Information Extraction

Theory and Practice

Ronen Feldman
Bar-Ilan University
ISRAEL
feldman@cs.biu.ac.il

What is Information Extraction?

- **IE does not indicate which documents need to be read by a user, it rather extracts pieces of information that are salient to the user's needs.**
- **Links between the extracted information and the original documents are maintained to allow the user to reference context.**
- **The kinds of information that systems extract vary in detail and reliability.**
- **Named entities such as persons and organizations can be extracted with reliability in the 90th percentile range, but do not provide attributes, facts, or events that those entities have or participate in.**

Relevant IE Definitions

- **Entity: an object of interest such as a person or organization.**
- **Attribute: a property of an entity such as its name, alias, descriptor, or type.**
- **Fact: a relationship held between two or more entities such as Position of a Person in a Company.**
- **Event: an activity involving several entities such as a terrorist act, airline crash, management change, new product introduction.**

IE Accuracy by Information Type

Information Type	Accuracy
Entities	90-98%
Attributes	80%
Facts	60-70%
Events	50-60%

MUC Conferences

Conference	Year	Topic
MUC 1	1987	Naval Operations
MUC 2	1989	Naval Operations
MUC 3	1991	Terrorist Activity
MUC 4	1992	Terrorist Activity
MUC 5	1993	Joint Venture and Micro Electronics
MUC 6	1995	Management Changes
MUC 7	1997	Spaces Vehicles and Missile Launches

The ACE Evaluation

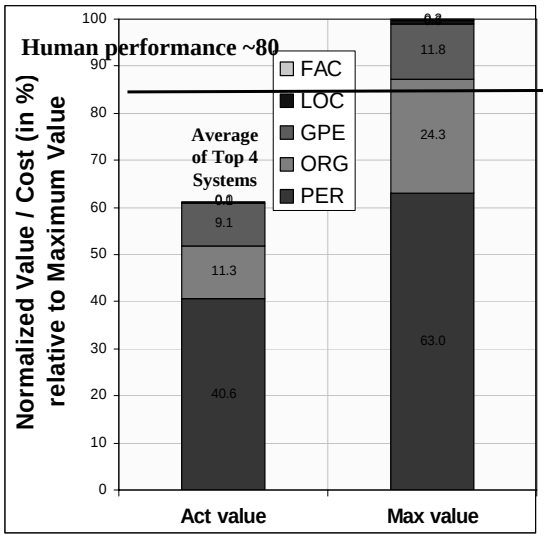
- **The ACE program is dedicated to the challenge of extracting content from human language. This is a fundamental capability that the ACE program addresses with a basic research effort that is directed to master first the extraction of "entities", then the extraction of "relations" among these entities, and finally the extraction of "events" that are causally related sets of relations.**
- **After two years of research on the entity detection and tracking task, top systems have achieved a capability that is useful by itself and that, in the context of the ACE EDT task, successfully captures and outputs well over 50 percent of the value at the entity level.**
- **Here value is defined to be the benefit derived by successfully extracting the entities, where each individual entity provides a value that is a function of the entity type (i.e., "person", "organization", etc.) and level (i.e., "named", "unnamed"). Thus each entity contributes to the overall value through the incremental value that it provides.**

ACE Entity Detection & Tracking Evaluation -- 2/2002

Goal: Extract entities. Each entity is assigned a value. This value is a function of its *Type* and *Level*. This value is gained when the entity is successfully detected. This value is lost when an entity is missed, spuriously detected, or mischaracterized.

Table of Entity Values

	PER	ORG	GPE	LOC	FAC
NAM	1	0.5	0.25	0.1	0.05
NOM	0.2	0.1	0.05	0.02	0.01
PRO	0.04	0.02	0.01	0.004	0.002



Miss 36%, False Alarm 22%, Type Error 6%

Applications of Information Extraction

- **Routing of Information**
- **Infrastructure for IR and for Categorization (higher level features)**
- **Event Based Summarization.**
- **Automatic Creation of Databases and Knowledge Bases.**

Where would IE be useful?

- **Semi-Structured Text**
- **Generic documents like News articles.**
- **Most of the information in the document is centered around a set of easily identifiable entities.**

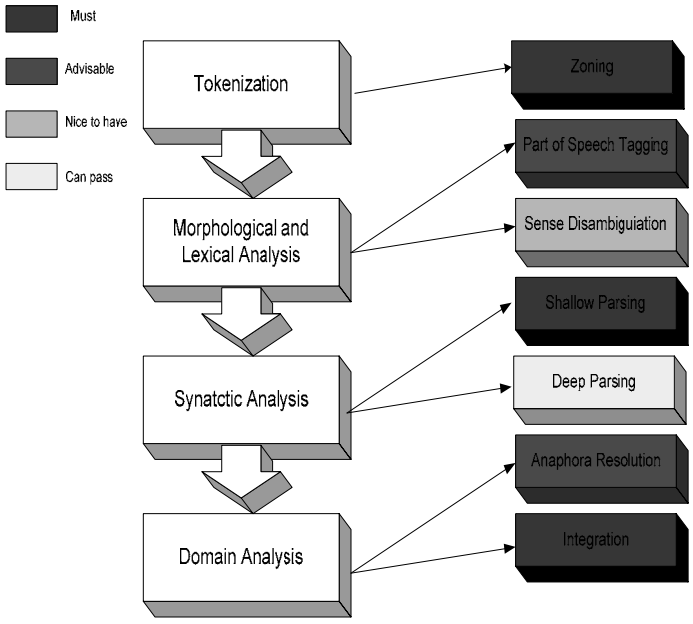
Approaches for Building IE Systems

- **Knowledge Engineering Approach**
  - **Rules are crafted by linguists in cooperation with domain experts.**
  - **Most of the work is done by inspecting a set of relevant documents.**
  - **Can take a lot of time to fine tune the rule set.**
  - **Best results were achieved with KB based IE systems.**
  - **Skilled/gifted developers are needed.**
  - **A strong development environment is a MUST!**

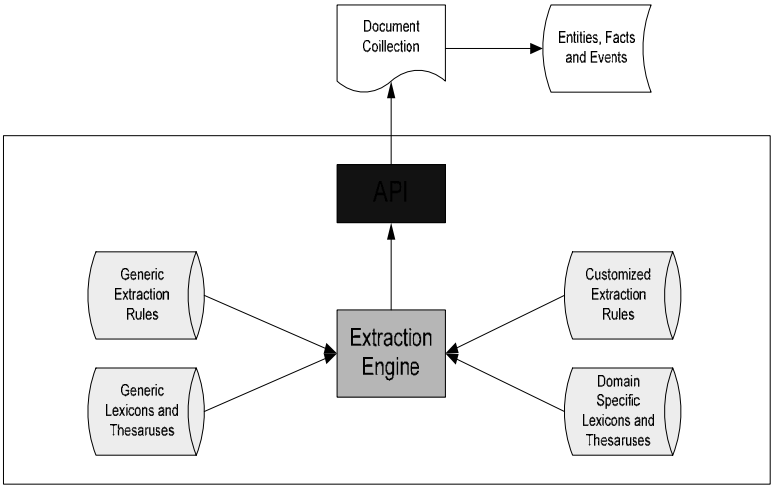
Approaches for Building IE Systems

- **Automatically Trainable Systems**
  - **The techniques are based on pure statistics and almost no linguistic knowledge**
  - **They are language independent**
  - **The main input is an annotated corpus**
  - **Need a relatively small effort when building the rules, however creating the annotated corpus is extremely laborious.**
  - **Huge number of training examples is needed in order to achieve reasonable accuracy.**
  - **Hybrid approaches can utilize the user input in the development loop.**

Components of IE System



The Extraction Engine



Why is IE Difficult?

- **Different Languages**
  - **Morphology is very easy in English, much harder in German and Hebrew.**
  - **Identifying word and sentence boundaries is fairly easy in European language, much harder in Chinese and Japanese.**
  - **Some languages use orthography (like english) while others (like hebrew, arabic etc) do no have it.**
- **Different types of style**
  - **Scientific papers**
  - **Newspapers**
  - **memos**
  - **Emails**
  - **Speech transcripts**
- **Type of Document**
  - **Tables**
  - **Graphics**
  - **Small messages vs. Books**

Morphological Analysis

- **Easy**
  - **English, Japanese**
  - **Listing all inflections of a word is a real possibility**
- **Medium**
  - **French Spanish**
  - **A simple morphological component adds value.**
- **Difficult**
  - **German, Hebrew, Arabic**
  - **A sophisticated morphological component is a must!**

Using Vocabularies

- **“Size doesn’t matter”**
  - **Large lists tend to cause more mistakes**
  - **Examples:**
    - **Said as a person name (male)**
    - **Alberta as a name of a person (female)**
- **It might be better to have small domain specific dictionaries**

Part of Speech Tagging

- **POS can help to reduce ambiguity, and to deal with ALL CAPS text.**
- **However**
  - **It usually fails exactly when you need it**
  - **It is domain dependent, so to get the best results you need to retrain it on a relevant corpus.**
  - **It takes a lot of time to prepare a training corpus.**

A simple POS Strategy

- **Use a tag frequency table to determine the right POS.**
  - **This will lead to elimination of rare senses.**
- **The overhead is very small**
- **It improve accuracy by a small percentage.**
- **Compared to full POS it provide similar boost to accuracy.**

Dealing with Proper Names

- **The problem**
  - **Impossible to enumerate**
  - **New candidates are generated all the time**
  - **Hard to provide syntactic rules**
- **Types of proper names**
  - **People**
  - **Companies**
  - **Organizations**
  - **Products**
  - **Technologies**
  - **Locations (cities, states, countries, rivers, mountains)**

Comparing RB Systems with ML Based Systems

	Rule Based	HMM
Wall Street Journal		
MUC6	**96.4**	**93**
MUC7	**93.7**	**90.4**
Transcribed Speech		
HUB4	**90.3**	**90.6**

Building a RB Proper Name Extractor

- A Lexicon is always a good start
- The rules can be based on the lexicon and on:
  - The context (preceding/following verbs or nouns)
  - Regular expressions
    - Companies: Capital* [,] inc, Capital* corporation..
    - Locations: Capital* Lake, Capital* River
  - Capitalization
  - List structure
- After the creation of an initial set of rules
  - Run on the corpus
  - Analyze the results
  - Fix the rules and repeat…
- This Process can take around 2-3 weeks and result in performance of between 85-90% break even.
- Better performance can be achieved with more effort (2-3 months) and then performance can get to 95-98%

DIAL – Declarative Information Analysis Language

An Information Extraction Language

The DIAL4 Language

D**eclarative** I**nformation** A**nalysis** L**anguage**

- **Modular**
- **Entity Oriented**
- **Regular Expressions Based**
- **Rapid and Easy RuleBook Development**

DIAL4 Concepts

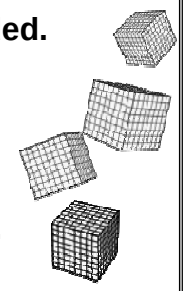
Each concept may have:

CONCEPTS

Attributes
Guards
a set of logical conditions on the concept attributes' values
Actions
a set of operations to perform after finding a concept instance
Internal
a section for defining internal concepts which can only be used within the scope of the concept
Functions
a section for defining add-on (Perl) functions that can only be used within the scope of the concept
Context
text units in which concepts instances will be searched

DIAL4 Rule Structure

- Pattern
  defines the text pattern to match when searching for a concept instance.
- Constraints
  defines logical conditions to apply to values extracted from the pattern match. If these conditions are not met the match is discarded.
- Actions
  a set of operations to perform after finding a pattern match. This is where concept instances are added to the Shared Memory.



A Full Rule – an Example

```
concept Company {};

rule Company {
pattern:
   (Capital+) -> name wcCompanyExt “.”?;
constraints:
   !(name.FirstToken() IS_IN wcCompNameNonStarters);
actions:
      Add();
};
```

Example of an Instance:

Crown Central Petroleum Corp.

