

Information Extraction

Lecture 3 – Rule-based Named Entity Recognition

CIS, LMU München

Winter Semester 2015-2016

Dr. Alexander Fraser, CIS

Administravia

- Seminars tomorrow and next Wednesday:
Gobi Computerlabor
 - Bring your linux laptop if you want
 - Exercise with Fabian Dreer (and me) doing rule-based extraction with python
 - People only in the Vorlesung are also invited if interested (but bonus points are only available as part of the Hausarbeit in the Seminar unfortunately)
 - Will practically apply handcrafted rule-based NER and measure performance with precision and recall (presented later in this lecture)
 - Later we will use the same data to build classifiers

Administravia II

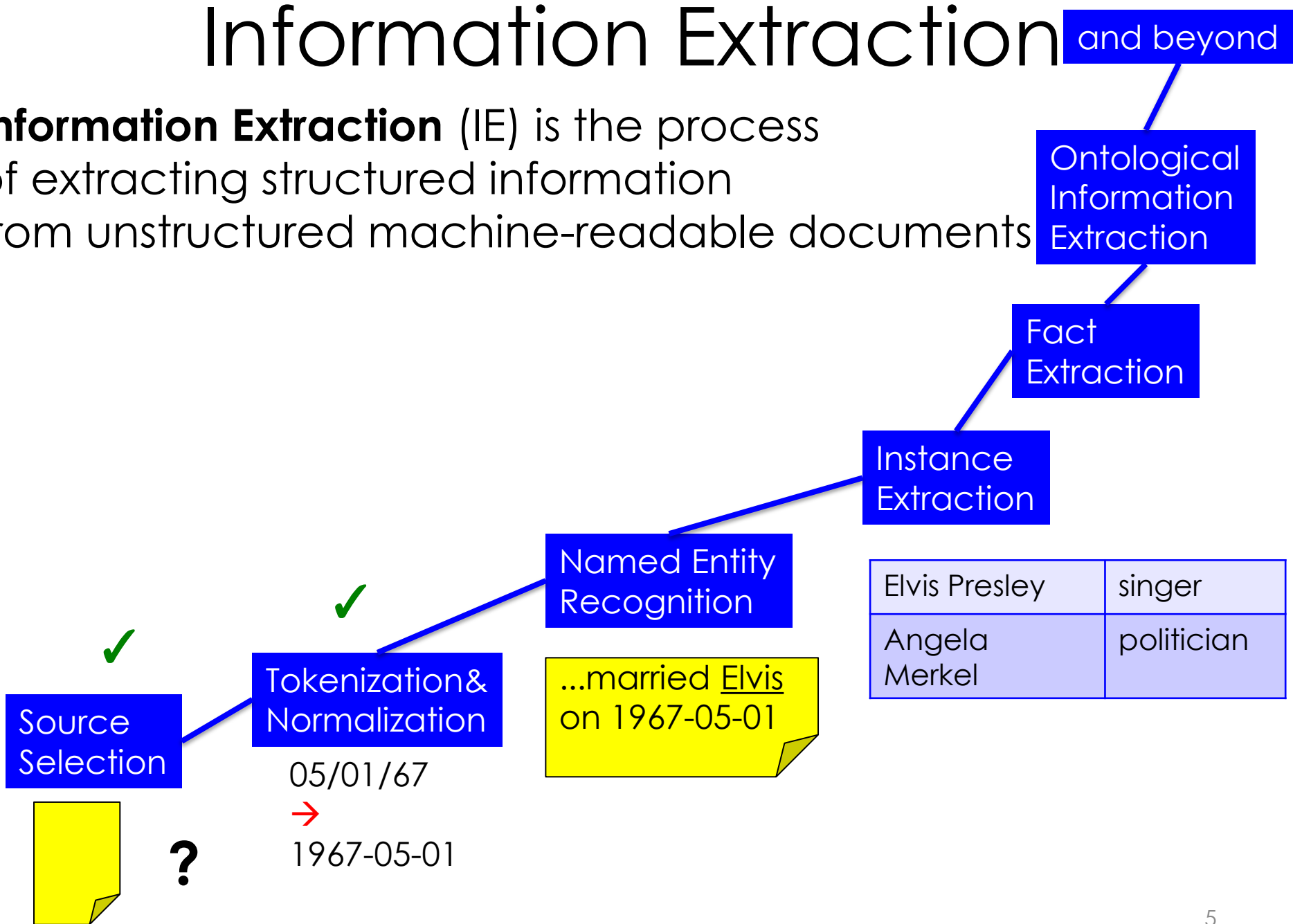
- As I warned before, first seminar topics on Thursdays are one week from tomorrow (!)
- First seminar topics on Wednesdays are two weeks from today
- Take a look at topics from the previous year (you should know the password)
- Seminar topics sign-up is tomorrow evening from 19:00
 - **No emails before this!!** (or it is the comfy chair for you!)
 - You can specify multiple topics (ranked) if you want
 - I might counter-propose a similar topic if there is a topic that you particularly want (but I might not be able to do this, don't count on it!)
- You can look at the seminar web page as I update it (probably Friday), click the refresh button in your browser due to possible caching problems

Outline

- Basic evaluation: Precision/Recall
- Rule-based Named Entity Recognition
- Learning Rules
- Evaluation

Information Extraction and beyond

Information Extraction (IE) is the process of extracting structured information from unstructured machine-readable documents



Relation Extraction: Disease Outbreaks

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...

**Information
Extraction System**

<i>Date</i>	<i>Disease Name</i>	<i>Location</i>
Jan. 1995	Malaria	Ethiopia
July 1995	Mad Cow Disease	U.K.
Feb. 1995	Pneumonia	U.S.

Named Entity Recognition

Named Entity Recognition (NER) is the process of finding entities (people, cities, organizations, dates, ...) in a text.

Elvis Presley was born in 1935 in East Tupelo, Mississippi.



Classification

- Can include up to some hundreds of types

- e.g. ACE competition

- Examples:

- Named Entity Recognition:

- Classic tasks (e.g. MUC conferences)

- Includes Named Entities, Time Expressions and Numerical Expression

- Terminology recognition

- Recognition of technical terminology in specialistic documents

- E.g. names of genes, parts of an aircraft, etc.

WASHINGTON, D.C. (October 5, 1999) -
nQuest Inc. today announced that Paul Jacobs, former
Vice-President of E-Commerce at SRA International
has joined the company's executive management

Evaluation

How can the performance of a system be evaluated?

Standard Methodology from Information Retrieval:

- Precision
- Recall
- F-measure (combination of Precision/Recall)

Recall

Measure of how much relevant information the system has extracted (coverage of system).

Basic idea:

$$\text{Recall} = \frac{\text{\# of correct answers given by system}}{\text{total \# of possible correct answers in text}}$$

Recall

Measure of how much relevant information the system has extracted (coverage of system).

