CMSC724: Information Extraction

Amol Deshpande

University of Maryland, College Park

April 18, 2013

Example: Answering Queries Over Text

For years, Microsoft Corporation CEO Bill Gates was against open source. But today he appears to have changed his mind. "We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

Select Name
From PEOPLE
Where Organization = 'Microsoft'

PEOPLE

Name	Title	Organization
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	Founder	Free Soft..

Bill Gates

Bill Veghte

(from William Cohen's IE tutorial, 2003)

Overview

- Goal: automatically extract structured information from unstructured text
- Applications:
  - News tracking, Customer care, Data cleaning, Classified ads, PIM, Citation databases, Opinion databases
- Evolution:
  - Early systems: rule-based with manually coded rules
  - Then: automatically learning rules from examples
  - Statistical learning
    - Generating models based on HMMs
    - Conditional models based on maximum entropy
    - Conditional random fields
    - ... and so on.

Overview

- Types of structure extracted
  - Entities
    - Named entities: names of persons, locations, companies
    - Disease names, protein names, paper titles, journal names
  - Relationships
    - Binary vs multi-way
  - Adjectives describing entities
  - Structures: lists, tables, ontologies

Overview

- Types of structure extracted
  - Entities
    - Named entities: names of persons, locations, companies
    - Disease names, protein names, paper titles, journal names
  - Relationships
    - Binary vs multi-way
  - Adjectives describing entities
  - Structures: lists, tables, ontologies
- Types of unstructured sources
  - Granularity of extraction: record/sentences vs paragraphs/documents
  - Heterogeneity
    - Machine generated pages: extractors often called *wrappers*
    - Partially-structure sources
    - Open-ended sources

Overview

- Input resources that are often available/used
  - Structured databases like ACM DL or DBLP
  - Labeled data
  - Preprocessing libraries: NLP tools

Overview

- Input resources that are often available/used
  - Structured databases like ACM DL or DBLP
  - Labeled data
  - Preprocessing libraries: NLP tools
- Challenges
  - Accuracy: Precision vs Recall
  - Efficiency

Entity Extraction: Rule-based

- Very useful for simple extraction tasks, and widely used
  - "Big Data" may make them even more viable today
- Typical rule-based system:
  - A collection of rules
  - Policies dictating how to use them
- Basically pattern-matching
  - With some context around it

Entity Extraction: Rule-based

Rule: TheGazOrganization
Priority: 50
// Matches “The <in list of company names>”
({Part of speech = DT | Part of speech = RB} {DictionaryLookup = organization})
→ Organization

Rule: LocOrganization
Priority: 50
// Matches “London Police”
({DictionaryLookup = location | DictionaryLookup = country} {DictionaryLookup
= organization} {DictionaryLookup = organization}?) → Organization

Rule: INOrgXandY
Priority: 200
// Matches “in Bradford & Bingley”, or “in Bradford & Bingley Ltd”
({Token string = “in”})
({Part of speech = NNP}+ {Token string = “&”} {Orthography type = upperInitial}+ {DictionaryLookup = organization end}?):orgName → Organization=:orgName

Rule: OrgDept
Priority: 25
// Matches “Department of Pure Mathematics and Physics”
({Token.string = “Department”} {Token.string = “of”} {Orthography type = upperInitial}+ ({Token.string = ”and”} {Orthography type = upperInitial}+)?) → Organization

Fig. 2.1 A subset of rules for identifying company names paraphrased from the Named Entity recognizer in Gate.

Hand Coded Rule Example: Conference Name

```
# These are subordinate patterns
$wordOrdinals="(?:first|second|third|fourth|fifth|sixth|seventh|eighth|ninth|tenth|eleventh|twelfth|thirtee
teenth|fifteenth)";
my $numberOrdinals="(?:\\d?(?:1st|2nd|3rd|1th|2th|3th|4th|5th|6th|7th|8th|9th|0th))";
my $ordinals="(?:$wordOrdinals|$numberOrdinals)";
my $confTypes="(?:Conference|Workshop|Symposium)";
my $words="(?:[A-Z]\\w+\\s*)"; # A word starting with a capital letter and ending with 0 or more spaces
my $confDescriptors="(?:international\\s+|[A-Z]+\\s+)"; # .e.g "International Conference ...' or the confere
name for workshops (e.g. "VLDB Workshop ...")
my $connectors="(?:on|of)";
my $abbreviations="(?:\\([A-Z]\\w\\w+[\\W\\s]*?(?:\\d\\d+)?\\))"; # Conference abbreviations like "(SIGMOD
# The actual pattern we search for.  A typical conference name this pattern will find is
# "3rd International Conference on Blah Blah Blah (ICBBB-05)"
my
$fullNamePattern="((?:$ordinals\\s+$words*|$confDescriptors)?$confTypes(?:\\s+$connectors\\s+.*?|\\s+
reviations?)(?:\\n|\\r|\\.|<)";
################################## ##################################
# Given a <dbworldMessage>, look for the conference pattern
#############################################################
lookForPattern($dbworldMessage, $fullNamePattern);
#############################################################
# In a given <file>, look for occurrences of <pattern>
# <pattern> is a regular expression
#############################################################
sub lookForPattern {
   my ($file,$pattern) = @_;
```

Entity Extraction: Rule-based

- Usually very large number of rules
  - May lead to conflicts/overlaps etc.
  - Often rules depend on each other (application of one rule enables another rule)
- Policies:
  - Specify how to resolve conflicts (largest match etc)
  - Order the rules
  - Encode the rules in a Finite State Machine

Entity Extraction: Rule-based

- How to learn rules?
  - Domain expert specified
  - Learn from a training dataset

(1) Rset = set of rules, initially empty.
(2) While there exists an entity $\mathbf{x} \in D$ not covered by any rule in Rset
    (a) Form new rules around $\mathbf{x}$.
    (b) Add new rules to Rset.
(3) Post process rules to prune away redundant rules.

