ECML-2003 Tutorial

Adaptive Text Extraction and Mining

Fabio Ciravegna
Department of Computer Science
University of Sheffield
F.Ciravegna@dcs.shef.ac.uk
www.dcs.shef.ac.uk/~fabio

Nicholas Kushmerick
Department of Computer Science
University College Dublin
nick@ucd.ie
www.cs.ucd.ie/staff/nick

What is IE

What can we extract from the Web and why?

The ‘canonical’ IE task

- **Input**:
  - Document
    - newspaper article, Web page, email message, …
  - Pre-defined “information need”
    - frame slots, template fillers, database tuples, …
- **Output**
  - The specific substrings/fragments of the document or labels that satisfy the stated information need, possibly organised in a template

- DARPA’s ‘**M***essage* **U***nderstanding* **C***onferences*/**C***ompetitions*’ since late1980’s; most recent: MUC-7, 1998.
- Recent interest in the machine learning and Web communities.

IE Standard Tasks

- Preprocessing
  - Tokenization
  - Morphological Analysis
  - Part of Speech Tagging
- Information Identification
  - Named Entity Recognition
  - Template Filling (from the MUC)
    - Template Elements
    - Template Relations
    - Scenario Template

NE Recognition & Coreference

& Coreference

Organisation

19:16 Moody’s rates Province of Saskatchewan A3

Moody's Investors Service Inc said it assigned an A3 rating to the Province of Saskatchewan's [redacted] bond offering that was priced today. [redacted]
The sale is a reopening of the province's 9.6 percent bonds due February 4, 2022. Proceeds will be used for government purposes, mainly Saskatchewan Power Corp.

%

Date

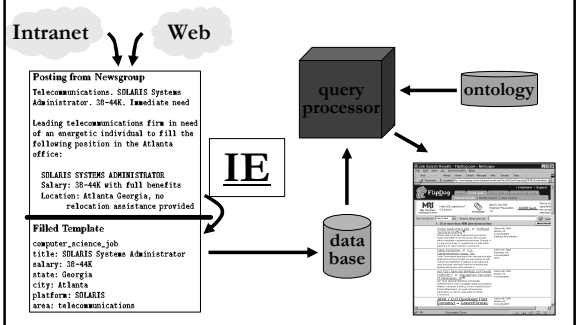
Template Filling

19:16 Moody's rates Province of Saskatchewan A3

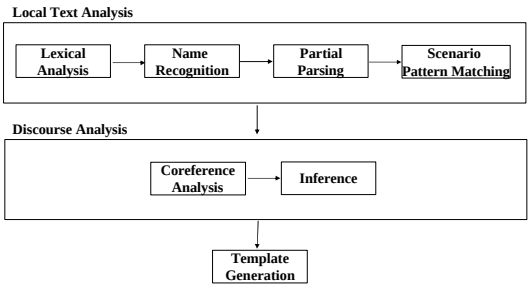
Moody's Investors Service Inc said it assigned an A3 rating to the Province of Saskatchewan's C$115 million bond offering that was priced today.
The sale is a reopening of the province's 9.6 percent bonds due February 4, 2022. Proceeds will be used for government purposes, mainly Saskatchewan Power Corp.

amount	C$115 million
issuer	Province of Saskatchewan
placement-date	today
maturity	February 4, 2022
rate	9.6 percent

The Big Picture



NYU Architecture: a MUC architecture



Semantic Web

- A brain for Human Kind
- From Information-based to Knowledge-Based
- Processable Knowledge means:
  - Better Retrieval
  - Reasoning
- Where can IE contribute?

Building the SW

- Document annotation
  - Manually associate documents (or parts) to ontological descriptions
    - Document classification for retrieval
      - Where can I buy an Hamster?
      - Pet shop web page -> pet shop concept -> hamster
    - Knowledge annotation
      - Where can I find a hotel in Berlin where single rooms cost less than 400€?
      - The Hotel is located in central Berlin and the cost for a single room is 300€
  - Editors are currently available for manual annotation of texts

IE for Annotating Documents

- Manual annotation is
  - Expensive
  - Error prone
- IE can be used for annotating documents
  - Automatically
  - Semi-Automatically
    - As user support
- Advantages
  - Speed
  - Low cost
  - Consistency
  - Can provide automatic annotation different from the one provided by the author(!)

SW for Knowledge Management

- SW is important for everyday Internet users
- SW is necessary for large companies
  - Millions of documents where knowledge is interspersed
  - Most documents are now
    - web-based
    - Available over an Intranet
  - Companies are valued for their
    - Tangible assets (e.g. plants)
    - Intangible assets (e.g. knowledge)
  - Knowledge is stored in
    - mind of employees
    - Documentation
  - Companies spend 7-10% of revenues for KM

Why Adaptive Systems?

- Writing IE systems by hand is difficult and error prone
  - Extraction languages can be quite complex
  - Tedious write-test-debug-rewrite cycle
- Adaptive systems learn from user annotations
  - the person tells the learning algorithm **what** to extract: The learner figures out **how**
- Advantages
  - Annotating text is simpler & faster than writing rules.
  - Domain independent
  - Domain experts don't need to be linguists or programers.
  - Learning algorithms ensure full coverage of examples.