Exact definition:

$$\text{Recall} = \begin{cases} 1 & \text{if no possible correct answers} \\ \frac{\text{\# of correct answers given by system}}{\text{total \# of possible correct answers in text}} & \text{else:} \end{cases}$$

Precision

Measure of how much of the information the system returned is correct (accuracy).

Basic idea:

$$\text{Precision} = \frac{\text{\# of correct answers given by system}}{\text{\# of answers given by system}}$$

Precision

Measure of how much of the information the system returned is correct (accuracy).

Exact definition:

Precision = 1 if no answers given by system

else:

$$\frac{\# \text{ of correct answers given by system}}{\# \text{ of answers given by system}}$$

Evaluation

Every system, algorithm or theory should be **evaluated**, i.e. its output should be compared to the **gold standard** (i.e. the ideal output). Suppose we try to find scientists...

Algorithm output:

$O = \{\text{Einstein}, \text{Bohr}, \text{Planck}, \text{Clinton}, \text{Obama}\}$

Gold standard:

$G = \{\text{Einstein}, \text{Bohr}, \text{Planck}, \text{Heisenberg}\}$

Precision:

What proportion of the output is correct?

$$\frac{|O \cap G|}{|O|}$$

Recall:

What proportion of the gold standard did we get?

$$\frac{|O \cap G|}{|G|}$$

Explorative Algorithms

Explorative algorithms extract everything they find.
(very low threshold)

Algorithm output:

$O = \{\text{Einstein, Bohr, Planck, Clinton, Obama, Elvis, ...}\}$

Gold standard:

$G = \{\text{Einstein, Bohr, Planck, Heisenberg}\}$

Precision:

What proportion of the output is correct?

BAD

Recall:

What proportion of the gold standard did we get?

GREAT

Conservative Algorithms

Conservative algorithms extract only things about which they are very certain

(very high threshold)

Algorithm output:

$O = \{\text{Einstein}\}$

Gold standard:

$G = \{\text{Einstein, Bohr, Planck, Heisenberg}\}$

Precision:

What proportion of the output is correct?

GREAT

Recall:

What proportion of the gold standard did we get?

BAD

Precision & Recall Exercise

What is the algorithm output, the gold standard, the precision and the recall in the following cases?

1. Nostradamus predicts a trip to the moon for every century from the 15th to the 20th inclusive
2. The weather forecast predicts that the next 3 days will be sunny. It does not say anything about the 2 days that follow. In reality, it is sunny during all 5 days.
3. On a sample of Elvis Radio TM, 90% of songs are by Elvis. An algorithm learns to detect Elvis songs. Out of the 100 songs on Elvis Radio, the algorithm says that 20 are by Elvis (and says nothing about the other 80). Out of these 20 songs, 15 were by Elvis and 5 were not.

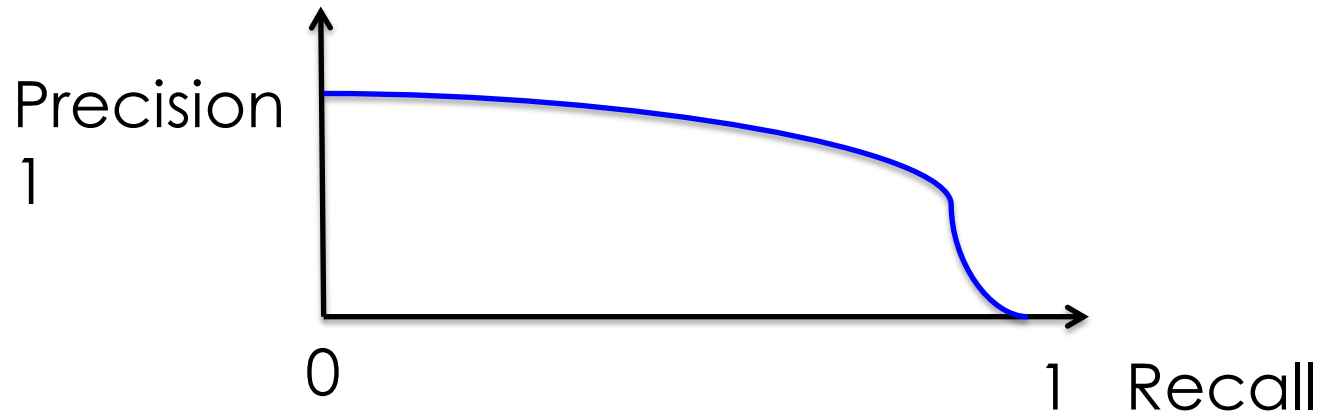
$output = \{e_1, \dots, e_{15}, x_1, \dots, x_5\}$

$gold = \{e_1, \dots, e_{90}\}$

$prec = 15/20 = 75\%$, $rec = 15/90 = 16\%$

F1 - Measure

You can't get it all...



The F1-measure combines precision and recall as the harmonic mean:

$$F1 = 2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall})$$

F-measure

Precision and Recall stand in opposition to one another. As precision goes up, recall usually goes down (and vice versa).

The F-measure combines the two values.

$$F\text{-measure} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- When $\beta = 1$, precision and recall are weighted equally (same as F1).
- When $\beta > 1$, precision is favored.
- When $\beta < 1$, recall is favored.

Summary: Precision/Recall

- Precision and recall are very key concepts
 - Definitely know these formulas, they are applicable everywhere (even real life)!
- F-Measure is a nice way to combine them to get a single number
 - People sometimes don't specify Beta when they say F-Measure
 - In this case Beta=1, i.e., they mean F1, equal weighting of P and R
- We will return to evaluation in more detail later in this lecture
- Now let's look at rules for (open-class) NER



The
University
Of
Sheffield.

Rule-Based Methods for Entity Extraction

Why Rules?