Improving the Person Concept:
Assigning values to the concept attributes

```
concept Person {
attributes:
    string FirstName;
    string MiddleName;
    string LastName;
};

rule Person {
pattern:
   wcFirstName -> first Capital -> last;
actions:
   Add(FirstName<-first, LastName<-last);
};
```


Rule for Extraction of Person Names Based on Title/Position

```
wordclass wcPosition = adviser minister spokesman
                       president (vice president) general (gen .);
/* note that wordclass members are tokenized and entries containing multiple
    tokens should be enclosed within () */
concept PersonNameStruct {          //we define this concept to allow the code reuse
attributes:
    string FirstName;
    string MiddleName;
    string LastName;
};
wordclass wcNamePrefix = ben abu abed von al;
rule PersonNameStruct {
pattern:
    Capital -> first (Capital ".")? -> middle ((wcNamePrefix "-")? Capital) ->last;
actions:
    Add(FirstName <- first.Text(), MiddleName <- middle.Text(), LastName <- last.Text());
};

rule Person {
pattern:
    LONGEST(wcPosition PersonNameStruct -> name);
actions:
    Add(FirstName <- name.FirstName, MiddleName <- name.MiddleName,
        LastName <- name.LastName);
};
```

List of Person Names

```
concept PersonsList{};
   wordclass wcCriminalIndicatingVerbs = charged blamed arrested;
    wordclass wcPluralBe = are were;
    rule PersonsList {
    pattern:
          wcCriminalIndicatingVerbs wcPluralBe
          (PersonNameStruct->> pList ",")?+  "and"
           PersonNameStruct ->> pList;
    actions:
          iterate (pList)  begin
            currPerson = pList.CurrentItem();
            Add(Person, currPerson, FirstName<-currPerson.FirstName,
                LastName <- currPerson.LastName);
          end
    };
```


Person Concept: Applying Constraints

We can conclude with high probability that a proper name is a person name if it has a known first name or a middle name:

- John **Smith**
- **Emil** I. **Singer**

```
rule Person {
pattern:
          LONGEST(Capital -> first (MiddleName -> middle)?
          ((wcNamePrefix "-")? Capital) ->last);
constraints:
          (first IS_IN wcFirstNames) OR !(middle.IsEmpty());

          //in "President V. Putin" don't match "President" as the first name.
          !(first.FirstToken() IS_IN wcPosition);
actions:
          Add(FirstName <- first.Text(), MiddleName <- middle.Text(),
                LastName <- last.Text());
           Block(Person, this_match);
};
```

Person Concept: Applying Constraints (cont.)

We have two rules for concept Person: the first rule extracts names with sure internal evidence (first or middle name), and the second extracts names without internal evidence but with positions/titles preceding them. Let's merge these two rules into a single rule.

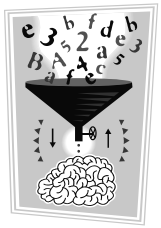
```
rule Person {
pattern:
        Capital -> first (MiddleName -> middle)?
        ((wcNamePrefix "-"?)? Capital) ->last;
constraints:
        (first IS_IN wcFirstNames) OR !(middle.IsEmpty())
        OR (first {1} AFTER wcPosition);

        !(first.FirstToken() IS_IN wcPosition);
actions:
        Add(FirstName <- first.Text(), MiddleName <- middle.Text(),
            LastName <- last.Text());
};
```


Concept Guards

- **Guards are applied to concept attributes when a rule attempts to add a concept instance to the Shared Memory.**
  A concept instance will be added only if all guard conditions are met .
- **Guards enable the concept to ensure conditions on its attribute values in a central location, without having to add these conditions to each rule of the concept.**

Example:

```
concept Date {
attributes:
        number nDay;
        number nMonth;
        number nYear;
guards:
        (nDay >= 1) AND (nDay <= 31);
        (nMonth >= 1) AND (nMonth <=12);
        (nYear > 0);
};
```



Introduction to HMMs for IE

Motivation

- **We can view the named entity extraction as a classification problem, where we classify each word as belonging to one of the named entity classes or to the no-name class.**
- **One of the most popular techniques for dealing with classifying sequences is HMM.**
- **Example of using HMM for another NLP classification task is that of part of speech tagging (Church, 1988 ; Weischedel et. al., 1993).**

What is HMM?

- **HMM (Hidden Markov Model) is a finite state automaton with stochastic state transitions and symbol emissions (Rabiner 1989).**
- **The automaton models a probabilistic generative process.**
- **In this process a sequence of symbols is produced by starting in an initial state, transitioning to a new state, emitting a symbol selected by the state and repeating this transition/emission cycle until a designated final state is reached.**

Disadvantage of HMM

- **The main disadvantage of using an HMM for Information extraction is the need for a large amount of training data. i.e., a carefully tagged corpus.**
- **The corpus needs to be tagged with all the concepts whose definitions we want to learn.**

Notational Conventions

- **T = length of the sequence of observations (training set)**
- **N = number of states in the model**
- **q_t = the actual state at time *t***
- **S = {S_1,...S_N} (finite set of possible states)**
- **V = {O_1,...O_M} (finite set of observation symbols)**
- **π = {π_i} = {*P*($q_1 = S_i$)} starting probabilities**
- **A = {*aij*}=P($q_{t+1} = S_i \mid q_t = S_j$) transition probabilities**
- **B = {$b_i(O_t)$} = {P($O_t \mid q_t = S_i$)} emission probabilities**

The Classic Problems Related to HMMs

- **Find *P*(O | λ): the probability of an observation sequence given the HMM model.**
- **Find the most likely state trajectory given λ and O.**
- **Adjust λ = (π, A, B) to maximize *P*(O | λ).**

Calculating $P(O \mid \lambda)$

- **The most obvious way to do that would be to enumerate every possible state sequence of length T (the length of the observation sequence). Let Q = Q1,...QT , then by assuming independence between the states we have**
  - **P(O|Q, λ) =** $\prod_{i=1}^{T} P(O_i \mid q_i, \lambda) = \prod_{i=1}^{T} b_{q_i}(O_i)$
  - **P(Q|λ) =** $\pi_{q_1} \prod_{i=1}^{T-1} a_{q_i q_{i+1}}$
- **By using Bayes theorem we have**
  - **P(O,Q|λ) = P(O|Q, λ) P(Q|λ)**
- **Finally The main problem with this is that we need to do $2TN^T$ multiplications, which is certainly not feasible even for a modest T like 10.**

The forward-backward algorithm

- **In order to solve that we use the forward-backward algorithm this is far more efficient. The forward part is based on the computation of terms called the alpha terms. We define the alpha values as follows,**

$$\alpha_1(i) = \pi_i b_i(O_1)$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1})$$

$$P(O \mid \lambda) = \sum_{i=1}^{N} a_T(i)$$

- **We can compute the alpha values inductively in a very efficient way.**
- **This calculation requires just N^2T multiplications.**

The backward phase

- **In a similar manner we can define a backward variable called beta that computes the probability of a partial observation sequence from t+1 to T. The beta variable will also be computed inductively but in a backward fashion.**

$$\beta_T(i) = 1$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_i(O_{t+1}) \beta_{t+1}(j)$$

Solution for the second problem

- **Our main goal is to find the "optimal" state sequence. We will do that by maximizing the probabilities of each state individually.**
- **We will start by defining a set of gamma variables the measure the probabilities that at time t we are at state S_i.**

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

- **The denominator is used just to make gamma a true probability measure.**
- **Now we can find the best state at each time slot in a local fashion.**

$$q_t = \underset{1 \le i \le N}{\arg\max}[\gamma_t(i)]$$

- **The main problem with this approach is that the optimization is done locally, and not on the whole sequence of states. This can lead either to a local maximum or even to an invalid sequence. In order to solve that problem we use a well known dynamic programming algorithm called the Viterbi algorithm.**

The Viterbi Algorithm

- Intuition
  - **Compute the most likely sequence starting with the empty observation sequence; use this result to compute the most likely sequence with an output sequence of length one; recurse until you have the most likely sequence for the entire sequence of observations.**
- Algorithmic Details
  - **The delta variables compute the highest probability of a partial sequence up to time *t* that ends in state *Si*. The psi variables enables us to accumulate the best sequence as we move along the time slices.**
- **1. Initialization:**

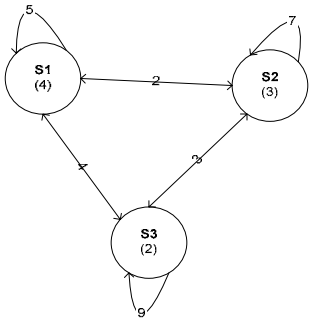
$$\delta_1(i) = \pi_i b_i(O_1)$$
$$\psi_1(i) = 0$$

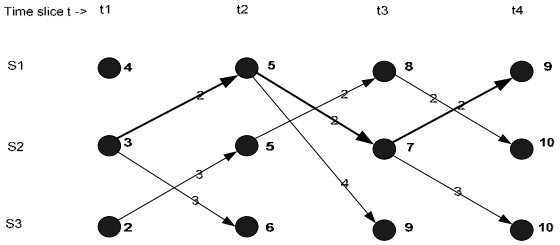
Viterbi (Cont).

- **Recursion:** $\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_i(O_t)$
- **Termination:** $\psi_t(j) = \arg\max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right]$
- **Reconstruction:** $P^* = \max_{1 \le i \le N} \left[\delta_T(i)\right]$ $q_T^* = \arg\max_{1 \le i \le N} \left[\delta_T(i)\right]$
- **For t = T-1,T-2,...,1.** $q_t^* = \psi_{t+1}(q_{t+1}^*)$

The resulting sequence, $q_1^*, q_2^* \cdots q_T^*$ **, solves Problem 2.**

Viterbi (Example)

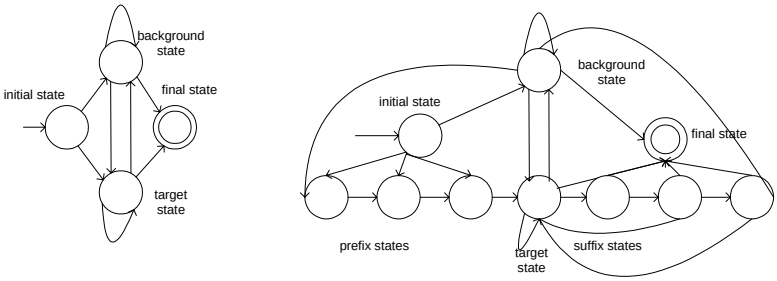




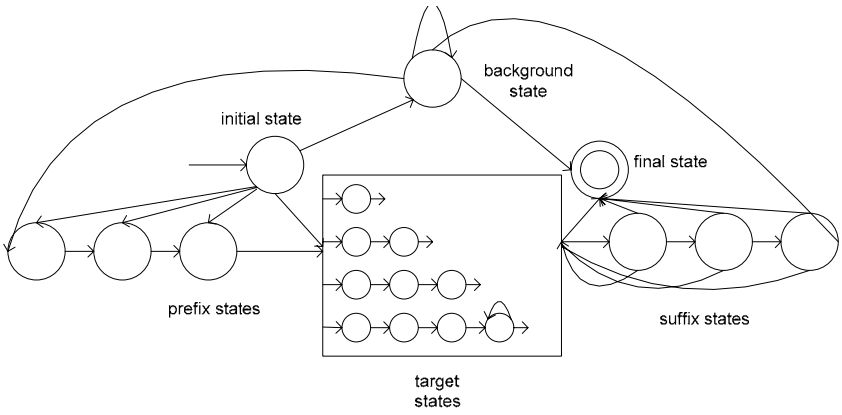
The Just Research HMM

- **Each HMM extracts just one field of a given document. If more fields are needed, several HMMs need to be constructed.**
- **The HMM takes the entire document as one observation sequence.**
- **The HMM contains two classes of states, background states and target states. The background states emit words in which are not interested, while the target states emit words that constitute the information to be extracted.**
- **The state topology is designed by hand and only a few transitions are allowed between the states.**

Possible HMM Topologies



A more General HMM Architecture



Experimental Evaluation

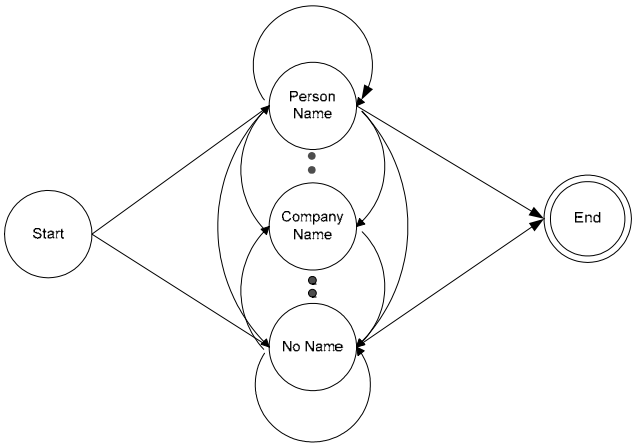
Acquiring Company	Acquired Company	Abbreviation of Acquired Company	Price of Acquisition	Status of Acquisition
30.9%	48.1%	40.1%	55.3%	46.7%

Speaker	Location	Start Time	End Time
71.1%	83.9%	99.1%	59.5%

BBN's Identifinder

- An ergodic bigram model.
- Each Named Class has a separate region in the HMM.
- The number of states in each NC region is equal to |V|. Each word has its own state.
- Rather then using plain words, extended words are used. An extended word is a pair <w,f>, where f is a feature of the word w.