Algorithms and Methodologies

A dip into the details of IE for the Web

- Introduction: (20 minutes)
- Algorithms and methodologies (100 min)
  - Wrapper induction
  - Boosted wrapper induction
  - Hidden Markov models
  - Exploiting linguistic constraints
- IE in practice (30 min)
- Conclusion, Future Work (10 min)
- Discussion

Algorithms: Outline

- Wrappers
- Hand-coded wrappers
- Wrapper induction
- Learning highly expressive wrappers
- Boosted wrapper induction
- Hidden Markov models
- Exploiting linguistic constraints

structured data

natural text

Wrapper induction

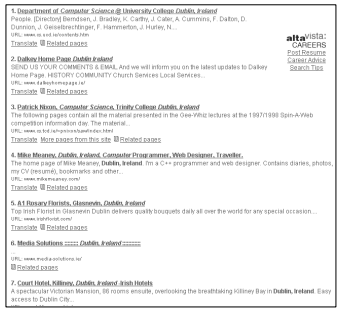
Highly regular source documents

⇓

Relatively simple extraction patterns

⇓

Efficient learning algorithms



Wrappers: Example and Toolkits

⟨ (Congo, 242)
(Egypt, 20)
(Belize, 501)
(Spain, 34) ⟩

- Wrapper toolkits: Specialized programming environments for writing & debugging wrappers by hand
- Examples
  - World Wide Web Wrapper Factory[db.cis.upenn.edu/W4F]
  - Java Extraction & Dissemination of Information [www.darmstadt.gmd.de/oasys/projects/jedi]

Wrappers: Delimiter-based extraction

<HTML><TITLE>Some Country Codes</TITLE>
<B>Congo</B> <I>242</I>
<B>Egypt</B> <I>20</I>
<B>Belize</B> <I>501</I>
<B>Spain</B> <I>34</I>
</BODY></HTML>

Use <**B**>, </**B**>, <**I**>, </**I**> for extraction

"Left-Right" wrappers

Left-**R**ight wrapper ≡ **2K strings**

$\langle l_1, r_1, \ldots, l_K, r_K \rangle$

left delimiters **right** delimiters

```
procedure ExtractCountryCodes
    while there are more occurrences of <B>
        1. extract Country between <B> and </B>
        2. extract Code between <I> and </I>
```

```
procedure ExtractAttributes:
    while there are more occurrence of l1
        1. extract 1st attribute between l1 and r1
        . . .
        K. extract Kth attribute between lK and rK
```

[Kushmerick et al, IJCAI-97; Kushmerick AIJ-2000]

Wrapper induction

examples ⮝ hypothesis

Thai food is spicy.
Vietnamese food is spicy.
German food isn't spicy.
⮝ Asian food is spicy.

⮝ wrapper

Learning LR wrappers

labeled pages ***wrapper***

<HTML><HEAD>Some Country Codes</HEAD>
<HTML><HEAD>Some Country Codes</HEAD>
<HTML><HEAD>Some Country Codes</HEAD>
<HTML><HEAD>Some Country Codes</HEAD>
<B>**Congo**</B> <I>**242**</I>
<B>**Egypt**</B> <I>**20**</I>
<B>**Belize**</B> <I>**501**</I>
<B>**Spain**</B> <I>**34**</I>
</BODY></HTML>

→ $\langle l_1, r_1, \ldots, l_K, r_K \rangle$

Example: Find **4** strings
⟨<B>, </B>, <I>, </I>⟩
⟨ l_1, r_1, l_2, r_2 ⟩

LR: Finding r_1

<HTML><TITLE>Some Country Codes</TITLE>
<B>Congo</B> <I>242</I>
<B>Egypt</B> <I>20</I>
<B>Belize</B> <I>501</I>
<B>Spain</B> <I>34</I>
</BODY></HTML>

r_1 can be any ***prefix***
eg </B>

LR: Finding l_1, l_2 and r_2

<HTML><TITLE>Some Country Codes</TITLE>
<B>Congo</B> <I>242</I>
<B>Egypt</B> <I>20</I>
<B>Belize</B> <I>501</I>
<B>Spain</B> <I>34</I>
</BODY></HTML>

r_2 can be any ***prefix***
eg </I>

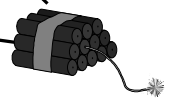
l_2 can be any ***suffix***
eg <I>

l_1 can be any ***suffix***
eg <B>

A problem with LR wrappers

Distracting text in head and tail

<HTML><TITLE>Some Country Codes</TITLE>
<BODY><B>Some Country Codes</B><P>
<B>Congo</B> <I>242</I>
<B>Egypt</B> <I>20</I>
<B>Belize</B> <I>501</I>
<B>Spain</B> <I>34</I>
<HR><B>End</B></BODY></HTML>



One (of many) solutions: HLRT

Ignore page's **head** and **tail**

```
<HTML><TITLE>Some Country Codes</TITLE>
<BODY><B>Some Country Codes</B><P>
<B>Congo</B> <I>242</I><BR>
<B>Egypt</B> <I>20</I><BR>
<B>Belize</B> <I>501</I><BR>
<B>Spain</B> <I>34</I><BR>
<HR><B>End</B></BODY></HTML>
```

end of head } head

} body

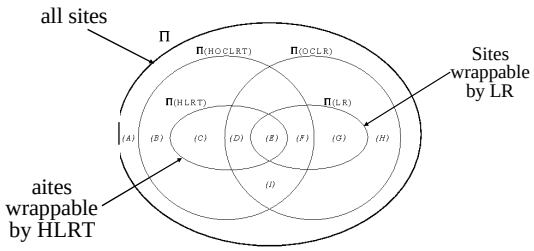
} tail

start of tail

"**H**ead-**L**eft-**R**ight-**T**ail wrappers

Expressiveness

Theorem:



Coverage

wrapper class(es)	coverage (%)
LR ∪ HLRT ∪ OCLR ∪ HOCLRT ∪ N-LR ∪ N-HLRT	70
LR ∪ HLRT ∪ OCLR ∪ HOCLRT	60
LR ∪ OCLR	53
LR	53
OCLR	53
HLRT ∪ HOCLRT	57
HLRT	57
HOCLRT	57
N-LR ∪ N-HLRT	53
N-LR	13
N-HLRT	50
N-LR ∪ N-HLRT but not LR ∪ HLRT ∪ OCLR ∪ HOCLR	25

Fraction of randomly-selected "data-heavy" Web sites (search engines, retail, weather, news, finance, …) for which wrapper in a given class was learned.