- Many real-life extraction tasks can be conveniently handled through a collection of rules, which are either hand-coded or learnt from examples
- A typical rule-based system consists of:
 - a collection of rules
 - a set of policies to control the firings of multiple rules

Basic rules

- Rules tend to have the form
 - *Contextual Pattern -> Action*
- E.g. Finite State Transducer Rules

```
Rule: Company1 from gate.ac.uk  
  ( ( {Token.orthography == upperInitial} )+  
    {Lookup.kind == companyDesignator}  
  ):match  
-->  
  :match.NamedEntity = { kind=company, rule="Company1" }
```

Token Features

- The String
- Orthography type
- Part of Speech
- Gazetteer information
- Any other information provided by any type of preprocessing

Word	Lemma	PoS	case	Gaze
the	the	Art	low	
seminar	Seminar	Noun	low	
at	at	Prep	low	
4	4	Digit	low	
pm	pm	Other	low	timeid
will	will	Verb	low	

Types of Entity Rules

- Identifying an entity requires recognition of a portion of the document and to insert an XML tag
 - SGML tags in the old days
- Three approaches tried in literature
 - Whole entity recognition
 - E.g. Annie (Cunnigham 2001), Rapier (Califf 1999), etc.
 - Boundary recognition
 - E.g. (LP)² (Ciravegna 2001), BWI (Kushmerick 2001)
 - Multiple entity recognition
 - E.g. Whisk (Soderland 1999)

Rules to Identify Entities

- The classic approach uses rules that model a whole entity
 - No dependency among entities
 - Rule models
 - Left context + Filler + Right context

Rule: Stime1

Pre:

Word="at"

Fill:

Cat=DIG+

Gaz=timeld

Post

PoS=Aux

Action: TAG(stime)

Matches

at

3

pm

will

Rules to identify boundaries

- Rules model
 - Left context + Right context of each tag
- Different rules to identify `<entity>` and `</entity>`
 - `<entity>` recognised independently from `</entity>`

Rule: Stime1

Pre:

Word="seminar":

Word="at":

Post:

Cat=DIG+

Gaz=timeId

Action: TAG(`<stime>`)

Matches

The
seminar
at
3
pm



Multiple Entities Rules

- Identify more than entity
 - Model the dependency that sometimes exist between entities
 - especially order in very structured pages

Example:

<p> Capitol Hill- **1** br twnhme. DW W/D. Pkg incl
\$**675**. **3** BR upper flr no gar. \$**995**. (206)999-9999

Rule:

ID:7

Pattern: * ('Capitol Hill') * (*Digit*) * '\$' (*Number*)

Output: Rental {Neighborhood \$1} {Bedrooms \$2} {Price \$3}

Rule from: STEPHEN SODERLAND:

Learning Information Extraction Rules for Semi-structured and Free Text,

Machine Learning 1, 440

Discussion

- Multi-entity rules are typically used when there is a lot of structure
- Single-entity rules are often used when manually writing rules
 - Humans are good at creating general rules from a limited number of examples
- Boundary rules are often used in learning approaches
 - They generalize well from few examples
 - For instance, they can use rules for `<stime>` and `</stime>` that are learned from different training examples
 - But they may overgeneralize!

Organising Rule Collections

- When rules are fired
 - More than one can apply for a specific span of text
 - Which rule is to be applied?
- Strategies
 - Unordered rules with ad-hoc strategies
 - E.g. Prefer rules marking larger span of text (longer entities)
 - E.g. <ORG> IBM Corp. </ORG> preferred to <ORG> IBM </ORG>
 - Ordered set of rules
 - E.g. rules are sorted by precision on the training corpus

Rule-based NER

- Through about 2000, handcrafted rule-based NER was better than statistical NER
 - For instance in the Message Understanding Conferences, which featured shared tasks
- Since 2000, statistical approaches have started to dominate the academic literature
- In industry, there is still diversity
 - High **precision** -> rule-based
 - High **recall** -> statistical
 - Between, many different solutions (including combining both approaches)
 - But it (debatably) takes less effort to tune statistical systems to improve precision than to tune rule-based systems to increase recall

Learning Rules

- We will now talk about learning rules
 - Still closely following Sarawagi Chapter 2
- The key resource required is a gold standard annotated corpus
 - This is referred to as the "training" corpus
 - The system "learns" through training
 - The goal is to learn rules which may generalize well to new examples which were not seen during training
- We will discuss bottom-up and top-down creation of rules

Rule Learning Algorithms

- Given an annotated corpus
 - Derive a minimal set of rules that cover all (and only) the annotated examples
 - Or at least to maximise recall and precision
 - As determining the optimal rule set is intractable
 - Existing algorithms follow a greedy hill climbing strategy
 - Learn one rule at a time i.e.:
 - (1) Rset = set of rules, initially empty.
 - (2) While there exists an entity $x \in D$ not covered by any rule in Rset
 - (a) Form new rules around x .
 - (b) Add new rules to Rset.
 - (3) Post process rules to prune away redundant rules.

Overfitting and Overgeneralization

- One key concept here is "**overfitting**" examples
 - What is meant here is that we memorize too much from one example
 - For instance, if we have:

Elvis Presley was born in 1935 in East Tupelo, Mississippi.

- and we memorize that **in this exact context** Elvis Presley is a person, we are failing to generalize to other contexts
- We can also "**overgeneralize**"
 - An example would be to learn that the first word of a sentence is a first name
 - This is true in this sentence
 - But **this rule will apply to every sentence**, and often be wrong

Bottom-Up Rule Formation

- For each annotated example
 - Create 1 rule by selecting a window of words to the left and right of entity/tag
 - Completely overfitting the example
 - Likely 100% precision, very low recall
 - Will cover just the current example (plus all the repetitions)
 - Drop constraints on words in window
 - Identify best rule (set) covering example
 - Remove all other instances covered by rules
 - Covering algorithm



Example

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

Condition	Additional Knowledge				Action
Word	Lemma	LexCat	case	SemCat	Tag
the	the	Art	low		
seminar	Seminar	Noun	low		
at	at	Prep	low		stime
4	4	Digit	low		
pm	pm	Other	low	timeid	
will	will	Verb	low		



Example

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

Condition	Additional Knowledge				Action
Word	Lemma	LexCat	case	SemCat	Tag
at	at	Prep	low		stime
4	4	Digit	low		
pm	pm	Other	low	timeid	



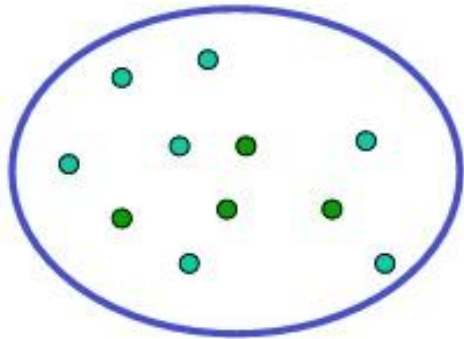
Example

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

Condition	Additional Knowledge				Action
Word	Lemma	LexCat	case	SemCat	Tag
	at				stime
		Digit			
				timeid	