BBN's HMM Architecture



Possible word Features

1. **2 digit number (01)**
2. **4 digit number (1996)**
3. **alphanumeric string (A34-24)**
4. **digits and dashes (12-16-02)**
5. **digits and slashes (12/16/02)**
6. **digits and comma (1,000)**
7. **digits and period (2.34)**
8. **any other number (100)**
9. **All capital letters (CLF)**
10. **Capital letter and a period (M.)**
11. **First word of a sentence (The)**
12. **Initial letter of the word is capitalized (Albert)**
13. **word in lower case (country)**
14. **all other words and tokens (;)**

Statistical Model

- **The design of the formal model is done in levels.**
- **At the first level we have the most accurate model, which require the largest amount of training data.**
- **At the lower levels we have back-off models that are less accurate but also require much smaller amounts of training data.**
- **We always try to use the most accurate model possible given the amount of available training data.**

Computing State Transition Probabilities

- **When we want to analyze formally the probability of annotating a given word sequence with a set of name classes, we need to consider three different statistical models:**
  - **A model for generating a name class**
  - **A model to generate the first word in a name class**
  - **A model to generate all other words (but the first word) in a name class**

Computing the Probabilities : Details

- **The model to generate a name class depends on the previous name class and on the word that precedes the name class; this is the last word in the previous name class and we annotate it by w_{-1}. So formally this amounts to $P(NC \mid NC_{-1}, w_{-1})$.**
- **The model to generate the first word in a name class depends on the current name class and the previous name class and hence is $P(<w,f>_{first} \mid NC, NC_{-1})$.**
- **The model to generate all other words within the same name class depends on the previoues word (within the same name class) and the current name class, so formally it is $P(<w,f> \mid <w,f>_{-1}, NC)$.**

The Actual Computation

$$P(NC \mid NC_{-1}, w_{-1}) = \frac{c(NC, NC_{-1}, w_{-1})}{c(NC_{-1}, w_{-1})}$$

$$P(< w, f >_{first} \mid NC, NC_{-1}) = \frac{c(< w, f >_{first}, NC, NC_{-1})}{c(NC, NC_{-1})}$$

$$P(< w, f > \mid < w, f >_{-1}, NC) = \frac{c(< w, f >, < w, f >_{-1}, NC)}{c(< w, f >_{-1}, NC)}$$

c(<w,f>,<w,f>-1,NC), counts the number of times that we have the pair <w,f> after the pair $<w,f>_{-1}$ and they both are tagged by the name class NC.

Modeling Unknown Words

- **The main technique is to create a new entity called UNKNOWN (marked _UNK_), and create statistics for that new entity. All words that were no seen before are mapped to _UNK_.**
- **split the collection into 2 even parts, and each time use one part for training and one part as a hold out set. The final statistics is the combination of the results from the two runs.**
- **The statistics needs to be collected for 3 different classes of cases: _UNK_ and then a known word (|V| cases), a known word and then _UNK_ and two consecutive _UNK_ words. This statistics is collected for each name class.**

Name Class Back-off Models

- **The full model take into account both the previous name class and the previous word ($P(NC| NC_{-1},w_{-1})$)**
- **The first back-off model takes into account just the previous name class ($P((NC| NC_{-1}))$).**
- **The next back-off model would just estimate the probability of seeing the name class based on the distribution of the various name classes (P(NC)).**
- **Finally, we use a uniform distribution between all names classes (1/(N+1), where N is number of the possible name classes)**

First Word Back-off Models

- **The full model takes into account the current name class and the previous name class ($P(<w,f>_{first} | NC, NC_{-1})$).**
- **The first back-off model takes into account just the current name class ($P(<w,f>_{first} | NC)$).**
- **The next back-off model, breaks the <w,f> pair and just uses multiplication of two independent events given the current word class ($P(w|NC)P(f|NC)$)**
- **The next back-off model is a uniform distribution between all pairs of words and features ($\frac{1}{|V||F\#|}$, where F# is the # of possible word features)**

Rest of the Words Back-off Models

- **The full model takes into account the current name class and the previous word ($P(<w,f>|<w,f>_{-1}, NC)$).**
- **The first back-off model takes into account just the current name class ($P(<w,f>| NC)$).**
- **The next back-off model, breaks the <w,f> pair and just uses multiplication of two independent events given the current word class ($P(w|NC)P(f|NC)$)**
- **The next back-off model is a uniform distribution between all pairs of words and features ($\frac{1}{|V||F\#|}$, where F# is the # of possible word features)**

Combining all the models

- **The actual probability is a combination of the different models. Each model gets a different weight based on the amount of training available to that model.**
- **Lets assume we have 4 models (one full model, and 3 back-off models), and we are trying to estimate the probability of P(X|Y). Let P1 be probability of the event according to the full model, and P2, P3, P4 ate the back-off models respectively.**
- **The weights are computed based on a lambda parameter that is based on each model and it immediate back-off model. For instance λ_1 will adjust the wait between the full model and the first back-off model.**

$$\lambda = \left(1 - \frac{c(Y)}{bc(Y)}\right) \frac{1}{1 + \frac{\#(Y)}{bc(Y)}}$$

Analysis

- **Where c(Y) the count of event Y according to the full model, and bc(Y) is the count of event Y according to the back-off model. #(Y) is the number of unique outcomes of Y.**
- **Lambda has two desirable properties**
  - **If the full model and the back-off model both have the same support for event Y, then Lambda will be 0 and we will use just the full model.**
  - **If the possible outcomes of Y are distributed uniformly then the weight of lambda will be close to 0 since there is low confidence in the back-off model.**

$$\lambda = \left(1 - \frac{c(Y)}{bc(Y)}\right) \frac{1}{1 + \frac{\#(Y)}{bc(Y)}}$$

Example

- **We want to compute the probability of P("bank" | "river", "Not-A-Name"). Lets assume that river appears with 3 different words in the Not-A-Name" name class, and in total there are 9 different occurrences of river with any of the 3 words.**

$$\lambda_1 = \left(1 - \frac{0}{9}\right)\left(\frac{1}{1+\frac{3}{9}}\right) = 1 \cdot \frac{3}{4} = \frac{3}{4}$$

- **so we will use the full model (P1) with 0.75, and the other back-off models with 0.25. We then compute λ2 which computes the weight of the first back-off model (P2) against the other back-off models, and finally λ3 which is the weight of the second back-off model (P3) against the last back-off model. So to sum up, the probability of P(X|Y) would be:**
- **P(X|Y) = λ1 * P1(X|Y) + (1 - λ1) * (λ2 * P2(X|Y) + (1 - λ2) * (λ3 * P3(X|Y) + (1 - λ3) * P4(X|Y)))**

Using different modalities of text

- Mixed Case: **Abu Sayyaf carried out an attack on a south western beach resort on May 27, seizing hostages including three Americans. They are still holding a missionary couple, Martin and Gracia Burnham, from Wichita, Kansas, and claim to have beheaded the third American, Guillermo Sobero, from Corona, California. Mr. Sobero's body has not been found.**
- Upper Case: **ABU SAYYAF CARRIED OUT AN ATTACK ON A SOUTH WESTERN BEACH RESORT ON MAY 27, SEIZING HOSTAGES INCLUDING THREE AMERICANS. THEY ARE STILL HOLDING A MISSIONARY COUPLE, MARTIN AND GRACIA BURNHAM, FROM WICHITA, KANSAS, AND CLAIM TO HAVE BEHEADED THE THIRD AMERICAN, GUILLERMO SOBERO, FROM CORONA, CALIFORNIA. MR SOBERO'S BODY HAS NOT BEEN FOUND.**
- SNOR: **ABU SAYYAF CARRIED OUT AN ATTACK ON A SOUTH WESTERN BEACH RESORT ON MAY TWENTY SEVEN SEIZING HOSTAGES INCLUDING THREE AMERICANS THEY ARE STILL HOLDING A MISSIONARY COUPLE MARTIN AND GRACIA BURNHAM FROM WICHITA KANSAS AND CLAIM TO HAVE BEHEADED THE THIRD AMERICAN GUILLERMO SOBERO FROM CORONA CALIFORNIA MR SOBEROS BODY HAS NOT BEEN FOUND.**

Experimental Evaluation (MUC 7)

Modality	Language	Rule Based	HMM
Mixed case	**English**	**96.4%**	**94.9%**
Upper case	**English**	**89%**	**93.6%**
SNOR	**English**	**74%**	**90.7%**
Mixed case	**Spanish**	**93%**	**90%**

How much Data is needed to train an HMM?

Number of Tagged Words	English	Spanish
23,000	**NA**	**88.6%**
60,000	**91.5%**	**89.7%**
85,000	**91.9%**	**NA**
130,000	**92.8%**	**90.5%**
230,000	**93.1%**	**91.2%**
650,000	**94.9%**	**NA**

Limitations of the Model

- **The context which is used for deciding on the type of each word is just the word the precedes the current word. In many cases, such a limited context may cause classification errors.**
- **As an example consider the following text fragment "The Turkish company, Birgen Air, was using the plane to fill a charter commitment to a German company,". The token that precedes Birgen is a comma, and hence we are missing the crucial clue** company **which is just one token before the comma.**
- **Due to the lack of this hint, the IndentiFinder system classified Birgen Air as a location rather than as a company. One way to solve this problem is to augment the model with another token when the previous token is a punctuation mark.**

Example - input

<DOCUMENT>
<TYPE>NEWS</TYPE>
<ID> 4 Ryan daughters tied to cash (Thu Feb 13, 9:49 AM)</ID>
<TITLE> 4 Ryan daughters tied to cash </TITLE>
<DATE> Thu Feb 13, 9:49 AM </DATE>
<SOURCE> Yahoo-News </SOURCE>
<BODY> By Matt O'Connor, Tribune staff reporter. Tribune staff reporter Ray Gibson contributed to this report <p> Four of former Gov. George Ryan's daughters shared in almost 10,000 in secret payments from Sen. Phil Gramm's presidential campaign in the mid-1990s, according to testimony Wednesday in federal court. <p>
Alan Drazek, who said he was brought in at Ryan's request as a point man for the Gramm campaign in Illinois, testified he was told by Scott Fawell, Ryan's chief of staff, or Richard Juliano, Fawell's top aide, to cut the checks to the women. According to court records made public Wednesday, Ryan's daughter, Lynda Pignotti, now known as Lynda Fairman, was paid a combined 5,950 in 1995 by the Gramm campaign in four checks laundered through Drazek's business, American Management Resources.
<p>
<p>
In 1996, individual checks went to Ryan daughters Nancy Coghlan, who received 1,725, and Joanne Barrow and Julie R. Koehl, who each pocketed 1,000, the records showed.
<p>
<p>
A source said all four daughters had been given immunity from prosecution by federal authorities and testified before the grand jury investigating Fawell as part of the Operation Safe Road probe.
<p>
<p> Full story at Chicago Tribune <p>
<p>

<p> </BODY>
</DOCUMENT>

Example - Output

Full story at <LOCATION>Baltimore Sun</LOCATION>

By <PERSON>Matt O ' Connor</PERSON> , <ORGANIZATION>Tribune</ORGANIZATION> staff reporter . Tribune staff reporter <PERSON>Ray Gibson</PERSON> contributed to this report

Four of former Gov . <PERSON>George Ryan</PERSON> ' s daughters shared in almost <MONEY>10 , 000</MONEY> in secret payments from Sen . <PERSON>Phil Gramm</PERSON> ' s presidential campaign in the <DATE>mid - 1990 s</DATE> , according to testimony <DATE>Wednesday</DATE> in federal court .