Sample complexity

- The key problem with machine learning: Training data is expensive and tedious to generate
- In practice, active learning and specialized algorithms have reduced training requirements considerably
- But this isn't theoretically satisfying
- Computational Learning Theory:
  - **Time complexity**: Time require by an algorithm to terminate, as a function of problem parameters
  - **Sample complexity**: Training data required by a learning algorithm to converge to correct hypothesis, as a function of problem parameters

A Model of Sample Complexity

P[correct wrapper] = f(size of documents,
number of documents,
number of attributes per record)

Analyze wrapper learning task to derive this function

(Actually, we can compute only a bound on this probability.)

Just like time/space complexity:

time[learn wrapper] = g(size of documents,
number of documents,
number of attributes per record)

PAC results - LR wrappers

Theorem: Suppose we learn LR wrapper W from training set E, where the longest document has length R and each record contains K attributes. If

$$|E| \geq \frac{1}{\varepsilon}(2K \ln R - \ln \delta)$$

then W is probably approximately correct

error(W) < ε

with probability at least 1-δ

More sophisticated wrappers

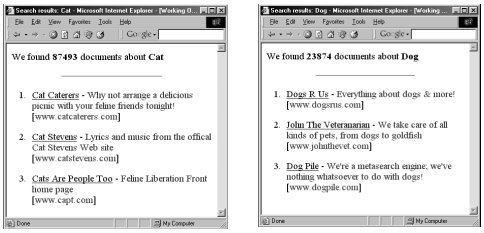
- LR & HLRT wrappers are extremely simple
  (though useful for ~ $^2/_3$ of real Web sites!)
- Recent wrapper induction research has explored…
  - **more expressive wrapper classes**
    [Muslea et al, Agents-98; Hsu et al, JIS-98; Thomas et al, JIS-00, …]
    - Disjunctive delimiters
    - Sequential/landmark-based delimiters
    - Multiple attribute orderings
    - Missing attributes
    - Multiple-valued attributes
    - Hierarchically nested data
  - **Wrapper verification/maintenance**
    [Kushmerick, AAAI-1999; Kushmerick WWWJ-00; Cohen, AAAI-1999; Minton et al, AAAI-00]

One of my favorites

- **Roadrunner**
  [Valter Crescenzi et al; Univ Roma 3]
- Unsupervised wrapper induction
  - They research databases, not machine learning, so they didn't realize training data was needed :-)
- Intuition:
  - Pose two different queries
  - The common bits of the documents come from the template and can be ignored
  - The bits that are different are the data that we're looking for

Roadrunner - Example

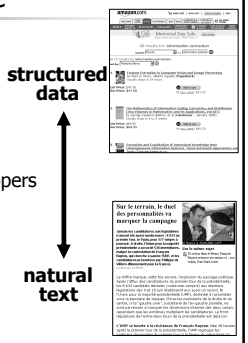
- Common content = Part of template
  Varying content = The data!



- Complications: Dynamic but unwanted content -- eg advertisements or timestamps

Algorithms: Outline

- ✓ Wrappers
- ✓ Hand-coded wrappers
- ✓ Wrapper induction
- ✓ Learning highly expressive wrappers
- Boosted wrapper induction
- Hidden Markov models
- Exploiting linguistic constraints



Boosted wrapper induction

[Freitag & Kushmerick, AAAI-00]

- Wrapper induction is suitable only for rigidly-structured machine-generated HTML…
- … or is it?!
- Can we use simple patterns to extract from natural language documents?

… **Name:** Dr. Jeffrey D. Hermes …
… **Who:** Professor Manfred Paul …
… will be given by Dr. R. J. Pangborn …

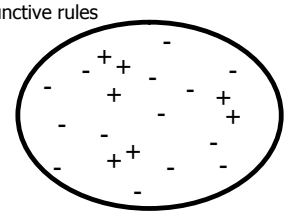
… Ms. Scott **will be speaking …**
… Karen Shriver**, Dept. of ...**
… Maria Klawe**, University of ...**

BWI: The basic idea

- Learn "wrapper-like" patterns for natural texts
  pattern = exact token sequence
- Learn many such "weak" patterns
- Combine with boosting to build "strong" ensemble pattern
- Of course, not all natural text is sufficiently regular!
- Demo: www.smi.ucd.ie/bwi

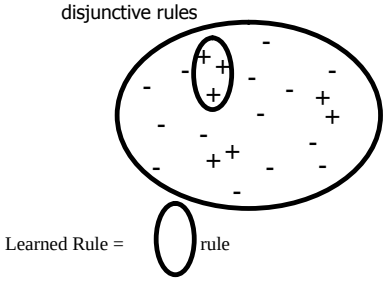
Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules



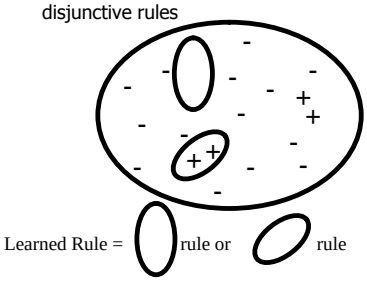
Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules



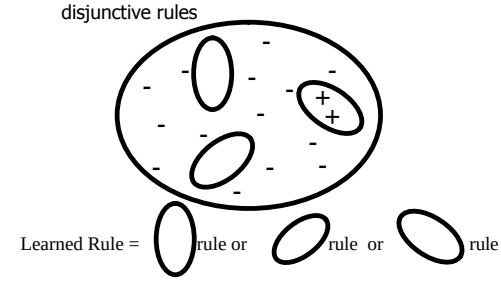
Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules



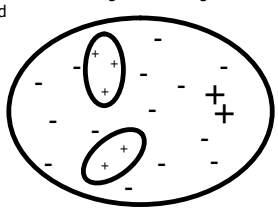
Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules



Boosting = Generalized Covering

- When learn rules on iteration t, give less weight to (but don't entirely discard) training examples successfully handled in iterations 1, 2, …, t-1
- Equivalently: Give more weight to training data that has not yet been covered

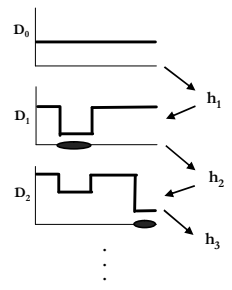


Boosting [Schapire & Singer, 1998]

$D_1(i)$ = uniform distribution over training examples

for t = 1, …, T
train: use distribution D_t to learn weak hypothesis: h_t: X Þ R
reweight: choose a_t, and modify distribution D_t to emphasize examples missed by h_t: $D_{t+1}(i) = D_t(i) \exp(-a_t y_i h_t(x_i))$

return:
$H(x) = \text{sign}(S_t a_t h_t(x))$



Weak hypotheses: Boundary Detectors

Boundary Detector: [**who :**][**dr . <Capitalized>**]

prefix *suffix*

matches (e.g.) "… **Who: Dr. Richard Nixon** …"

Weak Learning Algorithm

Greedy growth from null detector
Pick best prefix/suffix extension at each step
Stop when no further extension improves accuracy

Weighting

$a_t = \frac{1}{2} \ln[(W^+ + e) / (W^- + e)]$

[Cohen & Singer, 1999]

Boosted Wrapper Induction

Training

input: labeled documents

Fore = Adaboost fore detectors
Aft = Adaboost aft detectors
Lengths = length histogram

output: **Extractor** = <*Fore*, *Aft*, *Lengths*>

Execution

input: Document, Extractor, t

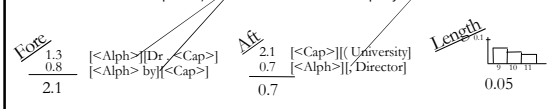
$F = \{\langle i, c_i \rangle \mid$ token i matches *Fore* with confidence $c_i\}$
$A = \{\langle j, c_j \rangle \mid$ token j matches *Aft* with confidence $c_j\}$

output:
$\{\langle i,j \rangle \mid \langle i, c_i \rangle \in F, \langle j, c_j \rangle \in A, c_i \cdot c_j \cdot L(j-i) > t\}$

BWI execution example

```
<0.26.4.95.09.31.03.bg02+@andrew.cmu.edu.0>
Type:     cmu.andrew.official.cmu-news
Topic:    Chem. Eng. Seminar
Dates:    2-May-95
Time:     10:45 AM
PostedBy: Bruce Gerson on 26-Apr-95 at 09:31 from andrew.cmu.edu
Abstract:

The Chemical Engineering Department will offer a seminar entitled
"Creating Value in the Chemical Industry," at 10:45 a.m., Tuesday, May 2
in Doherty Hall 1112.
The seminar will be given by Dr. R. J. (Bob) Pangborn, Director, Central
Research and Development, The Dow Chemical Company.
```



Confidence of **"Dr. R. J. (Bob) Pangborn"** = 2.1 ·0.7 ·0.05 = 0.074

Samples of learned patterns

Speaker: Reid Simmons, School of …
[speaker :][<Alph>]
[speaker <Any>][<FName>]

Presentation Abstract Joe Cascio, IBM
Set Constraints Alex Aiken (IBM, Almaden)
[<Cap>][<FName> <Any> <Punc> ibm]

John C. Akbari is a Masters student at
Michael A. Cusumano is an Associate Professor of
Lawrence C. Stewart is a Consultant Engineer at
[. <Any>][is <ANum> <Cap>]

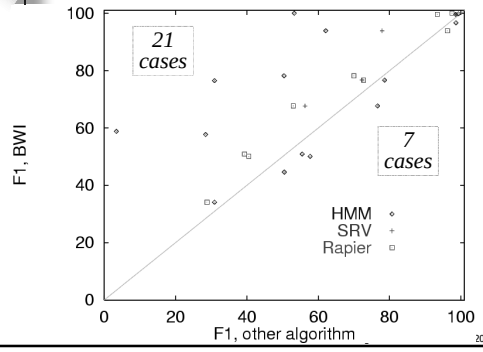
Evaluation

- Wrappers are usually 100% accurate, but perfection is generally impossible with natural text
- ML/IE community has a well developed evaluation methodology
  - **Cross-validation**: Repeat many times - randomly select 2/3 of the data for training, test on remaining 1/3.
  - **Precision**: fraction of extracted items that are correct
  - **Recall**: fraction of actual items extracted
  - **F_1** = 2 / (1/P + 1/R)
- 16 IE tasks from 8 document collections

seminar announcements	Zagats restaurant reviews
job listings	LA Times restaurant reviews
Reuters corporate acquisitions	Internet Address Finder
CS department faculty lists	Stock quote server

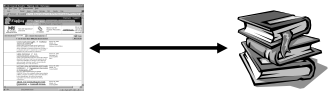
- Competitors: SRV, Rapier, HMM

Results: 16 tasks x 4 algorithms



Boosted Wrapper Induction: Controversial(?) Conclusion

- Is the **Great Web -vs- Natural Text Chasm** more apparent than real?