Rule Induction & Generalization



Positive Examples

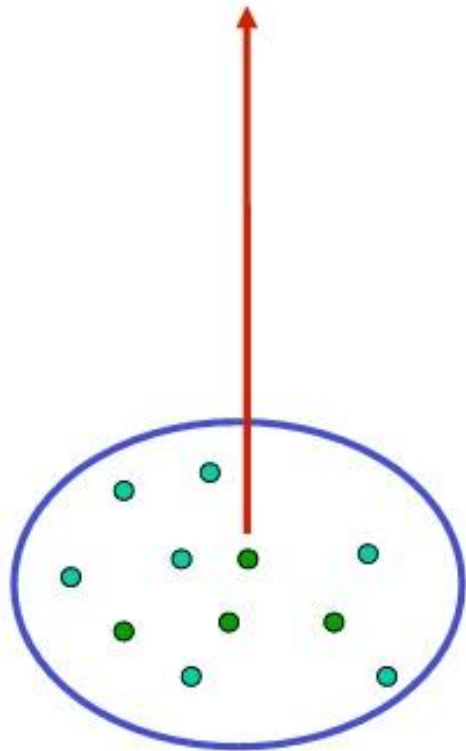


Final Ruleset

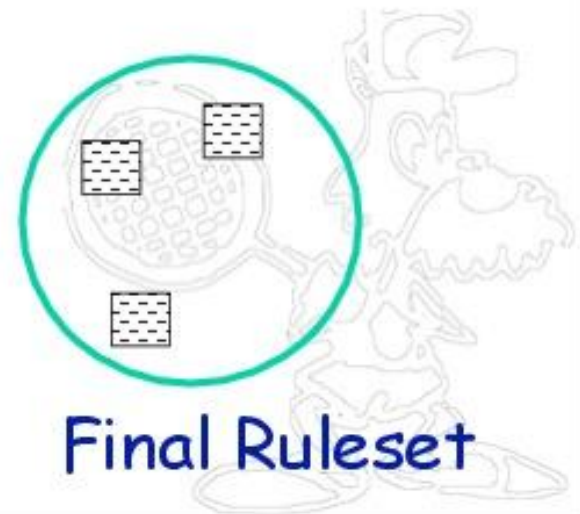


Rule Induction & Generalization

Rule 



Positive Examples

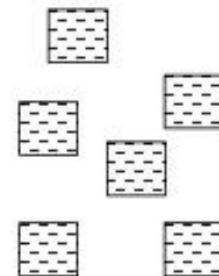


Final Ruleset

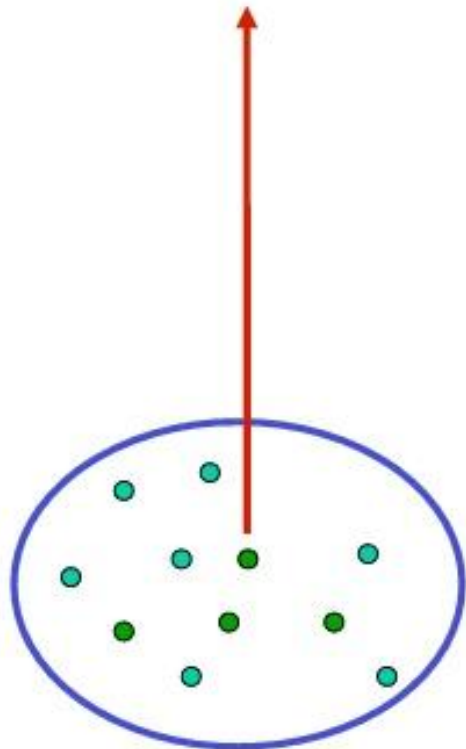


Rule Induction & Generalization

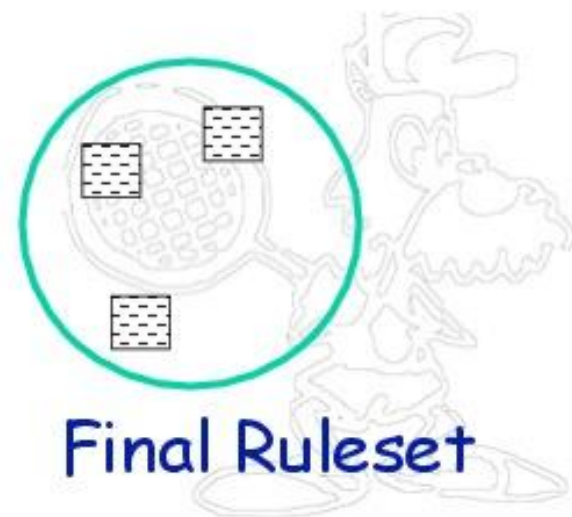
Rule



Generalizations



Positive Examples



Final Ruleset

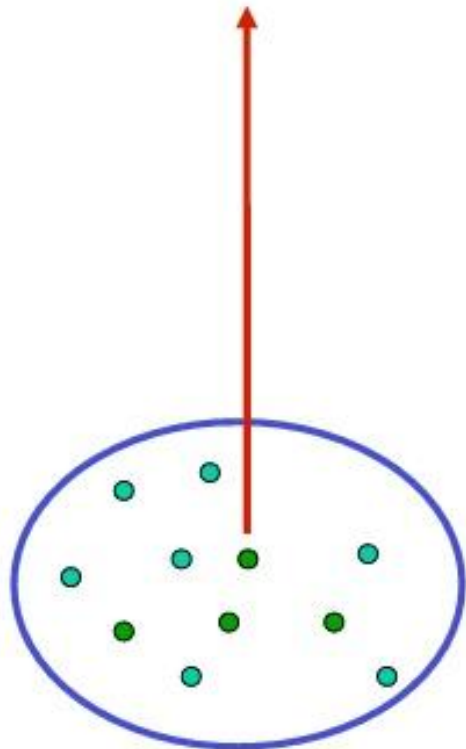