- Issues:
  - How to create a new rule given the already existing rules
  - Different approaches – mostly heuristics

Popular Machine Learning Methods for IE

- Naive Bayes
- SRV [Freitag-98], Inductive Logic Programming
- Rapier [Califf & Mooney-97]
- Hidden Markov Models [Leek, 1997]
- Maximum Entropy Markov Models [McCallum et al, 2000]
- Conditional Random Fields [Lafferty et al, 2000]
  - Implementations available:
    - Mallet (Andrew McCallum)
    - crf.sourceforge.net (Sunita Sarawagi)
    - MinorThird minorthird.sourceforge.net (William Cohen)

For details: [Feldman, 2006 and Cohen, 2004]

Entity Extraction: Statistical Methods

- Token-based methods
  - Tokenize the sentences
  - For each token, try to assign it a label among a fixed set of labels Y

Here is my review of Fermat's last theorem by S. Singh

i	1	2	3	4	5	6	7	8	9	10	11
x	Here	is	my	review	of	Fermat's	last	theorem	by	S.	Singh
y	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}

R. Fagin and J. Helpbern, Belief Awareness Reasoning

i	1	2	3	4	5	6	7	8	9
x	R.	Fagin	and	J.	Helpbern	,	Belief	Awareness	Reasoning
y	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9

Fig. 3.1 Tokenization of two sentences into sequence of tokens.

- Define a set of features (with many features) —- $f_i(y, x, i) : y \in Y, x \in X$ —- e.g., $f_1(y, x, i)$ = [[x_i equals "Fagin"]] . [[y = Author]] —- e.g., $f_3(y, x, i)$ = [[x_i matches INITIAL_DOT]] . [[y = Author]] —- e.g., $f_5(y, x, i)$ = [[x_i in Person_dictionary]] . [[y = Author]]

Entity Extraction: Statistical Methods

- Token-based methods
- Assigning labels:
  - Basic option: Assign independently
    - Learn a classifier (e.g., SVM) using the training data
    - Won't exploit any correlations across tokens
  - Left-to-right:
    - Assign labels going from left-to-right
    - Use the label on left to predict the label on the right token
  - Conditional Random Fields (CRFs)
    - Widely used for this and other tasks
    - A special type of graphical model with tractable inference complexity

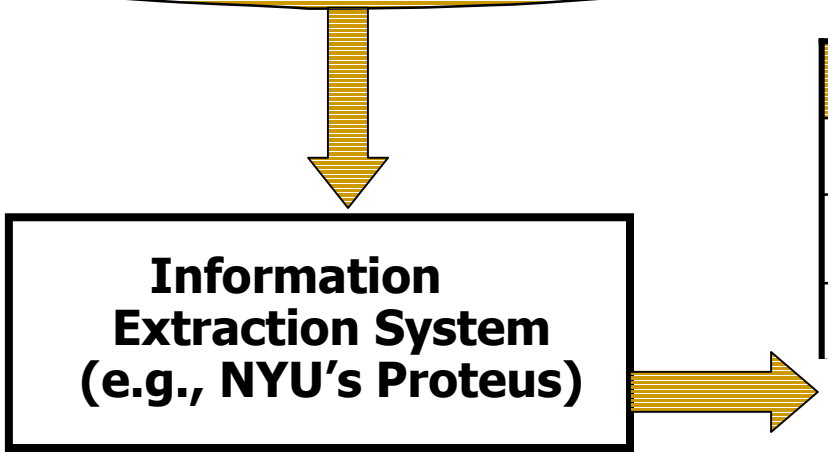
Entity Extraction: Statistical Methods

- Segment-based methods
  - Features defined over segments comprising multiple tokens
  - Segment-level features hard to capture in the token-based methods
    - e.g., $f(y_i, y_{i-1}, x, 3, 5)$ = [[$x_3 x_4 x_5$ appears in a list of journals]] . [[y_i = journal]]
    - e.g., $f(y_i, y_{i-1}, x, 3, 5)$ = MAX TF-IDF-similarly($x_3 x_4 x_5$, J) . [[y_i = journal]]

Relation Extraction: Disease Outbreaks

- **E**xtract structured relations from text

May 19 1995, Atlanta -- The Centers for Disease Control and Prevention, which is in the front line of the world's response to the deadly **Ebola** epidemic in **Zaire**, is finding itself hard pressed to cope with the crisis…



Disease Outbreaks in ***The New York Times***

Date	***Disease Name***	***Location***
Jan. 1995	Malaria	Ethiopia
July 1995	Mad Cow Disease	U.K.
Feb. 1995	Pneumonia	U.S.

Relation Extraction

- Typically require Entity Tagging as preprocessing

- Knowledge Engineering
  - Rules defined over lexical items
    - “<company> located in <location>”
  - Rules defined over parsed text
    - “((Obj <company>) (Verb located) (*) (Subj <location>))”
  - Proteus, GATE, …

- Machine Learning-based
  - Learn rules/patterns from examples
    Dan Roth 2005, Cardie 2006, Mooney 2005, …
  - Partially-supervised: bootstrap from “seed” examples
    Agichtein & Gravano 2000, Etzioni et al., 2004, …

- Recently, hybrid models [Feldman2004, 2006]