- IE is possible if the documents contain regularities that can be exploited
- But the "reason" (eg, linguistic -vs- markup) for these regularities doesn't much matter
- See also Soderland's WHISK & Webfoot

Algorithms: Outline

- ✓ Wrappers
- ✓ Hand-coded wrappers
- ✓ Wrapper induction
- ✓ Learning highly expressive wrappers
- ✓ Boosted wrapper induction
- Hidden Markov models
- Exploiting linguistic constraints

structured data

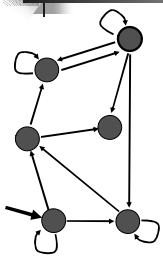
natural text

Hidden Markov models

- Previous discussion examine systems that use explicit extraction patterns/rules
- HMMs are a powerful alternative based on statistical token models rather than explicit extraction patterns.

[Leek, UC San Diego, 1997; Bikel et al, ANLP-97, MLJ 99; Freitag & McCallum, AAAI-99 MLIE Workshop; Seymore, McCallum & Rosenfeld, AAAI-99 MLIE Workshop; Freitag & McCallum, AAAI-2000]

HMM formalism



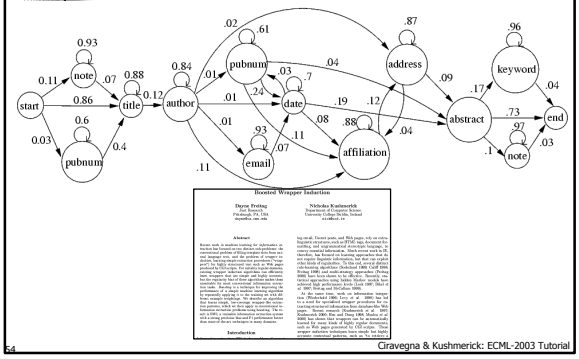
HMM = states s_1, s_2, …
special start state s_1
special end state s_n
token alphabet a_1, a_2, …
state transition probs $P(s_i | s_j)$
token emission probs $P(a_i | s_j)$

Widely used in many language processing tasks, *e.g.* speech recognition [Lee, 1989], POS tagging [Kupiec, 1992], topic detection [Yamron *et al,* 1998]

Applying HMMs to IE

- **Document** ⇒ generated by a stochastic process modelled by an HMM
- **Token** ⇒ word
- **State** ⇒ "reason/explanation" for a given token
  - 'Background' state emits tokens like 'the', 'said', …
  - 'Money' state emits tokens like 'million', 'euro', …
  - 'Organization' state emits tokens like 'university', 'company', …
- **Extraction**: The Viterbi algorithm is a dynamic programming technique for efficiently computing the most likely sequence of states that generated a document.

HMM for research papers **[Seymore et al, 99]**



Learning HMMs

- **Good news:**
  - If training data tokens are tagged with their generating states, then simple frequency ratios are a maximum-likelihood estimate of transition/emission probabilities. (Use smoothing to avoid zero probs for emissions/transitions absent in the training data.)
- **Great news:**
  - Baum-Welch algorithm trains HMM using unlabelled training data!
- **Bad news:**
  - How many states should the HMM contain?
  - How are transitions constrained?
  - Insufficiently expressive ⇒ Unable to model important distinctions
  - Overly-expressive ⇒ sparse training data, overfitting

HMM example

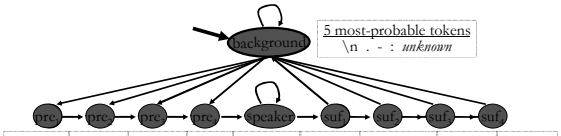
"Seminar announcements" task

```
<0.15.4.95.15.11.55.rudibear+@CMU.EDU.0>
Type:  cmu.andrew.assocs.UEA
Topic: Re: entreprenuership speaker
Dates: 17-Apr-95
Time: 7:00 PM
PostedBy: Colin S Osburn on 15-Apr-95 at 15:11 from CMU.EDU
Abstract:

hello again
to reiterate
there will be a speaker on the law and startup business
this monday evening the 17th
it will be at 7pm in room 261 of GSIA in the new building, ie
upstairs.
please attend if you have any interest in starting your own
business or
are even curious.
Colin
```


HMM example, continued

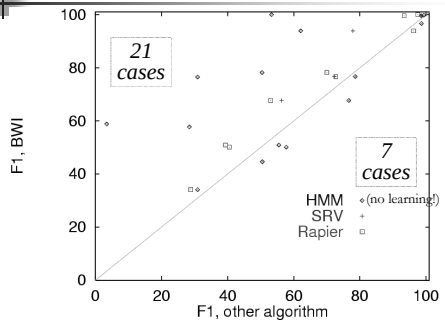
Fixed topology that captures limited context:
4 "prefix" states before & 4 "suffix" after target state



\n	\n	\n	\n	*unknown*	\n	\n	\n	\n
seminar	:	who	:	.	,	*unknown*	of	of
.	.	speaker	.	dr	will	.	.	*unknown*
robotics	-	:	with	professor	(	department	,	.
unknown	*unknown*	.	,	michael	-	the	*unknown*	:

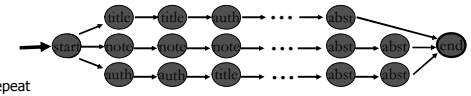
[Freitag, 99]

Evaluation



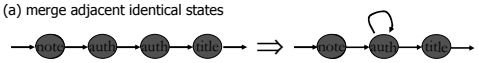
Learning HMM structure [Seymore et al, 1999]

start with maximally-specific HMM (one state per observed word):

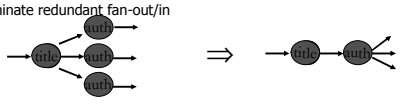


repeat

(a) merge adjacent identical states

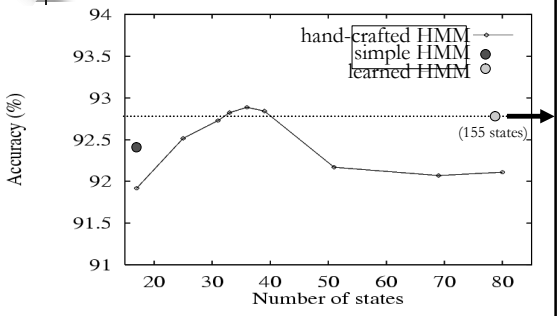


(b) eliminate redundant fan-out/in



until obtain good tradeoff between HMM accuracy and complexity

Evaluation



Algorithms: Outline

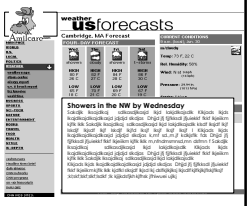
- ✓ Wrappers
- ✓ Hand-coded wrappers
- ✓ Wrapper induction
- ✓ Learning highly expressive wrappers
- ✓ Boosted wrapper induction
- ✓ Hidden Markov models
- Exploiting linguistic constraints

structured data

natural text

Exploiting linguistic constraints

- IE research has its roots in the NLP community
- many extraction tasks require non-trivial linguistic processing
- Web Documents types can range from free texts to rigid HTML documents (e.g. tables)
  - Even a mixture of them!
- Is NLP robust enough to cope with such situations?



Current Approaches

- NLP Approaches (MUC-like Approaches)
  - Ineffective on most Web-related texts:
    - web pages/emails
    - stereotypical but ungrammatical texts
  - Extra-linguistic structures convey information
    - HTML tags, Document formatting, Regular stereotypical language
- Wrapper induction systems
  - Designed for rigidly structured HTML texts
  - Ineffective on unstructured texts
    - Approaches avoid generalization over flat word sequence
    - Data Sparseness on free texts

Lazy NLP based Algorithm

- Learns the best level of language analysis for a specific IE task mixing deep linguistic and shallow strategies
  1. Initial rules: shallow wrapper-like rules
  2. Linguistic Information (LI) progressively added to rules
  3. Addition stopped when LI becomes
     - unreliable
     - ineffective
- Lazy NLP learns best strategy for each information/context separately
  - Example:
    - Using parsing for recognising the speaker in seminar announcements,
    - Using shallow approaches to spot the seminar location

$(LP)^2$ [Ciravegna 2001 – IJCAI 01- ATEM01]

- Covering algorithm based on LazyNlp
- Single tag learning (e.g. </speaker>)
- Tagging Rules
  - Insert annotation in texts
- Correction Rules
  - Correct imprecision in information identification by shifting tags to the correct position

TBL-like, with some fundamental differences

Tagging and Correction Rules: examples

the seminar at <time> 4 pm </time> will

Condition on Words	Action: Insert Tag
the	
seminar	
at	<time>
4	
pm	

Initial rules= window of conditions on words

The seminar at 4 </time> PM will be held in Room 201

Condition		Action
word	wrong tag	correct tag
at		
4	</time>	
pm		</time>

Rule Generalisation

- Each instance is generalised by reducing its pattern in length
- Generalizations are tested on training corpus
- Best k rules generated from each instance reporting:
  - Smallest error rate (wrong/matches)
  - Greatest number of matches
  - Cover different examples
- Conditions on words are replaced by information from NLP modules
  - Capitalisation
  - Morphological analysis
    - Generalizes over gender/number
  - POS tagging
    - Generalizes over lexical categories
  - User-defined dictionary or gazetteer
  - Named Entity Recognizer

Implemented as a general to specific beam search with pruning (AQ-like)

Example of generalization

`the seminar at <time> 4 pm will`

Condition	Additional Knowledge				Action
Word	Lemma	LexCat	Case	SemCat	Tag
the	the	det	low		
seminar	seminar	noun	low		
at	at	prep	low		
4		digit	low		<time>
pm		noun	low	timeid	
will	will	verb	low		

Condition					Action
Word	Lemma	LexCat	Case	SemCat	Tag
	at				
		digit			<time>
				timeid	

Details of the algorithm in [Ciravegna 2001 - ATEM01]

CMU: detailed results

	(LP)²	BWI	HMM	SRV	Rapier	Whisk
speaker	**77.6**	67.7	76.6	56.3	53.0	18.3
location	75.0	76.7	**78.6**	72.3	72.7	66.4
stime	99.0	**99.6**	98.5	98.5	93.4	92.6
etime	95.5	93.9	62.1	77.9	**96.2**	86.0
All Slots	**86.0**	**83.9**	**82.0**	**77.1**	**77.3**	**64.9**

1. Best overall accuracy
2. Best result on speaker field
3. No results below 75%

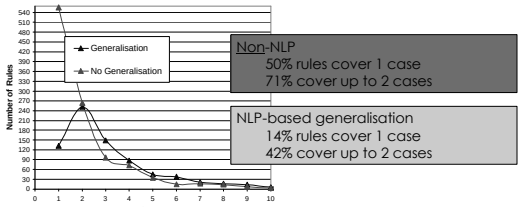
Effect of Generalization(1)

Effectiveness and reduction in data sparseness

Slot	$(LP)^2_G$	$(LP)^2_{NG}$
speaker	72.1	14.5
location	74.1	58.2
stime	100	97.4
etime	96.4	87.1
All slots	**89.7**	**78.2**

} Most Interesting

With comparable effectiveness on training corpus!