<PERSON>Alan Drazek</PERSON> , who said he was brought in at <ORGANIZATION>Ryan</ORGANIZATION> ' s request as a point man for the Gramm campaign in <LOCATION>Illinois</LOCATION> , testified he was told by <PERSON>Scott Fawell</PERSON> , Ryan ' s chief of staff , or <PERSON>Richard Juliano</PERSON> , <PERSON>Fawell</PERSON> ' s top aide , to cut the checks to the women . According to court records made public <DATE>Wednesday</DATE> , Ryan ' s daughter , <PERSON>Lynda Pignotti</PERSON> , now known as <PERSON>Lynda Fairman</PERSON> , was paid a combined <PERCENT>5</PERCENT> , 950 in <DATE>1995</DATE> by the Gramm campaign in four checks laundered through Drazek ' s business , <ORGANIZATION>American Management Resources</ORGANIZATION> .

In <DATE>1996</DATE> , individual checks went to Ryan daughters <PERSON>Nancy Coghlan</PERSON> , who received <MONEY>1 , 725</MONEY> , and <PERSON>Joanne Barrow and Julie R . Koehl</PERSON> , who each <MONEY>pocketed 1 , 000</MONEY> , the records showed .

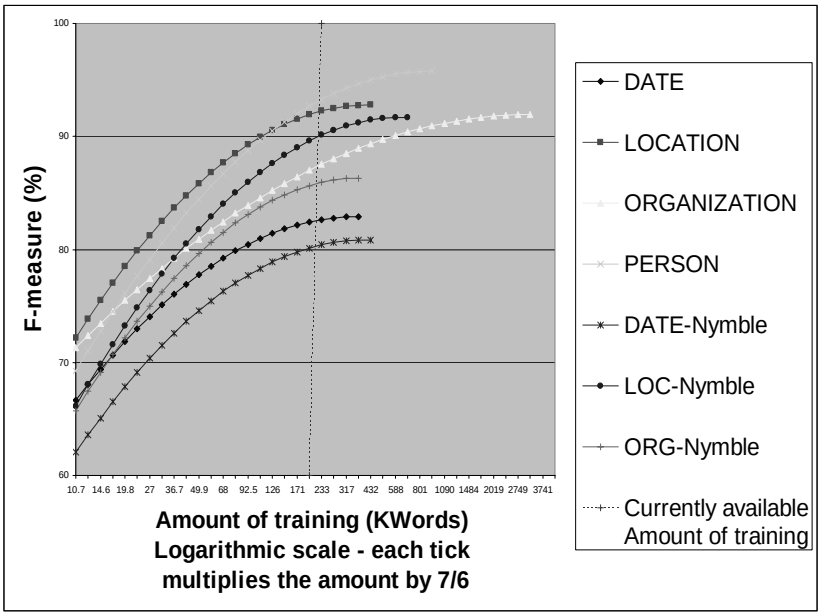
A source said all four daughters had been given immunity from prosecution by federal authorities and testified before the grand jury investigating Fawell as part of the Operation Safe Road probe .

Full story at <ORGANIZATION>Chicago Tribune</ORGANIZATION>

Training: ACE + MUC => MUC

r/p	muc7	ace+muc7
person	91.9/85.5	84.9/88.6
organization	91.1/93.7	83.1/95.9
date	90.9/76.6	59/89.5
time	76.4/77.6	68.6/92.5
location	90.7/91.3	77.7/91.7
money	97.6/82.1	86.6/82.1
percent	93.7/40.54	50/29.6

Results with our new algorithm



Some Useful Resources

- **Linguistic Data Consortium (LDC)**
  - **Lexicons**
  - **Annotated Corpora (Text and Speech)**
- **New Mexico State University**
  - **Gazetteers**
  - **Many lists of names**
  - **Lexicons for different languages**
- **Various Web Sources**
  - **CIA World Fact Book**
  - **Hoovers**
  - **SEC, Nasdaq (list of public company names)**
  - **US Census data**
  - **Private web sites (like Arabic, Persian, Pakistani names)**

Shallow Parsing in IE

- **Only Core Constituents are extracted**
- **No attempt is made at full parses**
- **Relevant prepositional attachments are extracted.**
  - ***I saw the man with a telescope***
- ***Only adverbials related to location and time are processed, others are ignored.***
- ***Quantifiers, modals, and propositional attitudes are ignored, or treated in a simplified way.***

Why not Full Parsing?

- **Full Parsing for IE was tried in:**
  - **SRI Tacitus system (MUC 3)**
  - **NYU Proteus (Muc-6)**
- **Main Issues:**
  - **Slow (combinatorial explosion of possible parses)**
  - **Erroneous**
  - **A simple predicate-argument structure is needed.**

Coreference

- **The general problem is related to co referential relations between expressions**
  - **Whole – part relationship**
  - **Containment relationship (set/subset)**
- **A simplest version is to find which noun phrases refer to the same entity**
- **An even more restricted version is to limit it just to proper names.**
- **Example:**
  - **The President, George Bush, George W. Bush, or even "W", all refer to the same entity.**

Easy and hard in Coreference

- **"Mohamed Atta, a suspected leader of the hijackers, had spent time in Belle Glade, Fla., where a crop-dusting business is located.** Atta **and other Middle Eastern men came to South Florida Crop Care nearly every weekend for two months.**
- **"Will Lee, the firm's general manager, said the** men **asked repeated questions about the crop-dusting business.** He **said the questions seemed "odd," but** he **didn't find the** men **suspicious until after the Sept. 11 attack."**

The “Simple” Coreference

- **Proper Names**
  - **IBM, “International Business Machines”, Big Blue**
  - **Osama Bin Ladin, Bin Ladin, Usama Bin Laden. (note the variations)**
- **Definite Noun Phrases**
  - **The Giant Computer Manufacturer, The Company, The owner of over 600,000 patents**
- **Pronouns**
  - **It, he , she, we…..**

Coreference Example

Granite Systems provides Service Resource Management (SRM) software for communication service providers with wireless, wireline, optical and packet technology networks. Utilizing Granite' Xng System, carriers can manage inventories of network resources and capacity, define network configurations, order and schedule resources and provide a database of record to other operational support systems (OSSs). An award-winning company, including an Inc. 500 company in 2000 and 2001, Granite Systems enables clients including AT&T Wireless, KPN Belgium, COLT Telecom, ATG and Verizon to eliminate resource redundancy, improve network reliability and speed service deployment. Founded in 1993, the company is headquartered in Manchester, NH with offices in Denver, CO; Miami, FL; London, U.K.; Nice, France; Paris, France; Madrid, Spain; Rome, Italy; Copenhagen, Denmark; and Singapore.

A KE Approach to Corefernce

- **Mark each noun phrase with the following:**
  - **Type (company, person, location)**
  - **Singular vs. plural**
  - **Gender (male, female, neutral)**
  - **Syntactic (name, pronoun, definite/indefinite)**
- **For each candidate**
  - **Find accessible antecedents**
    - **Each antecedent has a different scope**
      - **Proper names's scope is the whole document**
      - **Definite clauses's scope is the preceding paragraphs**
      - **Pronouns might be just the previous sentence, or the same paragraph.**
  - **Filter by consistency check**
  - **Order by dynamic syntactic preference**

Filtering Antecedents

- **George Bush will not match "she", or "it"**
- **George Bush can not be an antecedent of "The company" or "they"**
- **Using a sort Hierarchy we can use background information to be smarter**
  - **Example: "The big automaker is planning to get out the car business. The company feels that it can never longer make a profit making cars."**

IE Via BootStrapping

AutoSlog (Riloff, 1993)

- **Creates Extraction Patterns from annotated texts (NPs).**
- **Uses Sentence analyzer (CIRCUS, Lehnert, 1991) to identify clause boundaries and syntactic constituents (subject, verb, direct object, prepositional phrase)**
- **It then uses heuristic templates to generate extraction patterns**

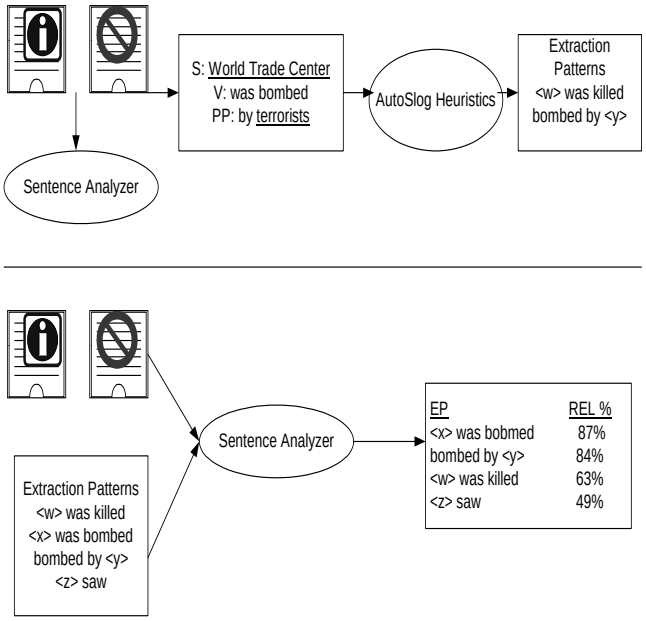
Example Templates

Template	Example
<subj> passive-verb	<victim> was murdered
<subj> aux noun	<victim> was victim
Active-verb <dobj>	Bombed <target>
Noun prep <np>	Bomb against <target>
Active-verb prep <np>	Killed with <instrument>
Passive-verb prep <np>	Was aimed at <target>

AutoSlog-TS (Riloff, 1996)

- It took 8 hours to annotate 160 documents, and hence probably a week to annotate 1000 documents.
- This bottleneck is a major problem for using IE in new domains.
- Hence there is a need for a system that can generate IE patterns from un-annotated documents.
- AutoSlog-TS is such a system.

AutoSlog TS



Top 24 Extraction Patterns

<subj> exploded	**Murder of <np>**	**Assassination of <np>**
<subj> was killed	**<subj> was kidnapped**	**Attack on <np>**
<subj> was injured	**Exploded in <np>**	**Death of <np>**
<subj> took place	**Caused <dobj>**	**Claimed <dobj>**
<subj> was wounded	**<subj> occured**	**<subj> was loctated**
Took place on <np>	**Responsibility for <np>**	**Occurred on <np>**
Was wounded in <np>	**Destroyed <dobj>**	**<subj> was murdered**
One of <np>	**<subj> kidnapped**	**Exploded on <np>**

Evaluation

- **Data Set: 1500 docs from MUC-4 (772 relevant)**
- **AutoSlog generated 1237 patterns which were manually filtered to 450 in 5 hours.**
- **AutoSlog-TS generated 32,345 patterns, after discarding singleton patterns, 11,225 were left.**
- **Rank(EP) =** $\frac{rel - freq}{freq} \log_2 freq$
- **The user reviewed the the top 1970 patterns and selected 210 of them in 85 minutes.**
- **AutoSlog achieved better recall, while AutoSlog-TS achieved better precision.**

Learning Dictionaries by Bootstrapping (Riloff and Jones, 1999)

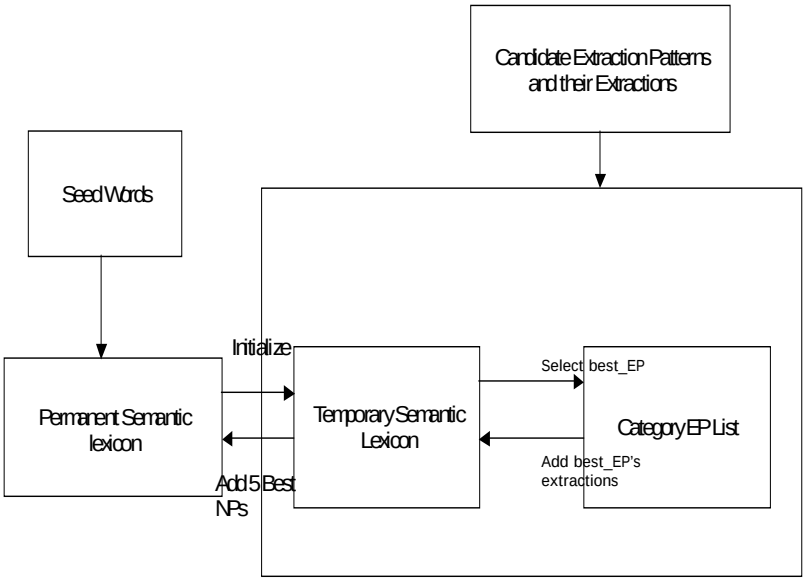
- **Learn dictionary entries (semantic lexicon) and extraction patterns simultaneously.**
- **Use untagged text as a training source for learning.**
- **Start with a set of seed lexicon entries and using mutual bootstrapping learn extraction patterns and more lexicon entries.**