Rule Induction & Generalization

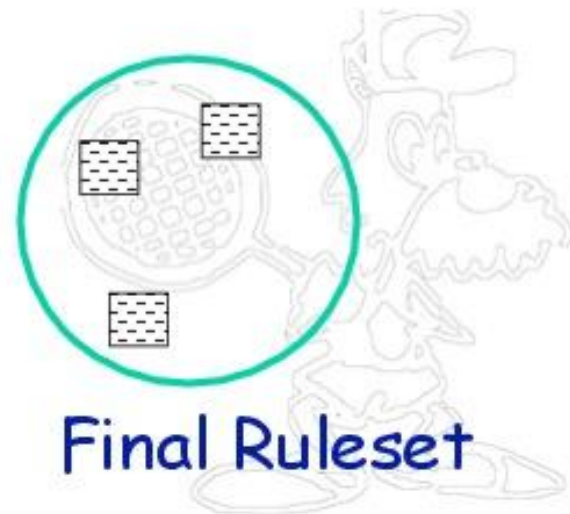
Rule



Generalizations



Positive Examples



Final Ruleset

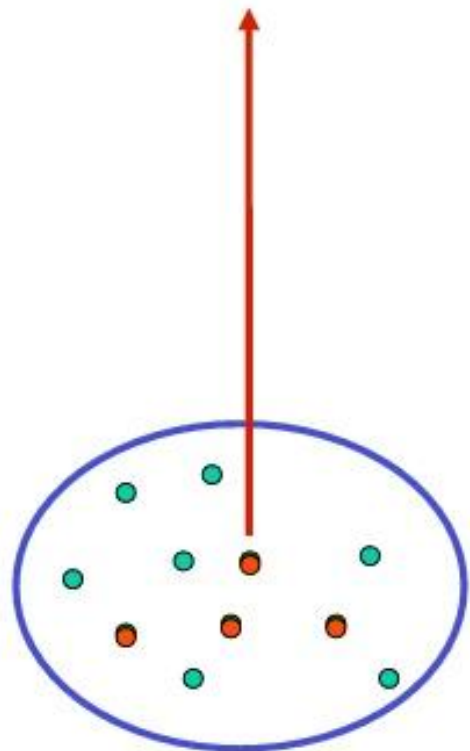


Rule Induction & Generalization

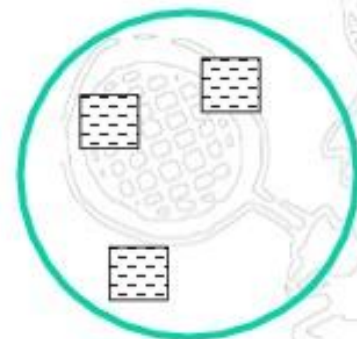
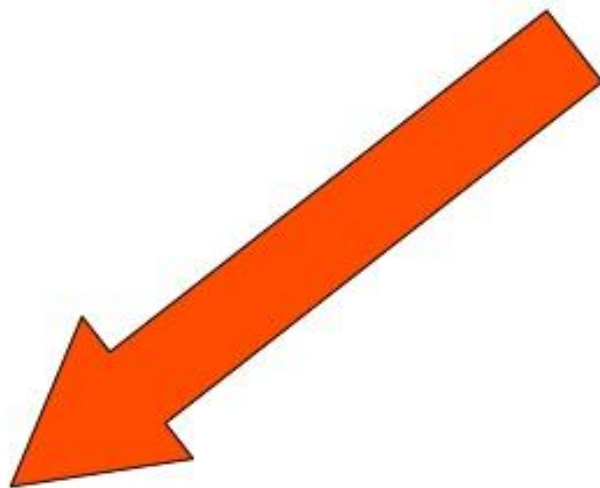
Rule



Generalizations



Positive Examples



Final Ruleset

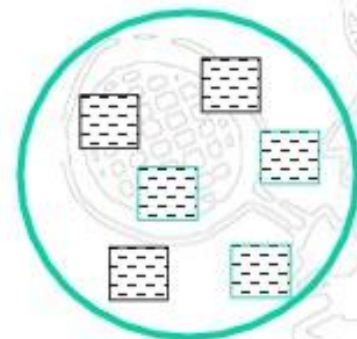
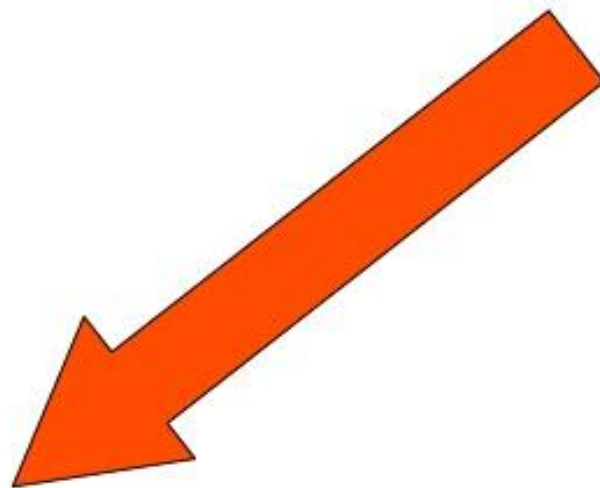
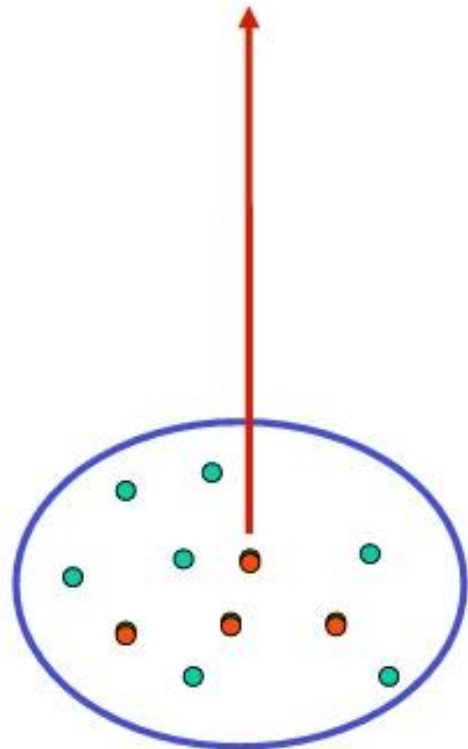