Non-NLP
50% rules cover 1 case
71% cover up to 2 cases

NLP-based generalisation
14% rules cover 1 case
42% cover up to 2 cases

Best level of Generalization

- ITC seminar announcements (mixed Italian/English)
  - Date, time, location generally in Italian
  - Speaker, title and abstract generally in English
  - English POS also for the Italian part
  - NLP-based outperforms other version

	Words	POS	NE
speaker	74.1	**75.4**	**84.3**
title	62.8	**62.4**	62.8
date	90.8	**93.4**	**93.9**
time	100	100	100
location	95.0	95.0	**95.5**

Linguistic constraints: Conclusions

- Linguistic phenomena can't be handled by simple wrapper-like extraction patterns
- Even shallow linguistic processing (eg POS tagging) can improve performance dramatically.
  - NOTE: linguistic processing must be regular, not necessarily correct!
  - Example
    `(LexCat:NNP + <BR> + <BR>)<SPEAKER>(NER:<person>)`
    none of the covered 32 examples starts actually with an NNP
- What about more sophisticated NLP techniques?
  - Extension to parsing and corefernce resolution?

Putting IE into Practice

Enabling non-experts to port IE systems

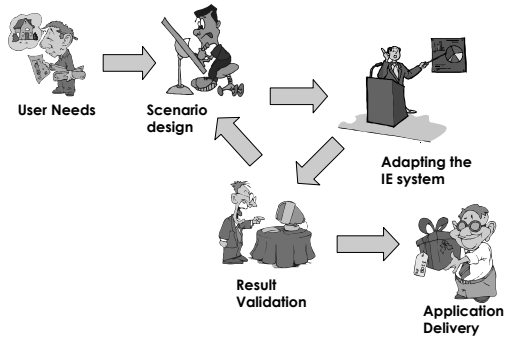
- Introduction: (20 minutes)
  - what is IE, what can we extract from the Web and why?
- Algorithms and methodologies (100 min)
- IE in practice (30 min)
  - The adaptation problem (20 min)
  - WEB + IE: examples of systems (10 min)
- Conclusion, Future Work (10 min)
- Discussion

Motivation

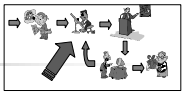
- Impact on the web community will come only if:
  - IE systems are portable by non IE experts
  - Low cost porting
- Non experts
  - Need specific easy to use tools to:
    - Design application
    - Tune application
    - Deliver application
  - Need support during the whole IE application definition process

In summarising the summary of the summary:
people are a problem.
Douglas Adams
The Restaurant at the End of the Universe

Application Development Cycle



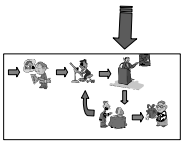
Scenario design



- Task: mapping user whishes into templates
- Necessity:
  - Supporting users in:
    - relevant information identification
    - scenario organization
- Relevant Information Identification:
  - Different situations:
    - User with developed scenario
      - System: no action, but…
    - User with preliminary scenario to be refined
      - System helps in refining
    - User with no scenario
      - System helps in
        - Identifying relevant information
        - Organising it into a scenario

Training

- User can select unrepresentative corpora
  - Unbalanced wrt genres
    - System validates corpus wrt a large corpus
      - Comparing formal features
  - Unwanted regularities (use of keywords for selection)
    - System looks for unusual regularities
  - Irrelevant texts (sensitive information)
    - No solution to stupidity



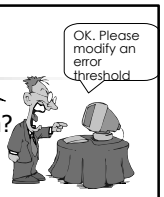
Tagging Corpora

- Problems:
  - Tagging texts can:
    - Be difficult and boring
    - Take a long time
- Effect:
  - Mistakes in tagging
  - High cost
- System:
  - reduce/eliminate need for annotated data
    - **Bootstrapping:** from user-defined "seed examples" to system-retrieved similar examples
    - **Active learning**: selection of examples to annotate from unlabeled corpus

Helps in discovering new relations

Helps in focusing on unusual information shape

Result Validation



- How well does the system perform?
  - Solution:
    - Facilities for:
      - Inspecting tagged corpus
      - Showing details on correctness
        - Statistics on corpus
        - Details on errors (highlight correct/incorrect/missing)

      (e.g. MUC scorer is an excellent tool)
- Influencing system behavior
  - Solution
    - Interface for bridging the user's qualitative vision and the system's numerical vision

Application Delivery

- Problem:
  - Incoming texts deviate from training data
    - Training corpus non representative
    - Document features change in time
- Solution:
  - Monitoring application.
    - Warn user if incoming texts' features are statistically different from training corpus:
      - Formal features: texts length, distribution of nouns
      - Semantic features: distribution of template fillers

Putting IE into Practice (2)