Mutual Bootstrapping Algorithm

- **Using AutoSlog generate all possible extraction patterns.**
- **Apply patterns to the corpus and save results to EPData**
- **SemLex = Seed Words**
- **Cat_EPList = {}**

1. **Score all extraction patterns in EPData**
2. **Best_EP = highest scoring pattern**
3. **Add Best_EP to Cat_EPList**
4. **Add Best_EP's extractions to SemLex**
5. **Goto 1**

Meta Bootstrapping Process



Sample Extraction Patterns

www location	www company	terrorism location
offices in <x>	owned by <x>	living in <x>
facilities in <x>	<x> employed	traveled to <x>
operations in <x>	<x> is distributor	become in <x>
operates in <x>	<x> positioning	sought in <x>
seminars in <x>	motivated <x>	presidents of <x>
activities in <x>	sold to <x>	parts of <x>
consulting in <x>	devoted to <x>	to enter <x>
outlets in <x>	<x> thrive	ministers of <x>
customers in <x>	message to <x>	part in <x>
distributors in <x>	<x> request information	taken in <x>
services in <x>	<x> has positions	returned to <x>
expanded into <x>	offices of <x>	process in <x>

Experimental Evaluation

	Iter 1	Iter 10	Iter 20	Iter 30	Iter 40	Iter 50
Web Company	5/5 (1)	25/32 (.78)	52/65 (.80)	72/113 (.64)	86/163 (.53)	95/206 (.46)
Web Location	5/5 (1)	46/50 (.92)	88/100 (.88)	129/150 (.86)	163/200 (.82)	191/250 (.76)
Web Title	0/1 (0)	22/31 (.71)	63/81 (.78)	86/131 (.66)	101/181 (.56)	107/231 (.46)
Terr. Location	5/5 (1)	32/50 (.64)	66/100 (.66)	100/150 (.67)	127/200 (.64)	158/250 (.63)
Terr. Weapon	4/4 (1)	31/44 (.70)	68/94 (.72)	85/144 (.59)	101/194 (.52)	124/244 (.51)

KDD Cup Task 1

Information Extraction from Biomedical Articles

System Description

June / July 2002

The Task: Curate or Not-Curate

Build a system for automatic analysis of scientific papers regarding the **Drosophila Fruit Fly**.

The system should extract (curate) only the papers that include **experimental results** regarding **expression** of **gene products**, and **identify** these genes and products

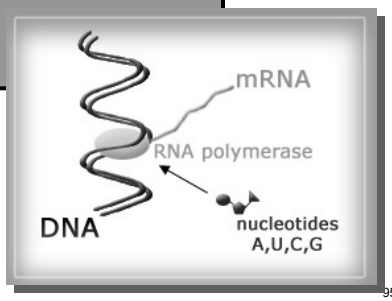
A **Product** – **mRNA or Protein** actually identified (**naturally**) within specific cells of the natural (**Wild-Type**) fly.

For each paper, a **list of all genes** mentioned in the paper - for which we must decide if there is a product result - is given

Quick Biological Background

Transcription

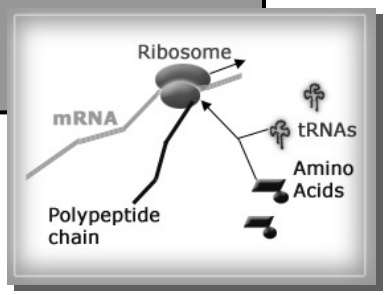
RNA **(Ribonucleic Acid) is a molecule that is "mathematically"** equivalent to **(but chemically different from) the** DNA **sequence of the gene.** Transcription **means transfer of the genetic information from the archival copy of** DNA **to the short-lived** messenger RNA **(**mRNA**)**



Quick Biological Background (Continued)

Translation

is the process that takes a sequence in one code – nucleotides, and creates the corresponding sequence in another code - amino acids (The building blocks of peptides / proteins). A protein will be expressed only if its code was "translated" from the mRNA.



The Task: So what's the problem?

- **Very often papers discuss** mutations **and forced (**ectopic**) expression of genes in addition to** natural **ones**
- **Many genes are "just mentioned" within the papers without actually citing results or are being used as an auxiliary tool for investigating other genes**

  (Example: The White/Red Eye **Gene -** *w***)**
- **The** Transcript **vs.** Protein **distinction is tricky (they usually have the same name …)**

Our System: Translating the problem into an Information Extraction Task

- **The scientific papers given are lengthy and complex …**
- **We're given only a** text version **without images**
- **But they have a very** fixed structure
- **We're actually interested only in specific, actual** experimental results
- **Fortunately, these results are obtained using a set of** well-known techniques
- **Our approach is** Knowledge-Based Information Extraction**, i.e. finding frequent patterns relevant to the** domain

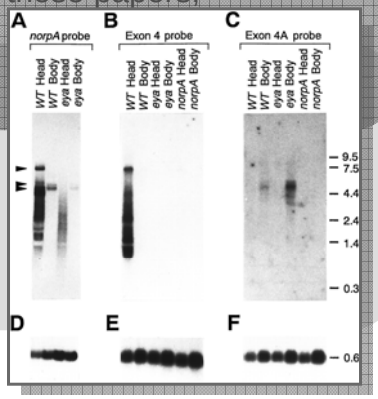
So our Solution is …

The Figure IS the Result

Molecular Biologists who review these papers, look mainly for the figures!

Example:

This figure (from *R100, in the Training Set) that shows that a specific transcript is present both in the eye and the body.



Obvious highlighted sections (Title and Abstract) are used too.

*Multiple Subtypes of Phospholipase C Are Encoded by the norpA Gene of Drosophila melanogaster
Sunkyu Kim , Richard R. McKay , Karen Miller , Randall D. Shortridge
J. Biol. Chem. 270(24): 14376-82.

The Figure IS the Result
(Continued)

But our system **can't read** figures and actually **doesn't have** them …

The Solution …

The Alternative: Focus on Figure Legend

This is how the extract from the same paper looks as a text file

```
@Northern Analysis of Adult RNA@

   When radiolabeled @norpA@ cDNA probes are hybridized to blots of
   poly(A) [_2747_tex2html_wrap740.xbm] RNA, three major transcripts can
   be identified. As shown in Fig. 3(@panel@@A@), a major @norpA@
   transcript that is 7.5 kb in length is easily detected in wild-type
   head but is absent from head of @eya@ mutant. The absence of the
   7.5-kb transcript from @eya@ head suggests that it is expressed in
     the
   compound eye. Two other transcripts, one that is 5.5 kb and one that
   is 5.0 kb in length, are visible in body. None of these transcripts
   are detectable in head or body of @norpA@
     [_2747_tex2html_wrap732.xbm]
   mutant (Zhu @et al.@, 1993), suggesting that they are encoded by the
   @norpA@ gene.

   [bc2558926003.gif]

   ---------------------------------------------------------------------

   Figure 3: Northern blot analysis of @norpA@ transcripts
     in adult @Drosophila@ tissues. Approximately 5 µg of poly(A)
   [_2747_tex2html_wrap740.xbm] RNA was loaded in each lane and probed
   with a 3.4-kb @norpA@ cDNA fragment (nucleotides 1-3453) (@A@), an
   80-bp exon 4 cDNA fragment (@B@), or an 80-bp exon 4A cDNA fragment
   (@C@). @Lane@ designations indicate RNA isolated from adult head or
   body (thorax and abdomen) of wild-type (@WT@) @Drosophila@, eyes
   absent (@eya@) mutant, or @norpA@ mutant. Mobility of RNA size
   standards (in kilobases) are indicated on the @right@. @Panels@@D@-
     @F@
   show the result of reprobing the blots with @Drosophila@ RP49 cDNA
   (O'Connell and Rosbash, 1984) as a control to test for RNA loading.
```

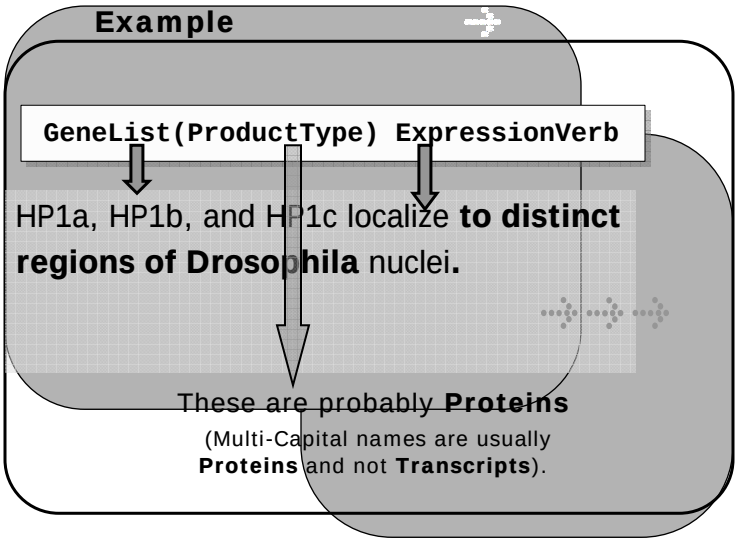

Extracting the Pattern from the Figure Legend

- **Extracting (finding) the Figure Title is easy :**

  "Figure #" **or** "Fig. #" **beginning at a new line**

- **Look for** patterns **incorporating a technique used in obtaining the results (for example,** Northern blot**), or noun phrase or verb describing an expression (“**expression**”, “**localization**”, “**expressed**” …) with a synonym of** Gene(s).

Extracting the Pattern from the Figure Legend



Making the Curate Decision : Extract Evidences and Score Them

- **Extract evidences from Title , Abstract , Figure Legend and GenBank footnotes**
- **Keep a** Score **entry for the whole document and for each product (transcript/protein) of a candidate gene**
- **At the end of the document, use the scores to decide regarding the curation of the document and the products of the candidate genes.**

(If a gene's score is above a certain threshold, mark the gene as having an experimental result, and mark the whole document as curatable).

Making the Curate Decision : Positive and Negative Evidences

Positive Evidence

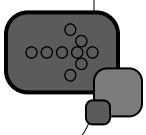
"Northern blot analysis of @norpA@ transcripts **in adult @Drosophila@ tissues**"