Rule Induction & Generalization

Rule



Generalizations

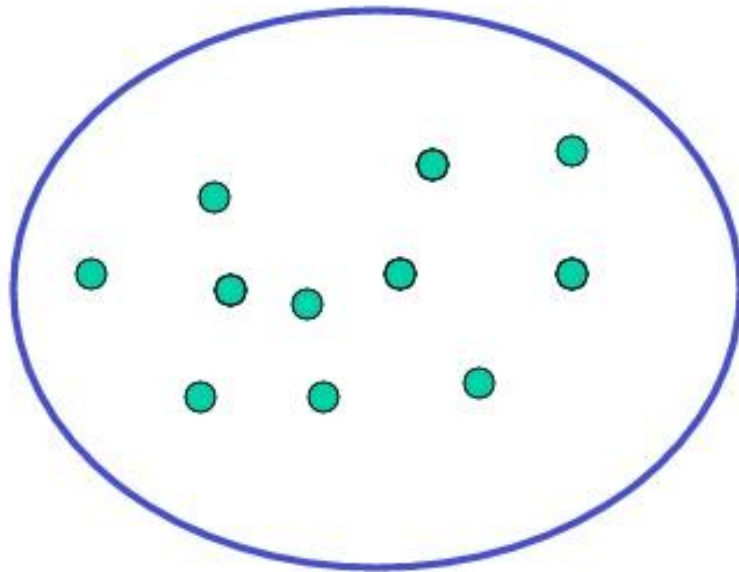


Positive Examples

Final Ruleset



Covering Algorithm

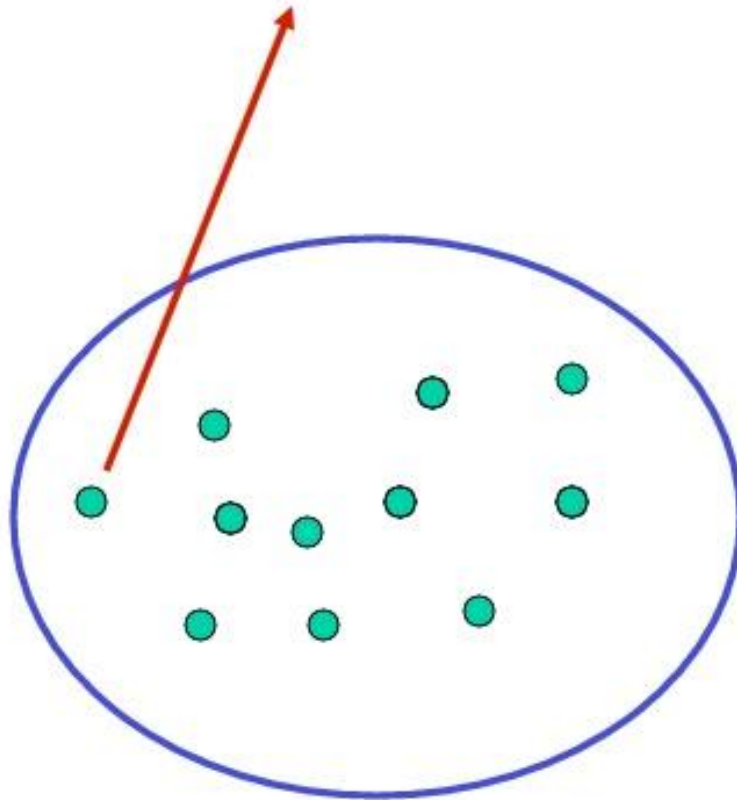


Positive Examples





Covering Algorithm

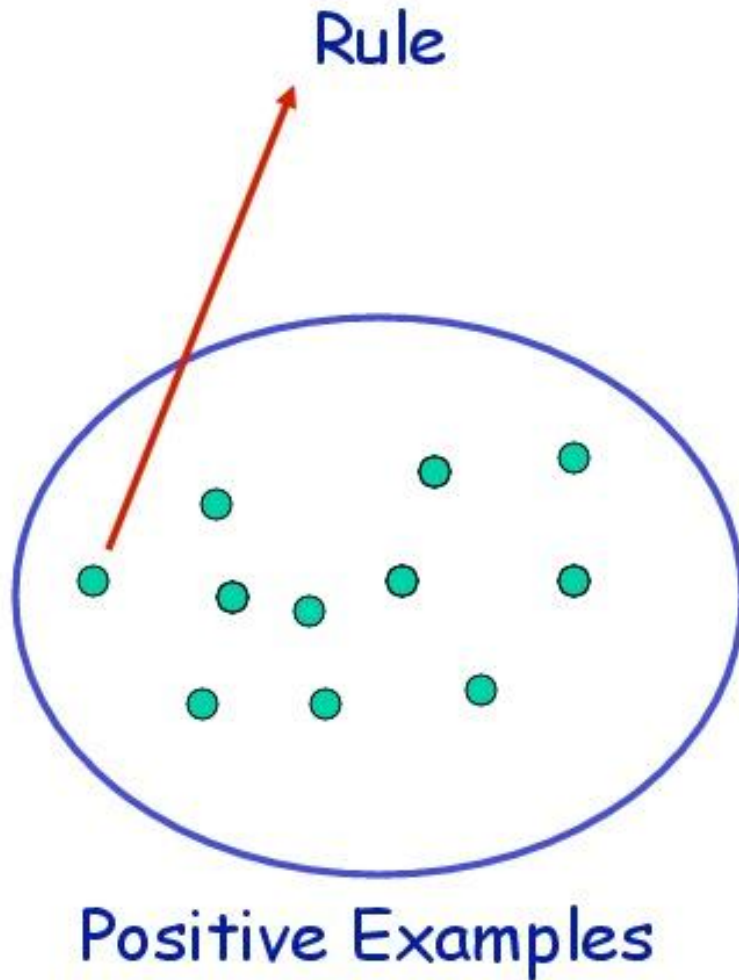


Positive Examples



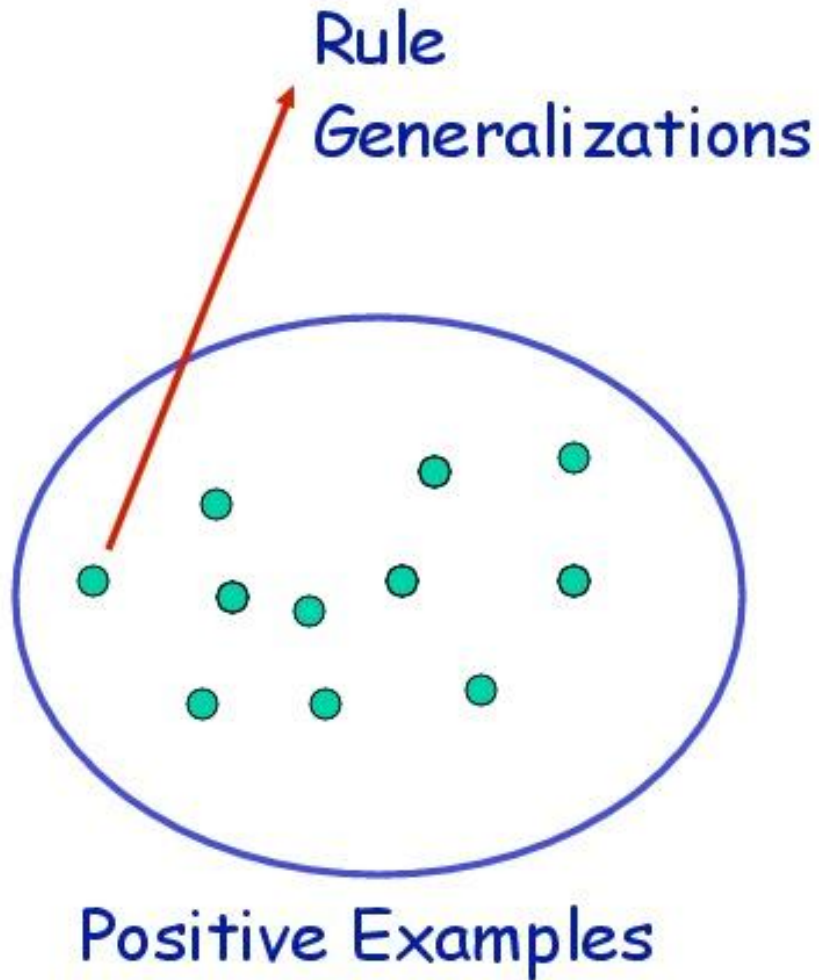


Covering Algorithm



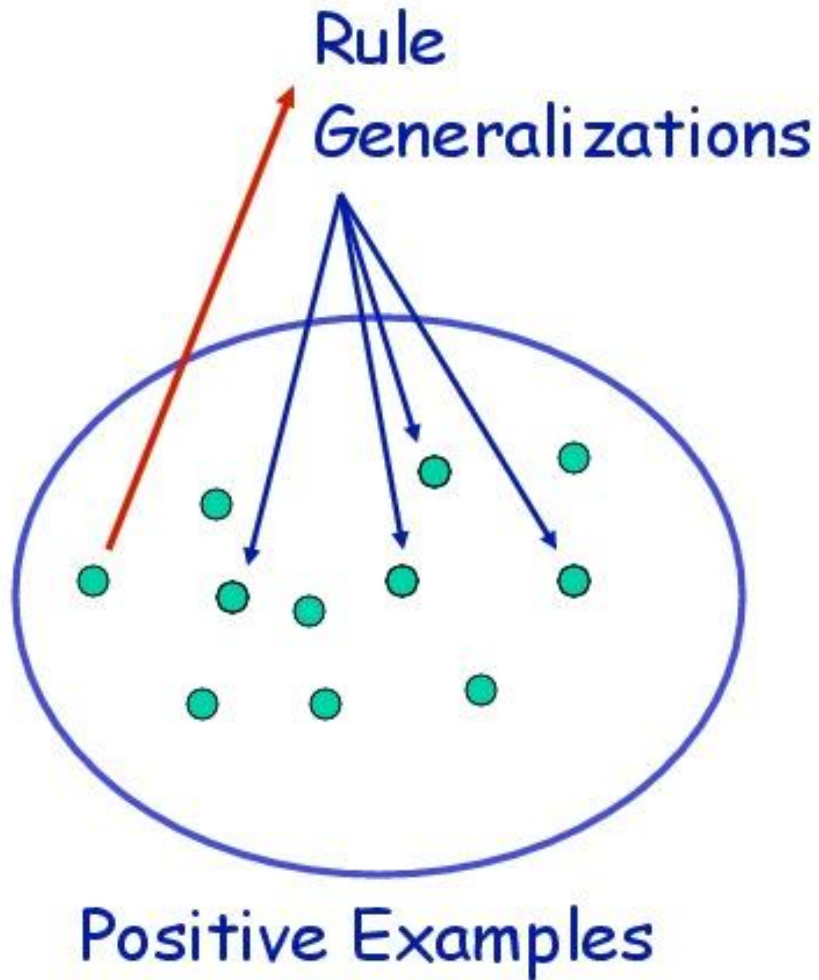


Covering Algorithm



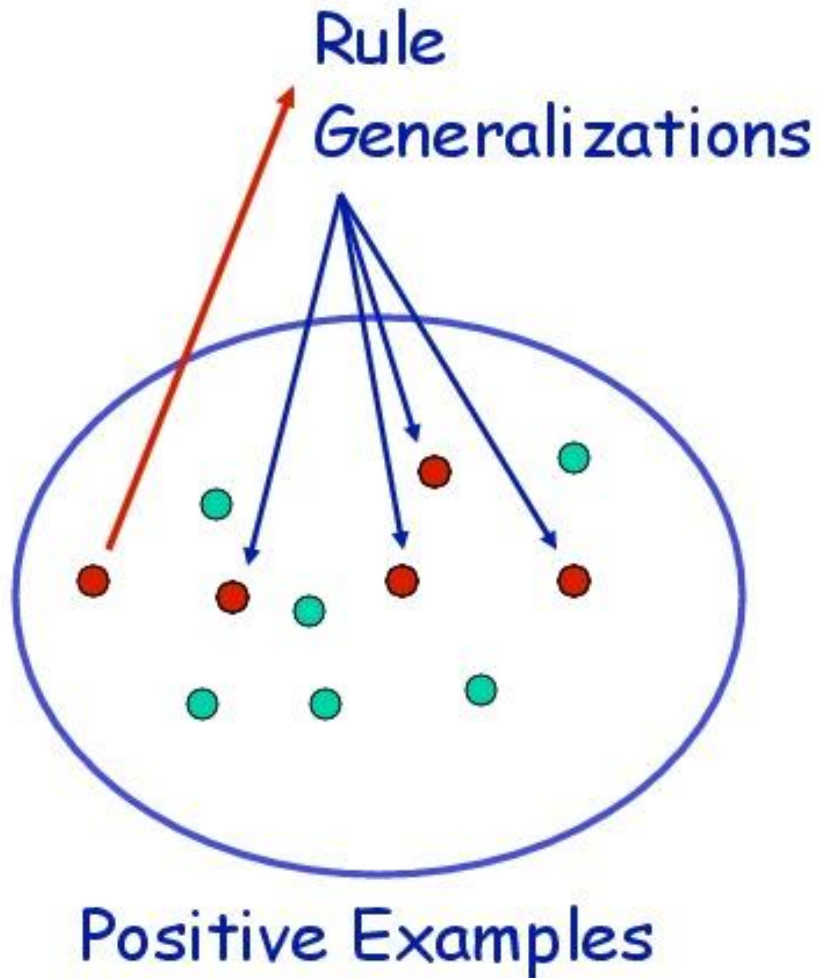


Covering Algorithm





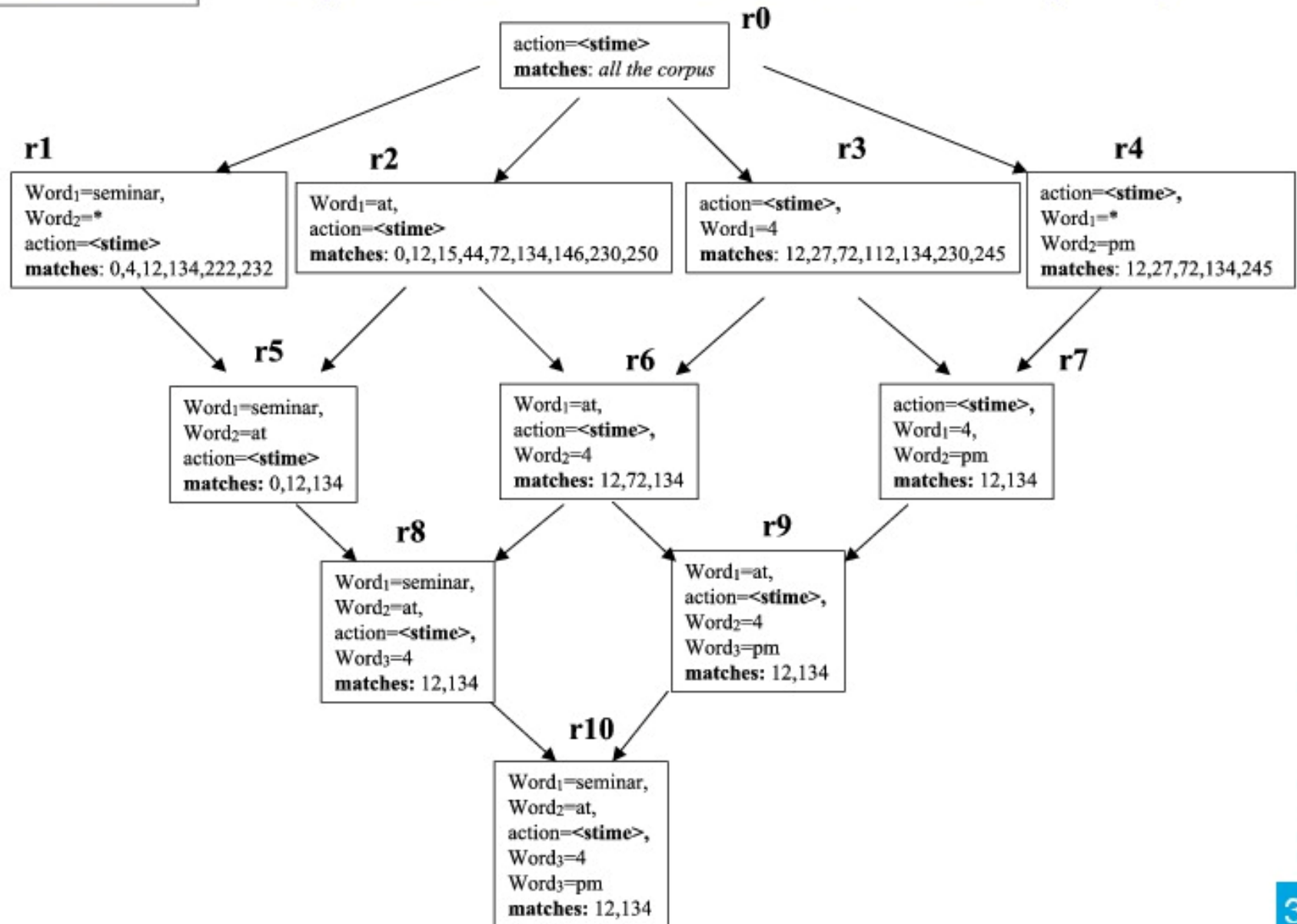
Covering Algorithm



Top Down Algorithm

- Starts from an empty rule
 - will match the whole corpus
 - 100% recall, low precision
- Progressively insert constraints on words to raise precision while keeping recall
 - Stop when rules are overfitting examples

Top-down version of (LP)²



- There are many papers on hand-crafted rule-based NER and learning rules for NER
 - Wikipedia also has a useful survey which I recommend
- Now we will return to evaluation