Some examples of Adaptive User-driven IE for real world applications

Learning Pinocchio

- Commercial tool for adaptive IE
  - Based on the $(LP)^2$ algorithm
  - Adaptable to new scenarios/applications by:
    - Corpus tagging via SGML
    - A user with analyst's knowledge
- Applications
  - "Tombstone" data from Resumees (Canadian company) (E)
  - IE from financial news (Kataweb) (I)
  - IE from classified ads (Kataweb) (I)
  - Information highlighting (intelligence)
  - (Many others I have lost track of…)
- A number of licenses released around the world for application development

[Ciravegna 2001 - IJCAI]
http://tcc.itc.it/research/textec/tools-resources/learningpinocchio/

Application development time

Resumees:

- Scenario definition: 10 person hours
- Tagging 250 texts: 14 person hours
- Rule induction: 72 hours on 450MHz computer
- Result validation: 4 hours

Contact:

Alberto Lavelli
ITC-Irst
lavelli@itc.it

http://tcc.itc.it/research/textec/tools-resources/learningpinocchio/

Amilcare

active annotation for the Semantic Web

Tool for adaptive IE from Web-related texts

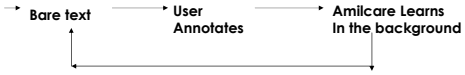
- Based on $(LP)^2$
- Uses Gate and Annie for preprocessing
- Effective on different text types
  - From free texts to rigid docs (XML,HTML, etc.)
- Integrated with
  - MnM (Open University) Ontomat (University of Karlsruhe)
  - Gate (U Sheffield)
- **Adapting Amilcare:**
  - Define a scenario (ontology)
  - Define a Corpus of documents
  - Annotate texts
    - Via MnM, Gate, Ontomat
  - Train the system
  - Tune the application (*)
  - Deliver the application

[Ciravegna 2002 -SIGIR]
www.dcs.shef.ac.uk/~fabio/Amilcare.html

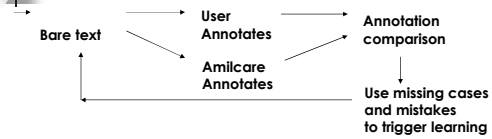
Non-Intrusive Active Learning

- Amilcare is specifically designed as companion for text annotation
  - It can be inserted in the usual tagging environment
  - It works in the background
  - At some point it will start helping the user in tagging

Bootstrapping Annotation

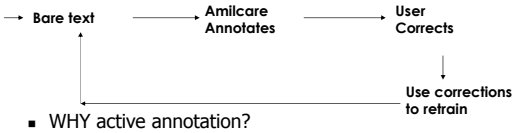


Learning to annotate



Active Annotation

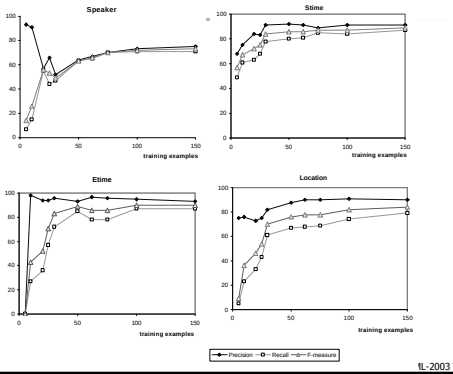
- When Amilcare's rules reach a user-defined accuracy



- WHY active annotation?
  - Focuses the slow and expensive user activity on uncovered cases
  - Avoids annotating covered cases
  - Validating extracted information is
    - Simpler & less error prone

    Than annotating bare texts speeding up the process of corpus annotation considerably.

Is IE useful as Help for Tagging?



Conclusions on IE and Tagging

Tag	Amount of Texts needed for training	Prec	Rec
stime	30	91	78
etime	20	96	72
location	30	82	61
speaker	100	75	70

- Integration of IE (Amilcare+Gate) and Ontology-based Annotation Tools (MnM and Ontomat)
- First step towards a new generation of OEs
- Active Learning can provide an interesting interaction modality
  - User friendly
  - Adaptable

Summary and Conclusions

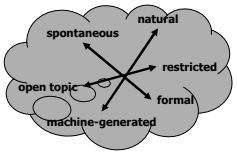
The summary of the summary
Where do we go from now?

Summary

- Information extraction:
  - core enable technology for variety of next-generation information services
    - Data integration agents
    - Semantic Web
    - Knowledge Management
- Scalable IE systems must be adaptive
  - automatically learn extraction rules from examples
- Dozens of algorithms to choose from
- State of the art is 70-100% extraction accuracy (after hand-tuning!) across numerous domains.
  - Is this good enough? Depends your application.
- Yeah, but does it really work?!
  - Several companies sell IE products.
  - SW ontology editors start including IE

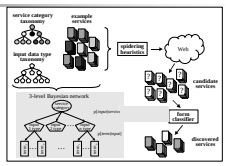
Open issues, Future directions

- Knob-tuning will continue to deliver substantial incremental performance increments
- Grand Unified Theory of text "structuredness", to automatically select optimal IE algorithm for a given task

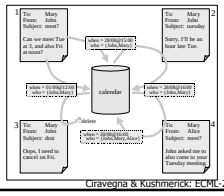


Open Issues, Future directions

- Resource Discovery



- Cross-Document Extraction



Open issues, Future directions

- Adaptive only?
  - Mentioned systems are designed for non experts
    - E.g. do not require users to revise or contribute rules.
    - Is this a limitation? What about experts or even the whole spectrum of skills?
  - Future direction: making the best use of user's knowledge
- Expressive enough?
  - What about filling templates?
    - Coreferences
      `(ACME is producing part for YMB Inc. The company will deliver…)`
    - Reasoning
      `(if X retires then X leaves his/her company)`