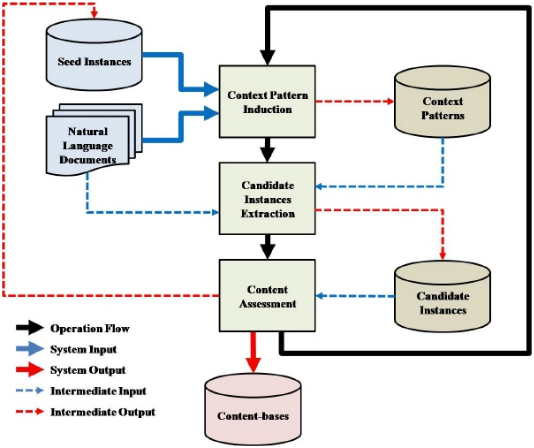
Information extraction: Conceptual hierarchies and relations

Vivi Nastase

with material from Marius Pasca's CIKM-2011 tutorial on IE

Summer semester 2012, ICL, University of Heidelberg

Bootstrapping in general – reminder



Seokhwan Kim et al., 2011 *Semi-supervised Information Extraction*

Bootstrapping for relation extraction

Start either with a non-empty set $S = (n_{i1}, n_{i2})$ of seed pair examples or a non-empty set P of patterns (let's assume examples):

1. find all occurrences of the examples (n_{i1}, n_{i2}) in the text collection
2. extract [and rank] patterns joining the terms in each pair: $n_{i1} w_1 ... w_k n_{i2}$
3. add the [highest ranking] extracted patterns to P
4. use the patterns in P to find additional pairs
5. add the [highest ranking] extracted pairs to S, go to step 1

Extracting taxonomical relations

Hearst, 1992: *Automatic acquisition of hyponyms from large text corpora*

NP *such as* NP, NP, ...	The bow lute, *such as* the Bambara ndang ...
such NP *as* {NP,}* {(*or*|*and*)} NP	... works by *such* authors *as* Herrick, Goldsmith, and Shakespeare.
NP{, NP}*{, (*or*|*and*) *other* NP	... temples, treasuries, *and other* important civic buildings
NP{,} (*including*|*especially*) {NP,}* (*or*|*and*) NP	... most European countries, *especially* France, England *and* Spain.

Adding pattern evaluation

Brin, 1998 *Extracting patterns and relations from the World Wide Web*

$$specificity(p) \approx -log(P(X \in M_D(p)))$$

$M_D(p)$ is the set of tuples that match the pattern p in the document set D, and X is a random variable uniformly distributed over the domain of tuples for the mined relation R. (In practice, the specificity of the pattern is measured based on the length of the pattern.)
And different style patterns:

URL Pattern	Text Pattern
www.sff.net/locus/c.*	<LI><B>*title*</B> by *author*
dns.city-net.com/ lmann/awards/hugos/1984.html	<i>*title*</i> by *author*

Conceptual hierarchies

Kozareva & Hovy, 2010 *A semi-supervised method to learn and construct taxonomies using the Web*

1. a semi-supervised algorithm that learns hyponym-hypernym pairs subordinated to a root concept
2. Web-based concept positioning procedure used to validate extracted relations
3. a graph algorithm that derives the taxonomy

Extracting hyponym-hypernym relations

Input
: root concept for the target hierarchy, specified as one-seed instance:
lions for **animals**, **cucumbers** for **plants**, ...

Data source
: Web documents

Extracting hyponym-hypernym relations – steps

gather hyponyms

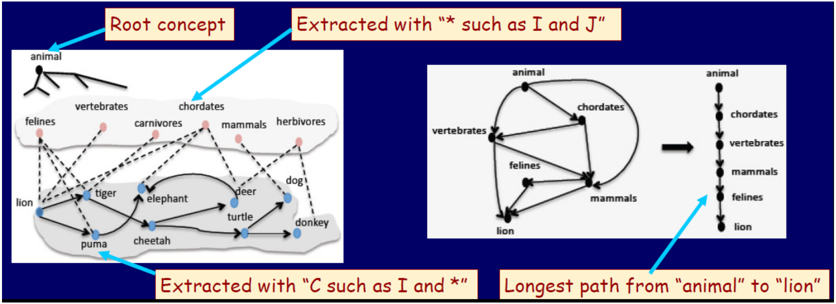
1. fill in extraction pattern **C such as I and *** from known pairs
2. convert patterns to queries, fetch Web documents
3. gather all terms that instantiate *
4. if new terms have been extracted, go to step 1

