, Sea-serpents, User blog: Alex Li-oo/e/ Kaldbrache the Dragon, Smaug, Dragon (Games, Workshop), Drake, Scatha, The Fall of Erebor</td>
</tr>
<tr>
<td>Which Black Numenoreans are servants of Morgoth</td>
<td>-</td>
<td>Black Númenórean</td>
</tr>
<tr>
<td>Which spiders are not agents of Saruman?</td>
<td>-</td>
<td>Shelob, Spider, Queen and Swarm, Samanthra, Spiderling, Great Spiders, Wicked, Wild, and Wrath</td>
</tr>
</tbody>
</table>

Table 11: Avg. #Answers and precision of entity search.

<table>
<thead>
<tr>
<th>Query</th>
<th>Text</th>
<th>Structured Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Google</td>
<td>Wikia</td>
</tr>
<tr>
<td>$t$</td>
<td>2 (52%)</td>
<td>7 (65%)</td>
</tr>
<tr>
<td>$t_1 \cap t_2$</td>
<td>1 (23%)</td>
<td>2 (11%)</td>
</tr>
<tr>
<td>$t_1 \setminus t_2$</td>
<td>1 (20%)</td>
<td>4 (36%)</td>
</tr>
<tr>
<td>Average</td>
<td>1 (32%)</td>
<td>4 (37%)</td>
</tr>
</tbody>
</table>
Open: Taxonomy Merging

~Complex alignment problem requiring joint optimization
Summary: Taxonomy induction

• Usually a **filtering process** on larger candidate set

• **Structure matters** for local decisions

• Local-only decision OK but not optimal

• Top-level situation
  • Sparse observations
  • Generality makes reuse easier

• **Relevance for AKBC:**
  • Queries for type conditions not explicitly observed
  • Constraints on relation arguments
Outline

1. Taxonomy induction
2. Entity disambiguation
Ready for fact extraction?

Homer is the main character of the TV series “Simpsons”.

Homer is the author of the Odyssey.

appearsIn(Homer, Simpsons)
wrote(Homer, Odyssey)?
Inputs

- Semi-Structured Data (Infoboxes, Tables, Lists ...)
- Premium Sources (Wikipedia, IMDB, ...)
- Web collections (Web crawls)
- Text Documents & Web Pages
- Conversations & Behavior
  - Online Forums & Social Media
  - Queries & Clicks

Methods

- Rules & Patterns
- Logical Inference
- Statistical Inference
- NLP Tools
- Deep Learning

Outputs

- Entity Names, Aliases & Classes
- Entities in Taxonomy
- Rules & Constraint
- Relational Statements
- Canonicalized Statements
Def: Disambiguation

Given an ambiguous name in a corpus and its meanings, disambiguation is the task of determining the intended meaning.
Also called “Wikification”, because everyone links to Wiki[pedia] data

Disambiguation

Usually Named Entity Recognition is to map the names to entities in the Knowledge Base.

NER’ed corpus

Homer eats a doughnut.
Def: Context of a word

The context of a word in a corpus is the multi-set of the words in its vicinity without the stopwords.

(The definition may vary depending on the application)

Homer eats a doughnut.

Context of "Homer": {eats, doughnut}
Def: Context of an entity

The context of an entity in a KB is the set of all labels of all entities in its vicinity.

(The definition may vary depending on the application)

Context of Homer: {doughnut, USA, America}
Def: Context-based disambiguation

Context-based disambiguation (also: bag of words disambiguation) maps a name in a corpus to the entity in the KB whose context has the highest overlap to the context of the name.

For USA Today, Homer is among the top 25 most influential people of the past 25 years.

Who wins?
What if there is little context?

This is very important for the Simpsons.

?  

Simpsons

The Robert Simpson Department Store.
Defunct since 1990.
**Def: Disambiguation Prior**

A disambiguation prior is a mapping from names to their meanings, weighted by the number of times that the name refers to the meaning in a reference corpus.

Can be computed e.g. from Wiki[pedia] by link disambiguation or page views.
Local or global solution?

• Features so far local (one entity mention at a time)
  • Context-similarity
  • Disambiguation prior

• Do disambiguations influence each other?
Def: Coherence Criterion

The Coherence Criterion postulates that entities that are mentioned in one document should be related in the KB.

Bart and Homer accidentally launch a rocket into the Springfield church, causing Lisa to leave Christianity.
Possible implementation (2)

Bart and Homer accidentally launch a rocket into the Springfield church, causing Lisa to leave Christianity.

$n$ entity mentions
Each with $m$ candidate KB entities

→ Compute coherence scores for $m^n$ combinations
Possible implementation (2)

Bart and Homer accidentally launch a rocket into the Springfield church, causing Lisa to leave Christianity.

1. Consider all entities with that name
2. Weight each entity by features (bag of words, prior, etc.)
3. Start a random walk in the KB
State of the art

- **Pre-trained neural models** again
  - Encode KB context
  - Encode text context
  - Predict match likelihood
  - ...or, predict KB identifier directly (GENRE, de Cao, ICLR 2021)

- **Automated training data**: Wikidata text links
Example systems (1):
Opentapioca

https://opentapioca.org/
Example systems (2): AIDA

Explicit parameter tuning — no more functioning 😞
https://gate.d5.mpi-inf.mpg.de/webaida/
Further solutions

• spaCy can do this
  • https://spacy.io/usage/linguistic-features#entity-linking
  • Though more complex setup, KB

• Commercial APIs
  • https://try.rosette.com/
  • https://cloud.google.com/natural-language/docs/analyzing-entities
  • https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/
Summary: Disambiguation

We saw 3 indicators for disambiguation:

1. Context
   
   Homer eats a doughnut.

2. Disambiguation prior

3. Coherence
Disambiguation vs. mention typing

• Like for typing, **context is decisive**

• Unlike typing, **no chance for supervised approach**
  • Can train classifiers that predict “Politician-ness” of a mention
  • Cannot train classifier to predict “Einstein-ness”

• **Disambiguation is ranking problem** (single solution), not multiclass classification

• **Type predictions can be used as intermediate features for context-based disambiguation**

• **Type prediction can augment disambiguation**, if KB has sparse content
References


• Slides adapted from Fabian Suchanek, Gina-Anne Levow and Chris Manning
Assignment 5 – Taxonomy induction

• Given: Set of terms
• Task: Build a small taxonomy that organizes them
  • Can become both leaves or inner nodes
• Noisy input provided from WebIsALOD
  • Cleaning, filtering, etc. highly recommended
  • Other inputs allowed too
• Evaluation:
  • Two known term sets
  • One unseen set (robustness)
Take home

• **Taxonomy induction:**
  • Structure matters
  • Important features: Lexical/semantic matches, structural properties

• **Entity disambiguation**
  • Context seen already in typing
  • Coherence as additional feature

• **Meta-observation:**
  • Both problems are better approached globally than locally
  • Both problems are complementary
Playing with the Wikidata taxonomy

https://angryloki.github.io/wikidata-graph-builder/?property=P279&item=Q74359