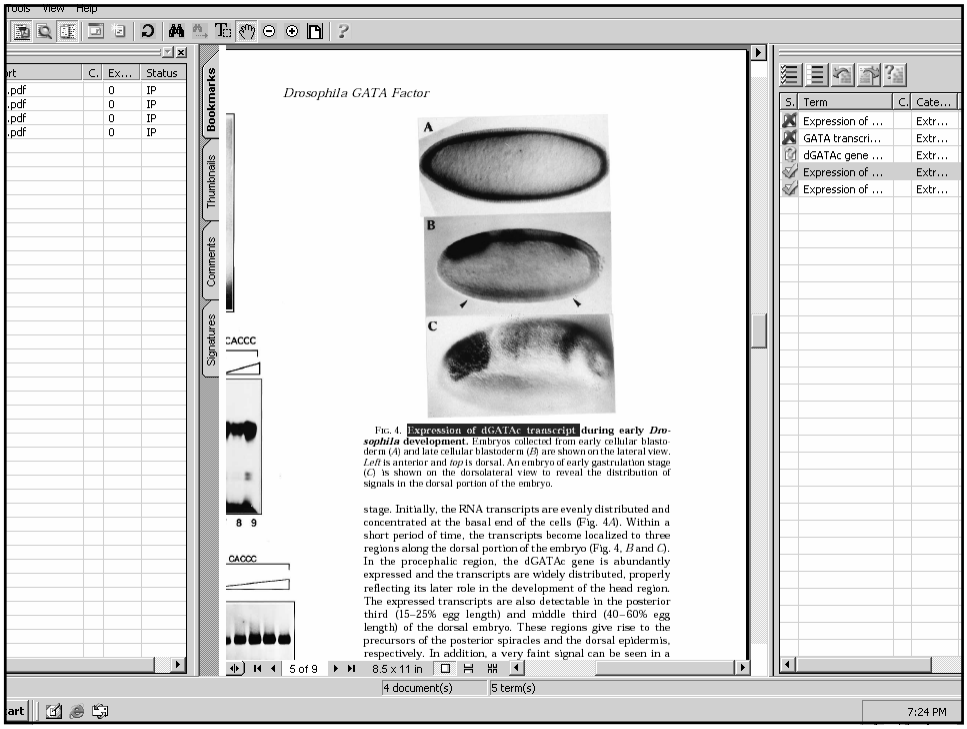
Negative Evidence

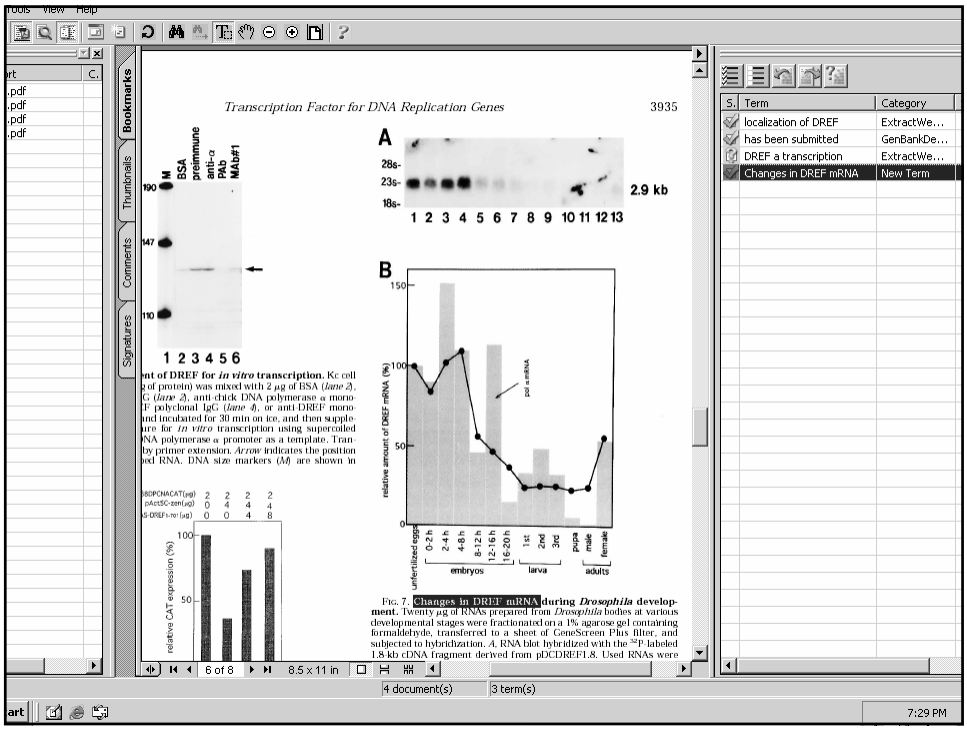
"Figure 2. **Ectopic expression of @dNSF1@** in the nervous system **rescues** the phenotypes of **@dNSF1@ mutations**"

Implementation : DIAL Rulebook

- **The System is implemented in** DIAL **(Declarative Information Analysis Language), a general IE language developed at** ClearForest
- **DIAL is based on** matching patterns **within the text and then checking constraints on the patterns.**
- **Patterns combine** syntactic **and** semantic **elements.**




Drosophila GATA Factor
A
B
C
CACCC
8 9
CACCC
FIG. 4. Expression of dGATAc transcript during early Drosophila development. Embryos collected from early cellular blastoderm (A) and late cellular blastoderm (B) are shown on the lateral view. Left is anterior and top is dorsal. An embryo of early gastrulation stage (C) is shown on the dorsolateral view to reveal the distribution of signals in the dorsal portion of the embryo.
stage. Initially, the RNA transcripts are evenly distributed and concentrated at the basal end of the cells (Fig. 4A). Within a short period of time, the transcripts become localized to three regions along the dorsal portion of the embryo (Fig. 4, B and C). In the procephalic region, the dGATAc gene is abundantly expressed and the transcripts are widely distributed, properly reflecting its later role in the development of the head region. The expressed transcripts are also detectable in the posterior third (15–25% egg length) and middle third (40–60% egg length) of the dorsal embryo. These regions give rise to the precursors of the posterior spiracles and the dorsal epidermis, respectively. In addition, a very faint signal can be seen in a
5 of 9
8.5 x 11 in
4 document(s)
5 term(s)
7:24 PM


Transcription Factor for DNA Replication Genes 3935
M
BSA
preimmune
anti-α PAb
MAb#1
190
147
110
1 2 3 4 5 6
nt of DREF for in vitro transcription. Kc cell
2 of protein) was mixed with 2 μg of BSA (lane 2),
G (lane 3), anti-chick DNA polymerase α mono-
F polyclonal IgG (lane 4), or anti-DREF mono-
nd incubated for 30 min on ice, and then supple-
re for in vitro transcription using supercoiled
NA polymerase α promoter as a template. Tran-
by primer extension. Arrow indicates the position
ed RNA. DNA size markers (M) are shown in
A
28s-
23s-
18s-
2.9 kb
1 2 3 4 5 6 7 8 9 10 11 12 13
B
150
100
50
0
relative amount of DREF mRNA (%)
pol α mRNA
unfertilized egg
0-2 h
2-4 h
4-8 h
8-12 h
12-16 h
16-20 h
1st
2nd
3rd
pupa
male
female
embryos
larva
adults
relative CAT expression (%)
100
50
FIG. 7. Changes in DREF mRNA during Drosophila development. Twenty μg of RNAs prepared from Drosophila bodies at various developmental stages were fractionated on a 1% agarose gel containing formaldehyde, transferred to a sheet of GeneScreen Plus filter, and subjected to hybridization. A, RNA blot hybridized with the 32P-labeled 1.8-kb cDNA fragment derived from pDCDREF1.8. Used RNAs were
6 of 8
8.5 x 11 in
4 document(s)
3 term(s)
7:29 PM

Results and Evaluation

Results achieved

- **Document Curation : 78% F-Measure**
- **Gene Products : 67% F-Measure**

Results and Evaluation
(Continued)

Evaluation

Information Extraction is more suitable than **Categorization** for this task.
(Best Categorization Curation Results – about 62-64% F-Measure)

- Most papers belong to a narrow **domain** (same **vocabulary**).
- Many curatable papers have both **relevant** results (wild-type expression) and **irrelevant** ones (Mutations etc.)
- Extracting evidences of **specific products of genes** cannot be achieved by categorization. **Patterns** with the **specific genes** must be found.
  (No real generalization can be made regarding specific genes, other than ***w***)

A Hybrid Approach

Merging the Rule Base and ML Approaches

Why is HMM not enough?

- **The HMM model is flat, so the most it can do is assign a tag to each token in a sentence.**
- **This is suitable for the tasks where the tagged sequences do not nest and where there are no explicit relations between the sequences.**
- **Part-of-speech tagging and entity extraction belong to this category.**
- **Extracting relationships is different, because the tagged sequences can (and must) nest, and there are relations between them which must be explicitly recognized.**
- **< ACQUISITION> <ACQUIRER>** ***Ethicon Endo-Surgery, Inc.*** **</ACQUIRER>** ***, a Johnson & Johnson company, has acquired*** **< ACQUIRED>** ***Obtech Medical AG*** **< /ACQUIRED> </ACQUISITION>**

A Hybrid Approach

- **The hybrid strategy, attempts to strike a balance between the two knowledge engineer chores**
  - **writing the extraction rules**
  - **manually tagging the documents.**
- **In this strategy, the knowledge engineer writes SCFG rules, which are then trained on the data which is available.**
- **The powerful disambiguating ability of the SCFG makes writing rules much simpler and cleaner task.**
- **The knowledge engineer has the control of the generality of the rules (s)he writes, and consequently on the amount and the quality of the manually tagged training the system would require.**

Defining SCFG

- Classical definition**: A stochastic context-free grammar (SCFG) is a quintuple $G = (T, N, S, R, P)$**
  - T **is the alphabet of terminal symbols (tokens)**
  - N **is the set of nonterminals**
  - S **is the starting nonterminal**
  - R **is the set of rules**
  - $P : R \rightarrow [0..1]$ **defines their probabilities.**
- **The rules have the form $n \rightarrow s_1 s_2 \ldots s_k$, where n is a nonterminal and each s_i either token or another nonterminal.**
- **SCFG is a usual context-free grammar with the addition of the P function.**

The Probability Function

- **If *r* is the rule $n \rightarrow s_1 s_2 \ldots s_k$, then *P*(*r*) is the frequency of expanding *n* using this rule.**
- **In Bayesian terms, if it is known that a given sequence of tokens was generated by expanding *n*, then *P*(*r*) is the apriory likelihood that *n* was expanded using the rule *r*.**
- **Thus, it follows that for every nonterminal *n* the sum $\sum P(r)$ over all rules *r* headed by *n* must equal to one.**

IE with SCFG

- **A very basic "parsing" is employed for the bulk of a text, but within the relevant parts, the grammar is much more detailed.**
- **The IE grammars can be said to define *sublanguages* for very specific domains.**

IE with SCFG

- **In the classical definition of SCFG it is assumed that the rules are all independent. In this case it is possible to find the (unconditional) probability of a given parse tree by simply multiplying the probabilities of all rules participating in it.**
- **The usual parsing problem is given a sequence of tokens (a *string*) *S*, to find the most probable parse tree *T* which could generate *S*. A simple generalization of the Viterbi algorithm is able to efficiently solve this problem.**
- **In practical applications of SCFGs, it is rarely the case that the rules are truly independent. Then, the easiest way to cope with this problem while leaving most of the formalism intact is to let the probabilities *P*(*r*) be conditioned upon the context where the rule is applied.**

Markovian CSFG

- **HMM entity extractors, are a simple case of markovian SCFGs.**
- **Every possible rule which can be formed from the available symbols has nonzero probability.**
- **Usually, all probabilities are initially set to be equal, and then adjusted according to the distributions found in the training data.**

Training Issues

- **For some problems the available training corpora appear to be adequate.**
- **In particular, markovian SCFG parsers trained on the Penn Treebank perform quite well (Collins 1997, Charniak 2000, Roark 2001, etc).**
- **But for the task of relationship extraction it turns out to be impractical to manually tag the amount of documents that would be sufficient to adequately train a markovian SCFG.**
- **At a certain point it becomes more productive to go back to the original hand-crafted system and write rules for it, even though it is a much more skilled labor!**

SCFG Syntax

- **A rulebook consists of declarations and rules. All nonterminals must be declared before usage.**
- **Some of them can be declared as *output concepts*, which are the entities, events, and facts that the system is designed to extract. Additionally, two classes of terminal symbols also require declaration: *termlists*, and *ngrams*.**
  - **A termlist is a collection of terms from a single semantic category, either written explicitly or loaded from external source.**
  - **An ngram can expand to any single token. But the probability of generating a given token is not fixed in the rules, but learned from the training dataset, and may be conditioned upon one or more previous tokens. Thus, ngrams is one of the ways the probabilities of the SCFG rules can be context-dependent.**

Example

```
output concept Acquisition(Acquirer, Acquired);
ngram AdjunctWord;

nonterminal Adjunct;
Adjunct  :-  AdjunctWord  Adjunct  |  AdjunctWord;

termlist AcquireTerm = acquired bought (has acquired) (has
    bought);

Acquisition  :-  Company→Acquirer [ “,”Adjunct “,” ]
AcquireTerm  Company→Acquired;
```


Emulation of HMM Entity Extractor in CSFG

```
output concept Company();
ngram CompanyFirstWord;
ngram CompanyWord;
ngram CompanyLastWord;
nonterminal CompanyNext;

Company :- CompanyFirstWord CompanyNext
   | CompanyFirstWord;

CompanyNext :- CompanyWord CompanyNext
   | CompanyLastWord;
```

Putting is all together

start Text;

nonterminal None;

ngram NoneWord;

None :- NoneWord None | ;

Text :- None Text | Company Text |
Acquisition Text | ;

SCFG Training

- **Currently there are three different classes of trainable parameters in a TEG rulebook:**
  - **the probabilities of rules of nonterminals**
  - **the probabilities of different expansions of ngrams**
  - **the probabilities of terms in a wordclass.**
- **All those probabilities are smoothed maximum likelihood estimates, calculated directly from the frequencies of the corresponding elements in the training dataset.**

Sample Rules

```
concept Text;
concept output Person;
ngram NGFirstName;
ngram NGLastName;
ngram NGNone;
wordclass WCHonorific = Mr Mrs Miss Ms Dr;
Person :- WCHonorific NGLastName;
Person :- NGFirstName NGLastName;
Text :- NGNone Text;
Text :- Person Text;
Text :- ;
```


Training the rule book

- **By default, the initial untrained frequencies of all elements are assumed to be 1. They can be changed using "<count>" syntax.**
- **Let us train this rulebook on the training set containing one sentence:**
  - ***Yesterday*, <Person>*Dr Simmons*</Person>, *the distinguished scientist*, *presented the discovery*.**
- **This is done in two steps. First, the sentence is parsed using the untrained rulebook, but with the constraints specified by the annotations. In our case the constraints are satisfied by two different parses:**

2 Possible Parse Trees

Text
- **NGNone**
  - *Yesterday*
- **Text**
  - **NGNone**
    - ,
  - **Text**
    - **Person**
      - **WCHonorific**
        - *Dr*
      - **NGLastName**
        - *Simmons*
    - **Text**
      - **NGNone**
        - ,
      - **Text**
        - ……

Text
- **NGNone**
  - *Yesterday*
- **Text**
  - **NGNone**
    - ,
  - **Text**
    - **Person**
      - **NGFirstName**
        - *Dr*
      - **NGLastName**
        - *Simmons*
    - **Text**
      - **NGNone**
        - ,
      - **Text**
        - ……

How to pick the right Parse?