 - Short discussion of precision/recall as actually used in IE (not IR)
- Next time:
 - More on evaluation and rule-based NER
 - Annotation of training sets

Importance of Evaluation in IE

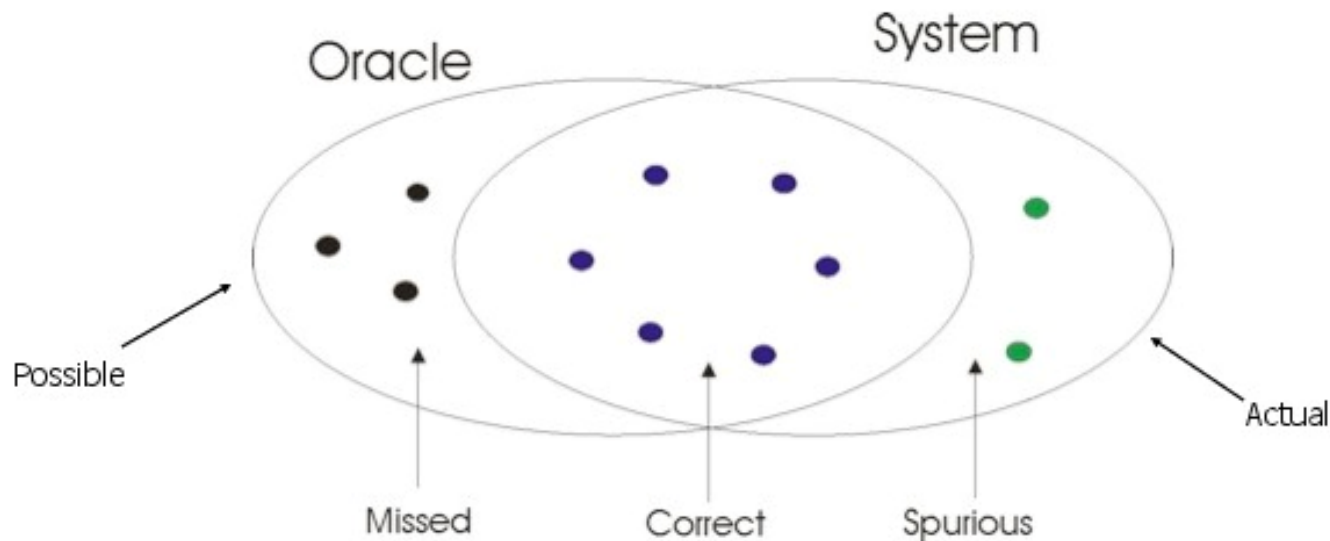
- IE was born from a series of competitive evaluations organised by DARPA in the US
 - MUC Conferences, 1989-1998
 - IE as a departure from IR but using the same types of measures of accuracy
 - The idea was to understand what worked and what not in text analysis
 - Finding a way to compare IE systems and approaches in a controlled way
- Evaluation is in IE's DNA
 - Publishing IE papers without evaluation is not considered acceptable

Organising Evaluation

- You will need:
 - An annotated training corpus
 - That you will use to develop rules or to train a machine learning algorithm
 - A result scorer
 - A tool that automatically computes accuracy of the system against an annotated corpus
 - E.g. The MUC Scorer
 - An annotated test corpus
 - To be used blindly to test results
 - Please note that run on test corpus should be a one off test
 - Test corpus is not be used to fine tuning accuracy in any way
 - E.g. By looking at the results and changing your rules or by tuning the learning parameters

The Rationale Behind

- **Precision:** how correct is the average answer provided by the system
- **Recall:** how many (correct) pieces of information are retrieved by the system
- **F-measure:** allows comparative evaluations



Evaluation Measures

$$\text{Recall} = \frac{\text{CORRECT} + (\text{PARTIAL} * 0.5)}{\text{POSSIBLE}}$$

$$\text{Precision} = \frac{\text{CORRECT} + (\text{PARTIAL} * 0.5)}{\text{ACTUAL}}$$

$$F(\beta) = \frac{(\beta^2 + 1) * \text{PREC} * \text{REC}}{\beta^2 * \text{PREC} + \text{REC}}$$

F-Measure is to be used to compare systems

In all evaluations all the three measures must be published

- Slide sources
 - Many of the slides today were from Fabio Ciravegna, University of Sheffield and Fabian Suchanek, France

- Thank you for your attention!
- (Don't forget the next Seminar meeting is in Gobi!)