gather hypernyms

1. filter concepts based on $outDegree(v) = \frac{\sum_{(v,x)} w(v,x)}{|V|-1}$
2. fill in pattern *** such as t_1 and t_2**
3. convert pattern to query, fetch documents
4. gather all terms that instantiate *
5. rank terms by $inDegree = \sum_{(t_1 - t_2, h)} w(t_1 - t_2, h)$

Organize extracted pairs into a hierarchy

1. for each pair, determine the most specific concept – based on instantiated pattern counts **X such as Y**, **X including Y**
2. eliminate edge cycles and transitive closures



Issues in relation extraction

concepts : what terms to link

relation types : what types of relations to target

- *is-a* (taxonomical relations)
- *part-of*
- other relations

Learning from infoboxes

Dr. Henry Walton "Indiana" Jones, Jr., Ph.D.[12] is a fictional character and the protagonist of the *Indiana Jones* franchise. George Lucas and Steven Spielberg created the character in homage to the action heroes of 1930s film serials. The character first appeared in the 1981 film *Raiders of the Lost Ark*, to be followed by *Indiana Jones and the Temple of Doom* in 1984, *Indiana Jones and the Last Crusade* in 1989, *The Young Indiana Jones Chronicles* from 1992 to 1996, and *Indiana Jones and the Kingdom of the Crystal Skull* in 2008. Alongside the more widely known films and television programs, the character is also featured in novels, comics, video games, and other media. Jones is also featured in the theme park attraction *Indiana Jones Adventure*, which exists in similar forms at Disneyland and Tokyo DisneySea.

Jones is most famously played by Harrison Ford and has also been portrayed by River Phoenix (as the young Jones in *The Last Crusade*), and in the television series *The Young Indiana Jones Chronicles* by Corey Carrier, Sean Patrick Flanery, and George Hall. Doug Lee has supplied Jones's voice to two LucasArts video games, *Indiana Jones and the Fate of Atlantis* and *Indiana Jones and the Infernal Machine*, while David Esch supplied his voice to *Indiana Jones and the Emperor's Tomb*.

Particularly notable facets of the character include his iconic look (bullwhip, fedora, and leather jacket), sense of humor, deep knowledge of many ancient civilizations and languages, and fear of snakes.

Indiana Jones remains one of cinema's most revered movie characters. In 2003, he was ranked as the second greatest movie hero of all time by the American Film Institute.[13] He was also named the sixth greatest movie character by *Empire* magazine.[14] *Entertainment Weekly* ranked Indy 2nd on their list of *The All-Time Coolest Heroes in Pop Culture*.[15] *Premiere* magazine also placed Indy at number 7 on their list of *The 100 Greatest Movie Characters of All Time*.[16] Since his first appearance in *Raiders of the Lost Ark*, he has become a worldwide star. On their list of the *100 Greatest Fictional Characters*, Fandomania.com ranked Indy at number 10.[17] In 2010, he ranked #2 on *Time* Magazine's list of the greatest fictional characters of all time, surpassed only by Sherlock Holmes.[*citation needed*]

Contents [hide]

Henry Jones, Jr.