- **The difference is in the expansion of the** Person **nonterminal. Both** Person **rules can produce the output instance, therefore there is an ambiguity.**
- **In this case it is resolved in favor of the** WCHonorific **interpretation, because in the untrained rulebook we have**
  - ***P*(*Dr* |** WCHonorific**) = 1/5 (choice of one term among five equiprobable ones),**
  - ***P*(*Dr* |** NGFirstName**) ≈ 1/*N*, where *N* is the number of all known words.**

The Trained Rule Book

```
concept Text;
concept output Person;
ngram NGFirstName;
ngram NGLastName;
ngram NGNone;
wordclass WCHonorific = Mr Mrs Miss Ms <2>Dr;
Person :- <2>WCHonorific NGLastName;
Person :- NGFirstName NGLastName;
Text :- <11>NGNone Text;
Text :- <2>Person Text;
Text :- <2>;
```


A real example

- **The PersonAffiliation relation contains three attributes – name of the person, name of the organization, and position of the person in the organization. It is declared as follows:**
  - concept output PersonAffiliation(Name, Position, Org);
- **Most often, this relation is encountered in the text in the form**
  - **"Mr.Name, Position of Org" or**
  - **"Org Position Ms.Name".**
  - **Almost any order of the components is possible, with commas and prepositions inserted as necessary.**
  - **Also, it is common for Name, Position, or both to be conjunctions of pairs of corresponding entities:**
  - **"Mr.Name1 and Ms.Name2, the Position1 and Position2 of Org",**
  - **"Org's Position1 and Position2, Ms.Name".**
- **In order to catch those complexities, and for general simplification of the rules, we use several auxiliary non-terminals:**
  - Names**, which catches one or two Names,**
  - Positions**, which catches one or two Positions, and**
  - Orgs**, which catches Organizations and Locations, which can also be involved in PersonAffiliation, as in "Bush, president of US":**

The Basic Rules

- nonterms Names, Positions, Orgs;
  - Names :- PERSON->Name | PERSON->Name "and" PERSON->Name;
  - Positions :- POSITION->Position | POSITION->Position "and" POSITION->Position;
  - Orgs :- ORGANIZATION->Org | LOCATION->Org;
- **We also use auxiliary non-terminals for catching pairs of attributes:** PosName**, and** PosOrg**:**
  - nonterms PosName, PosOrg;
  - PosName :- Positions Names | PosName "and" PosName;
  - wordclass wcPreposition = "at" "in" "of" "for" "with";
  - wordclass wcPossessive = ("'" " "s") "' ";
  - PosOrg :- Positions wcPreposition Orgs;
  - PosOrg :- Orgs [wcPossessive] Positions;
- **Finally, the PersonAffiliation rules:**
  - PersonAffiliation :- Orgs [wcPossessive] PosName;
  - PersonAffiliation :- PosName wcPreposition Orgs;
  - PersonAffiliation :- PosOrg [","] Names;
  - PersonAffiliation :- Names "," PosOrg;
  - PersonAffiliation :- Names "is" "a" PosOrg;

What is Missing?

- **The rules above catch about 50% of all PersonAffiliation instances in the texts.**
- **Other instances do not conform to the patterns above in several respects. So, in order to improve the accuracy, additional rules need to be written.**
- **First, the Organization name is often entered into a sentence as a part of a descriptive noun phrase, as in:**
  - **"Ms.Name is a Position of the industry leader Org".**
  - **In order to catch this in a general way, we define an** OrgNP **nonterm, which uses an external PoS tagger:**

Advanced Rules

- Using External POS Tagger
  - ngram ngOrgNoun featureset ExtPoS restriction Noun;
  - ngram ngOrgAdj featureset ExtPoS restriction Adj;
  - ngram ngNum featureset ExtPoS restriction Number;
  - ngram ngProper featureset ExtPoS restriction ProperName;
  - ngram ngDet featureset ExtPoS restriction Det;
  - ngram ngPrep featureset ExtPoS restriction Prep;
- nonterm OrgNounList;
  - OrgNounList :- ngOrgNoun [OrgNounList];
- nonterms OrgAdjWord, OrgAdjList;
  - OrgAdjWord :- ngOrgAdj | ngNum | ngProper;
  - OrgAdjList :- OrgAdjWord [OrgAdjList];
- nonterm OrgNP;
  - OrgNP :- [ngDet] [OrgAdjList] OrgNounList;
  - OrgNP :- OrgNP ngPrep OrgNP;
  - OrgNP :- OrgNP "and" OrgNP;

Experimental Evaluation

The INC Corpus

	Partial match results			Exact match results		
	Recall	**Prec**	**F**	**Recall**	**Prec**	**F**
PersonAffiliation	**89.61**	**94.52**	92.00	**75.33**	**79.46**	77.33
OrgLocation	**82.35**	**77.78**	80.00	**76.47**	**72.22**	74.29
Acquisition	**76.00**	**86.36**	80.85	**68.00**	**77.27**	72.34

MUC 7

	HMM entity extractor			***Emulation using SCFG***			***Full SCFG system***		
	Recall	***Prec***	***F***	***Recall***	***Prec***	***F***	***Recall***	***Prec***	***F***
Person	**86.91**	**85.13**	**86.01**	**86.31**	**86.83**	**86.57**	**93.75**	**90.78**	**92.24**
Organization	**87.94**	**89.75**	**88.84**	**85.94**	**89.53**	**87.7**	**89.49**	**90.9**	**90.19**
Location	**86.12**	**87.2**	**86.66**	**83.93**	**90.12**	**86.91**	**87.05**	**94.42**	**90.58**

ACE 2

	HMM entity extractor			Ergodic SCFG			Full SCFG system		
	Recall	**Prec**	**F**	**Recall**	**Prec**	**F**	**Recall**	**Prec**	**F**
Role				**50.99**	**76.24**	61.11	**83.44**	**77.3**	80.25
Person	**85.54**	**83.22**	84.37	**88.25**	**81.65**	84.82	**89.82**	**81.68**	85.56
Organization	**52.62**	**64.735**	58.05	**60.03**	**71.02**	65.06	**59.49**	**71.06**	64.76
GPE	**85.54**	**83.22**	84.37	**86.74**	**84.96**	85.84	**88.83**	**84.94**	86.84

Final Comparison

F1	Best TEG	DIAL	G TEG	HMM
ORG	**69.4895**	**68.5895**	**54.113**	**49.691**
PERSON	**84.835**	**83.5945**	**76.422**	**78.0975**
GPE	**88.957**	**89.392**	**85.256**	**83.8815**
LOC	**35.091**	**37.442**	**31.598**	**29.3645**
FAC	**38.0825**	**47.8705**	**24.074**	**22.9755**
{Sum}	**80.695**	**80.5605**	**72.633**	**71.4135**

Simple ROLE Rules (ACE-2)

- **ROLE :- [Position_Before] ORGANIZATION->ROLE_2 Position ["in" GPE] [","] PERSON->ROLE_1;**
- **ROLE :- GPE->ROLE_2 Position [","] PERSON->ROLE_1;**

- **ROLE :- PERSON->ROLE_1 "of" GPE->ROLE_2;**
- **ROLE :- ORGANIZATION->ROLE_2 "'" "s" [Position] PERSON->ROLE_1;**

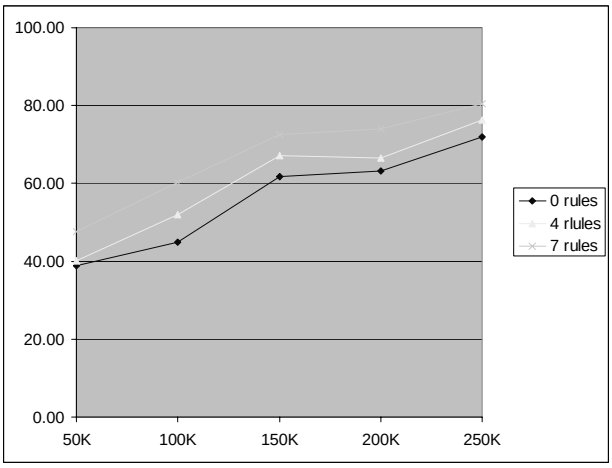
A little more complicated set of rules

- **ROLE :- [Position_Before] ORGANIZATION->ROLE_2 Position ["in" GPE] [","] PERSON->ROLE_1;**
- **ROLE :- GPE->ROLE_2 Position [","] PERSON->ROLE_1;**

- **ROLE :- PERSON->ROLE_1 "of" GPE->ROLE_2;**
- **ROLE :- ORGANIZATION->ROLE_2 "'" "s" [Position] PERSON->ROLE_1;**

- **ROLE :- GPE->ROLE_2 [Position] PERSON->ROLE_1;**
- **ROLE :- <5> GPE->ROLE_2 "'" "s" ORGANIZATION->ROLE_1;**
- **ROLE :- PERSON->ROLE_1 "," Position WCPreposition ORGANIZATION->ROLE_2;**

The Effect of the rules on the extraction accuracy



 141

Self-Supervised Relation Learning from the Web

Ronen Feldman
Data Mining Laboratory
Bar-Ilan University, ISRAEL

Joint work with Benjamin Rosenfeld

 142

Approaches for Building IE Systems

- **Knowledge Engineering Approach**
  - **Rules are crafted by linguists in cooperation with domain experts.**
  - **Most of the work is done by inspecting a set of relevant documents.**
  - **Can take a lot of time to fine tune the rule set.**
  - **Best results were achieved with KB based IE systems.**
  - **Skilled/gifted developers are needed.**
  - **A strong development environment is a MUST!**

Approaches for Building IE Systems

- **Automatically Trainable Systems**
  - **The techniques are based on pure statistics and almost no linguistic knowledge**
  - **They are language independent**
  - **The main input is an annotated corpus**
  - **Need a relatively small effort when building the rules, however creating the annotated corpus is extremely laborious.**
  - **Huge number of training examples is needed in order to achieve reasonable accuracy.**
  - **Hybrid approaches can utilize the user input in the development loop.**

KnowItAll (KIA)

- **KnowItAll is a system developed at University of Washington by Oren Etzioni and colleagues (Etzioni, Cafarella et al. 2005).**
- **KnowItAll is an autonomous, domain-independent system that extracts facts from the Web. The primary focus of the system is on extracting entities (unary predicates), although KnowItAll is able to extract relations (N-ary predicates) as well.**
- **The input to KnowItAll is a set of entity classes to be extracted, such as "city", "scientist", "movie", etc., and the output is a list of entities extracted from the Web.**