Indiana Jones character

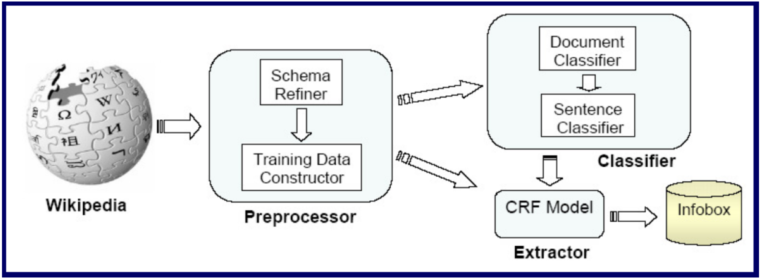
Harrison Ford as Indiana Jones in *Raiders of the Lost Ark*

First appearance	*Raiders of the Lost Ark*
Created by	George Lucas Steven Spielberg
Portrayed by	**Films:** Harrison Ford (ages 36–58) River Phoenix (age 13) **TV series:**

- they provide examples of relations of interest
- the associated articles provide (free and annotated!) training for these relations
- (reused) infobox templates

Creating missing infoboxes

Wu & Weld, 2007 *Autonomously semantifying Wikiepdia*



Creating missing infoboxes

- Preprocessor
  - identify relevant attributes from articles with the same infobox template
  - generate training data for classification and extraction
- Document classifier
  - determine whether an article belongs to a certain class
  - one classifier per class of articles
- Sentence classifier
  - determine whether a sentence contains the value of an attribute
  - one classifier per attribute per infobox template
- Extractors
  - extract a value from a (marked) sentence
  - aggregate across sentences, return values for attributes

Sentence classifier

Training data

For each article with an infobox:

1. split document in sentences
2. for each attribute value find a (unique) correspoding sentence in the article (positive training example)
3. take other sentences as negative training examples

Features

- sentence tokens
- tokens' POS tags

Multi-class classification – Maximum Entropy model

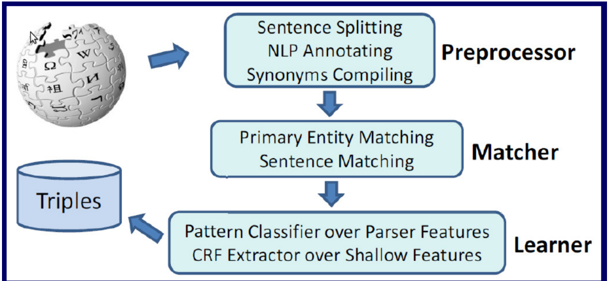
Learning extractors

Feature Description	**Example**
First token of sentence	*Hello* world
In first half of sentence	*Hello* world
In second half of sentence	Hello *world*
Start with capital	Hawaii
Start with capital, end with period	Mr.
Single capital	A
All capital, end with period	CORP.
Contains at least one digit	AB3
Made up of two digits	99
Made up of four digits	1999
Contains a dollar sign	20$
Contains an underline symbol	km_square
Contains an percentage symbol	20%
Stop word	the; a; of
Purely numeric	1929
Number type	1932; 1,234; 5.6
Part of Speech tag	
Token itself	
NP chunking tag	
String normalization: capital to "A", lowercase to "a", digit to "1", others to "0"	$TF-1 \Longrightarrow AA01$
Part of anchor text	Machine *Learning*
Beginning of anchor text	*Machine* Learning
Previous tokens (window size 5)	
Following tokens (window size 5)	
Previous token anchored	Machine *Learning*
Next token anchored	*Machine* Learning

Moving to the Web through Wikipedia

Wu & Weld, 2010 *Open information extraction using Wikipedia*

- **Data**
  - Wikipedia articles for acquiring positive examples
  - Web document for finding new relation instances
- **Output**: relational tuples (**Arg1**-**relation**-**Arg2**)



(Courtesy F. Wu)

Extraction components

Preprocessing Wikipedia articles

- sentence splitting
- POS tagging
- syntactic parsing

Infobox entries matcher

- find sentences that contains the article title (**Arg1**) and the value of the infobox attribute (**Arg2**)
- apply filters and heuristics to improve matching accuracy

Extraction components – continued

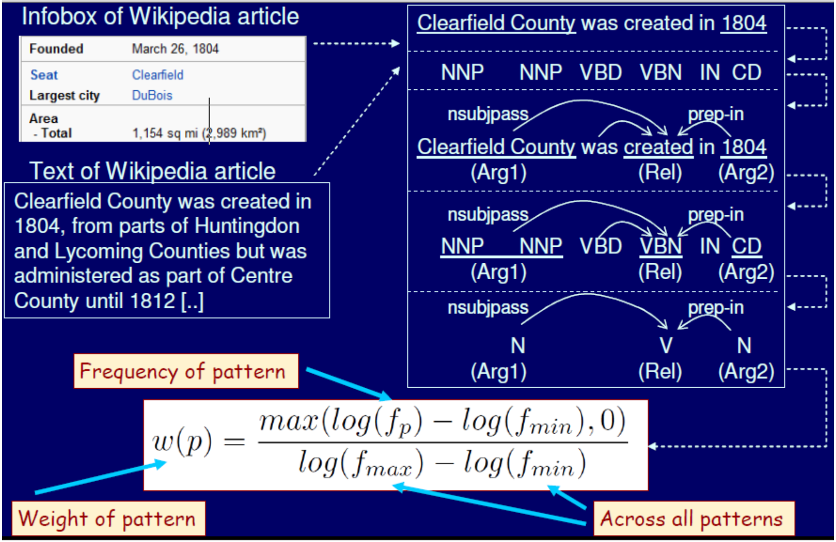
Learner

deep
- extract the syntactic path that connects **Arg1** and **Arg2** from each matching sentence
- collect and generalize unlexicalized patterns

shallow collect and generalize POS and lexical context

exploit deep (with parsing) and shallow (no parsing) patterns to extract tuples from Web documents

Learning patterns from Wikipedia



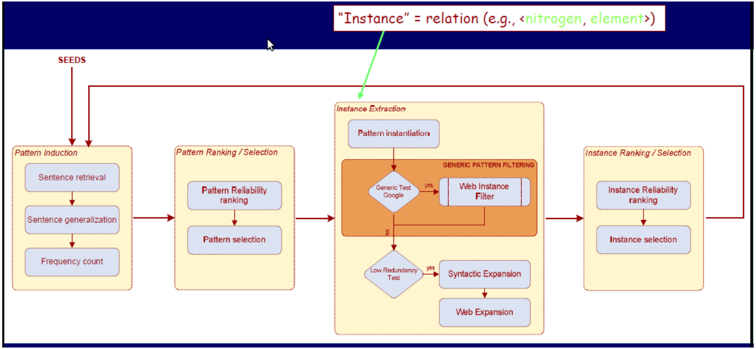
$$w(p) = \frac{max(log(f_p) - log(f_{min}), 0)}{log(f_{max}) - log(f_{min})}$$

Relation extraction from the Web

Pantel & Pennacchiotti, 2006 *Espresso: Leveraging generic patterns for automatically harvesting semantic relations*

- **Input**: target relation, as small sets of seed pairs
  - (**nitrogen**, **element**), (**wheat**, **crop**) for **IsA**
  - (**city**, **region**), (**hand**, **body**) for **PartOf**
- **Data sources**: corpora / Web documents
- **Output**: ranked lists of relations
- **Approach**: bootstrapping

Relation extraction from the Web



(Courtesy Pantel & Pennacchiotti)

Pattern Induction

Sentence retrieval

- match input seed relations to sentences

Sentence generalization

"Because/IN **HF/NNP** is/VBZ a/DT **weak/JJ acid/NN** and/CC ..."
"Because/IN **<TR>** is/VBZ a/DT **<TR>** and/CC ..."

Frequency count

- count frequency of occurrence of each pattern

Pattern ranking

Rank patterns according to reliability

Strength of association between pattern p and input relation i

Reliability of input relation i

$$r_\pi(p) = \frac{\sum_{i \in I'} \left(\frac{pmi(i,p)}{\max_{pmi}} * r_\iota(i) \right)}{|I'|}$$

Cardinality of set of input relations

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|}$$

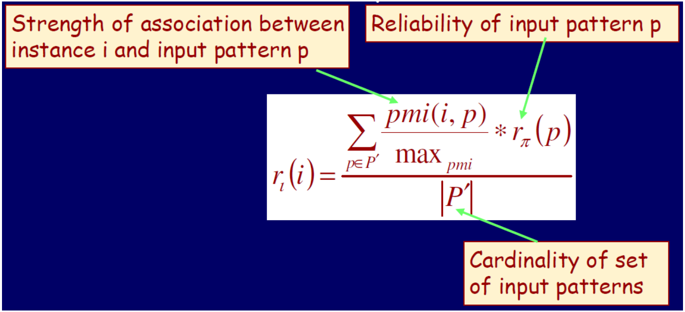
Relation extraction

- match patterns to sentences in the document collection
- if low-redundancy matches, expand:
  - convert patterns to queries:

$$\left.\begin{array}{c}(\textbf{italy},\textbf{country}) \\ C\ \textit{such as}\ I\end{array}\right\} \rightarrow \textbf{country}\ \textit{such as *}$$

$$(\textbf{european country}, \textbf{location}) \rightarrow (\textbf{country}, \textbf{location})$$

Rank extracted relations

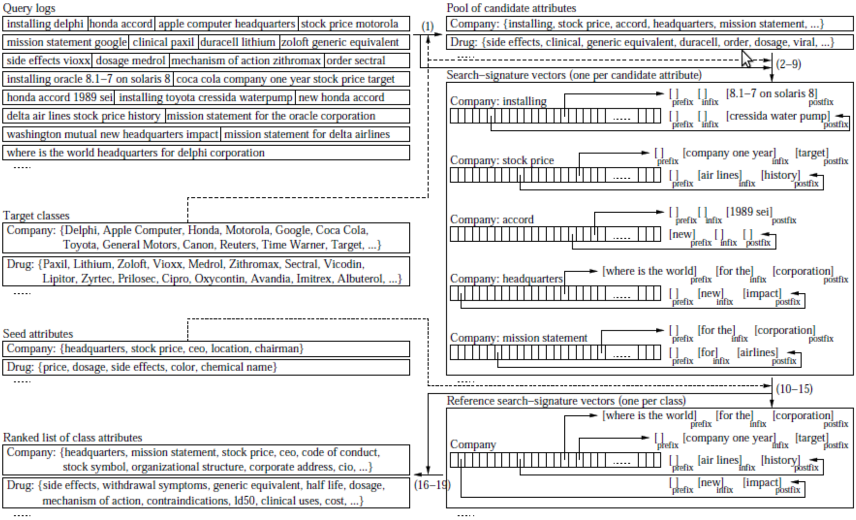


Select top relations

Fact extraction from queries

Pasca, 2007 *Organizing and searching the World Wide Web of facts - Step two: harnessing the wisdom of the crowds*

- **Input**
  - target classes, as sets of seeds: e.g. for **Company** – **Honda**, **Oracle**, **Reuters**, ...
  - seed attributes: e.g. for **Company** – **headquarters**, **stock price**, **ceo**, ...
- **Data** – anonymized search queries and their frequencies
- **Output** – ranked list of attributes, one per class



Query logs
installing delphi
honda accord
apple computer headquarters
stock price motorola
mission statement google
clinical paxil
duracell lithium
zoloft generic equivalent
side effects vioxx
dosage medrol
mechanism of action zithromax
order sectral
installing oracle 8.1−7 on solaris 8
coca cola company one year stock price target
honda accord 1989 sei
installing toyota cressida waterpump
new honda accord
delta air lines stock price history
mission statement for the oracle corporation
washington mutual new headquarters impact
mission statement for delta airlines
where is the world headquarters for delphi corporation
Target classes
Company: {Delphi, Apple Computer, Honda, Motorola, Google, Coca Cola, Toyota, General Motors, Canon, Reuters, Time Warner, Target, ...}
Drug: {Paxil, Lithium, Zoloft, Vioxx, Medrol, Zithromax, Sectral, Vicodin, Lipitor, Zyrtec, Prilosec, Cipro, Oxycontin, Avandia, Imitrex, Albuterol, ...}
Seed attributes
Company: {headquarters, stock price, ceo, location, chairman}
Drug: {price, dosage, side effects, color, chemical name}
Ranked list of class attributes
Company: {headquarters, mission statement, stock price, ceo, code of conduct, stock symbol, organizational structure, corporate address, cio, ...}
Drug: {side effects, withdrawal symptoms, generic equivalent, half life, dosage, mechanism of action, contraindications, ld50, clinical uses, cost, ...}
(1)
(2−9)
(10−15)
(16−19)
Pool of candidate attributes
Company: {installing, stock price, accord, headquarters, mission statement, ...}
Drug: {side effects, clinical, generic equivalent, duracell, order, dosage, viral, ...}
Search−signature vectors (one per candidate attribute)
Company: installing
[ ]prefix [ ]infix [8.1−7 on solaris 8]postfix
[ ]prefix [ ]infix [cressida water pump]postfix
Company: stock price
[ ]prefix [company one year]infix [target]postfix
[ ]prefix [air lines]infix [history]postfix
Company: accord
[ ]prefix [ ]infix [1989 sei]postfix
[new]prefix [ ]infix [ ]postfix
Company: headquarters
[where is the world]prefix [for the]infix [corporation]postfix
[ ]prefix [new]infix [impact]postfix
Company: mission statement
[ ]prefix [for the]infix [corporation]postfix
[ ]prefix [for]infix [airlines]postfix
Reference search−signature vectors (one per class)
Company
[where is the world]prefix [for the]infix [corporation]postfix
[ ]prefix [company one year]infix [target]postfix
[ ]prefix [air lines]infix [history]postfix
[ ]prefix [new]infix [impact]postfix

Extraction from queries

1. select candidate attributes from queries containing an instance
2. create internal representation of candidate attributes, from queries containing an instance and a candidate attribute
3. rank candidate attributes, from similarity between internal representation of a candidate attribte and combined internal representation of all seed attributes

Example attributes

actor	awards, height, age, date of birth, weight, ...
aircraft model	weight, length, history, fuel consumption, ...
award	recipients, date, winners list, result, gossip, printable ballot ...
basic food	calories, color, size, allergies, taste, carbs, nutritional information, ...
...	

Extreme knowledge acquisition

Davidov & Rappaport, 2008 *Unsupervised Discovery of Generic Relationships Using Pattern Clusters and its Evaluation by Automatically Generated SAT Analogy Questions*

Idea

- CW: content words – frequency $< F_C$
- HFW: high frequency words – frequency $> F_H$

  [Prefix] CW_1 [Infix] CW_2 [Postfix]
- Prefix, Infix, PostFix $\sim HFW+$

Focusing the extraction

- one of CW_i is a hook (seed) word, the other is the target
- filter documents to those that contain the hook word (hook corpus)
- sort targets by PMI relative to the hook
- use various hook words

Pattern clustering

1. cluster patterns that share both CW_is
2. merge clusters that share x% of their patterns
3. remove patterns generated from a single hook corpus (force generality)
4. iteratively merge clusters by looking at shared patterns (P_{core})
5. remove clusters that don't share patterns (contain only P_{unconf})

Cluster labels – top 5 pairs according to:

$Hits(C, (w_1, w_2)) =$

$$\frac{|\{p; (w_1, w_2) \text{ appears in } p \in P_{core}\}|}{|P_{core}|} + \alpha \frac{|\{p; (w_1, w_2) \text{ appears in } p \in P_{unconf}\}|}{|P_{unconf}|}$$

Clusters and labels

such X as Y	(pets, dogs)
X such as Y	
Y and other X	
buy Y accessory for X!	(phone, charger)
shipping Y for X	

Next week: large scale knowledge acquisition from the web

- Never Ending Language Learning (NELL) (CMU)
- KnowItAll by Machine Reading the World Wide Web (UofW)