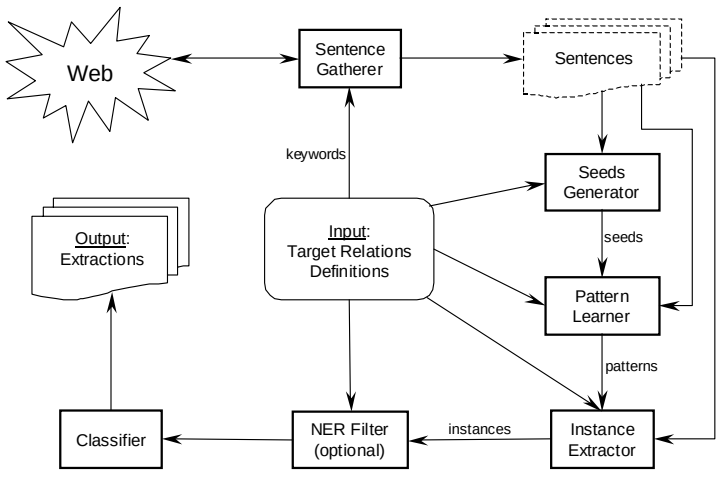
KnowItAll's Relation Learning

- **The base version of KnowItAll uses only the generic hand written patterns. The patterns are based on a general Noun Phrase (NP) tagger.**
- **For example, here are the two patterns used by KnowItAll for extracting instances of the** *Acquisition*(*Company*, *Company*) **relation:**
  - **NP2 "was acquired by" NP1**
  - **NP1 "'s acquisition of" NP2**
- **And the following are the three patterns used by KnowItAll for extracting the** *MayorOf*(*City*, *Person*) **relation:**
  - **NP ", mayor of" <city>**
  - **<city> "'s mayor" NP**
  - **<city> "mayor" NP**

SRES

- **SRES (**Self-Supervised Relation Extraction System**) which learns to extract relations from the web in an unsupervised way.**
- **The system takes as input the name of the relation and the types of its arguments and returns as output a set of instances of the relation extracted from the given corpus.**

SRES Architecture



Seeds for Acquisition

- **Oracle – PeopleSoft**
- **Oracle – Siebel Systems**
- **PeopleSoft – J.D. Edwards**
- **Novell – SuSE**
- **Sun – StorageTek**
- **Microsoft – Groove Networks**
- **AOL – Netscape**
- **Microsoft – Vicinity**
- **San Francisco-based Vector Capital – Corel**
- **HP – Compaq**

Major Steps in Pattern Learning

- **The sentences containing the arguments of the seed instances are extracted from the large set of sentences returned by the Sentence Gatherer.**
- **Then, the patterns are learned from the seed sentences.**
  - **We need to generate automatically**
    - **Positive Instances**
    - **Negative Instances**
- **Finally, the patterns are post-processed**

Positive Instances

- **The positive set of a predicate consists of sentences that contain an instance of the predicate, with the actual instance's attributes changed to "*<AttrN>*", where *N* is the attribute index.**
- **For example, the sentence**
  - ***"The Antitrust Division of the U.S. Department of Justice evaluated the likely competitive effects of Oracle's proposed acquisition of PeopleSoft."***
- **will be changed to**
  - ***"The Antitrust Division… …….effects of <Attr1>'s proposed acquisition of <Attr2>."***

Negative Instances II

- **We generate the negative set from the sentences in the positive set by changing the assignment of one or both attributes to other suitable entities in the sentence.**
- **In the shallow parser based mode of operation, any suitable noun phrase can be assigned to an attribute.**

Examples

- ***The Positive Instance***
  - ***"The Antitrust Division of the U.S. Department of Justice evaluated the likely competitive effects of <Attr1>'s proposed acquisition of <Attr2>"***
- ***Possible Negative Instances***
  - ***<Attr1> of the <Attr2> evaluated the likely…***
  - ***<Attr2> of the U.S. … …acquisition of <Attr1>***
  - ***<Attr1> of the U.S. … …acquisition of <Attr2>***
  - ***The Antitrust Division of the <Attr1> ….. acquisition of <Attr2>"***

Additional Instances

- **we use the sentences produced by exchanging "*<Attr1>*" and "*<Attr2>*" (with obvious generalization for n-ary predicates) in the positive sentences.**
- **If the target predicate is *symmetric*, like *Merger*, then such sentences are put into the positive set.**
- **Otherwise, for anti-symmetric predicates, the sentences are put into the negative set.**

Pattern Generation

- **The patterns for a predicate *P* are generalizations of pairs of sentences from the positive set of *P*.**
- **The function *Generalize*(*S*1, *S*2) is applied to each pair of sentences *S*1 and *S*2 from the positive set of the predicate. The function generates a pattern that is the best (according to the objective function defined below) generalization of its two arguments.**
- **The following pseudo code shows the process of generating the patterns:**

For each predicate *P*
For each pair *S*1, *S*2 from *PositiveSet*(*P*)
Let *Pattern* = *Generalize*(*S*1, *S*2).
Add *Pattern* to *PatternsSet*(*P*).

The Pattern Language

- **The patterns are sequences of *tokens*, *skips*, and *slots*. The tokens can match only themselves, the skips match zero or more arbitrary tokens, and slots match instance attributes.**
- **Examples of patterns:**
  - ***<Attr1> * was acquired by <Attr2>***
  - ***<Attr1> * merged with * <Attr2>***
  - ***<Attr2> is * ceo of * <Attr1>***
- **Note, that the sentences from the positive and negative sets of predicates are also patterns, the least general ones since they do not contain skips.**

The Generalize Function

- **The *Generalize*(*s*1, *s*2) function takes two patterns (e.g., two sentences with slots marked as <*AttrN*>) and generates the least (most specific) common generalization of both.**
- **The function does a dynamical programming search for the best match between the two patterns.**
- **The cost of the match is defined as the sum of costs of matches for all elements.**
  - **two identical elements match at no cost,**
  - **a token matches a skip or an empty space at cost 2,**
  - **a skip matches an empty space at cost 1.**
  - **All other combinations have infinite cost.**
- **After the best match is found, it is converted into a pattern by copying matched identical elements and adding skips where non-identical elements are matched.**

Example

- ***S1 = "Toward this end, <Arg1> in July acquired <Arg2>"***
- ***S2 = "Earlier this year, <Arg1> acquired <Arg2>"***
- **After the dynamical programming-based search, the following match will be found:**

Toward		(cost 2)
	Earlier	(cost 2)
this	*this*	(cost 0)
end		(cost 2)
	year	(cost 2)
,	,	(cost 0)
<Arg1 >	*<Arg1 >*	(cost 0)
in July		(cost 4)
acquired	*acquired*	(cost 0)
<Arg2 >	*<Arg2 >*	(cost 0)

Generating the Pattern

- **at total cost = 12. The match will be converted to the pattern**
  - **** * this * * , <Arg1> * acquired <Arg2>***
- **which will be normalized (after removing leading and trailing skips, and combining adjacent pairs of skips) into**
  - ***this * , <Arg1> * acquired <Arg2>***

Post-processing, filtering, and scoring of patterns

- **In the first step of the post-processing we remove from each pattern all function words and punctuation marks that are surrounded by skips on both sides. Thus, the pattern from the example above will be converted to**

, *<Arg1> * acquired <Arg2>*

- **Note, that we do not remove elements that are adjacent to meaningful words or to slots, like the comma in the pattern above, because such anchored elements may be important.**

Content Based Filtering

- **Every pattern must contain at least one word relevant to its predicate. For each predicate, the list of relevant words is automatically generated from WordNet by following all links to depth at most 2 starting from the predicate keywords. For example, the pattern**

  <Arg1> * by <Arg2>

- **will be removed, while the pattern**

  <Arg1> * purchased <Arg2>

- **will be kept, because the word "*purchased*" can be reached from "*acquisition*" via synonym and derivation links.**

Scoring the Patterns

- **The filtered patterns are then scored by their performance on the positive and negative sets.**
- **We want the scoring formula to reflect the following heuristic: it needs to rise monotonically with the number of positive sentences it matches, but drop very fast with the number of negative sentences it matches.**

$$Score(Pattern) = \frac{\left|S \in PositiveSet : Pattern \text{ matches } S\right|}{\left(\left|S \in NegativeSet : Pattern \text{ matches } S\right| + 1\right)^2}$$

Sample Patterns - Inventor

- X , .* inventor .* of Y
- X invented Y
- X , .* invented Y
- when X .* invented Y
- X ' s .* invention .* of Y
- inventor .* Y , X
- Y inventor X
- invention .* of Y .* by X
- after X .* invented Y
- X is .* inventor .* of Y
- inventor .* X , .* of Y
- inventor of Y , .* X ,
- X is .* invention of Y
- Y , .* invented .* by X
- Y was invented by X

Sample Patterns – CEO (Company/X,Person/Y)

- X ceo Y
- X ceo .* Y ,
- former X .* ceo Y
- X ceo .* Y .
- Y , .* ceo of .* X ,
- X chairman .* ceo Y
- Y , X .* ceo
- X ceo .* Y said
- X ' .* ceo Y
- Y , .* chief executive officer .* of X
- said X .* ceo Y
- Y , .* X ' .* ceo
- Y , .* ceo .* X corporation
- Y , .* X ceo
- X ' s .* ceo .* Y ,
- X chief executive officer Y
- Y , ceo .* X ,
- Y is .* chief executive officer .* of X

Shallow Parser mode

- **In the first mode of operation (without the use of NER), the predicates may define attributes of two different types: P*roperName* and *CommonNP*.**
- **We assume that the values of the *ProperName* type are always heads of proper noun phrases. And the values of the *CommonNP* type are simple common noun phrases (with possible proper noun modifiers, e.g. "*the Kodak camera*").**
- **We use a Java-written shallow parser from the OpenNLP (http://opennlp.sourceforge.net/) package. Each sentence is tokenized, tagged with part-of-speech, and tagged with noun phrase boundaries. The pattern matching and extraction is straightforward.**

Building a Classification Model

- **The goal is to set the score of the extractions using the information on the instance, the extracting patterns and the matches. Assume, that extraction *E* was generated by pattern *P* from a match *M* of the pattern *P* at a sentence *S*. The following properties are used for scoring:**
  1. **Number of different sentences that produce *E* (with any pattern).**
  2. **Statistics on the pattern *P* generated during pattern learning – the number of positive sentences matched and the number of negative sentences matched.**
  3. **Information on whether the slots in the pattern *P* are anchored.**
  4. **The number of non-stop words the pattern *P* contains.**
  5. **Information on whether the sentence *S* contains proper noun phrases between the slots of the match *M* and outside the match *M*.**
  6. **The number of words between the slots of the match *M* that**

Building a Classification Model

- **During the experiments, it turned out that the pattern statistics (2) produced detrimental results, and the proper noun phrase information (5) did not produce any improvement. The rest of the information was useful, and was turned into the following set of binary features:**
  - ***f*1(*E*, *P*, *M*, *S*) = 1, if the number of sentences producing *E* is greater than one.**
  - ***f*2(*E*, *P*, *M*, *S*) = 1, if the number of sentences producing *E* is greater than two.**
  - ***f*3(*E*, *P*, *M*, *S*) = 1, if at least one slot of the pattern *P* is anchored.**
  - ***f*4(*E*, *P*, *M*, *S*) = 1, if both slots of the pattern *P* are anchored.**

Building a Classification Model

  - ***f*5…*f*9(*E*, *P*, *M*, *S*) = 1, if the number of nonstop words in *P* is 0, 1 or greater, 2 or greater,… 4 or greater, respectively**
  - ***f*10…*f*15(*E*, *P*, *M*, *S*) = 1, if the number of words between the slots of the match *M* that were matched to skips of the pattern *P* is 0, 1 or less, 2 or less, 3 or less, 5 or less, and 10 or less, respectively.**
- **As can be seen, the set of features above is rather small, and is not specific to any particular predicate. This allows to train a model using a small amount of labeled data for one predicate, and then to use the model for all other predicates.**

Using an NER Component

- **In the SRES-NER version the entities of each candidate instance are passed through a simple rule-based NER filter, which attaches a score ("yes", "maybe", or "no") to the argument(s) and optionally fixes the arguments boundaries. The NER is capable of identifying entities of type PERSON and COMPANY (and can be extended to identify additional types).**

NER Scores

- **The scores mean:**
  - **"yes" – the argument is of the correct entity type.**
  - **"no" – the argument is not of the right entity type, and hence the candidate instance should be removed.**
  - **"maybe" – the argument type is uncertain, can be either correct or no.**

Utilizing the NER Scores

- **If "no" is returned for one of the arguments, the instance is removed. Otherwise, an additional binary feature is added to the instance's vector:**
  - **f_{16} = 1 iff the score for both arguments is "yes".**
- **For bound predicates, only the second argument is analyzed, naturally.**

Experimental Evaluation

- **We want to answer the following 4 questions:**
  1. **Can we train SRES's classifier once, and then use the results on all other relations?**
  2. **What boost will we get by introducing a simple NER into the classification scheme of SRES?**
  3. **How does SRES's performance compare with KnowItAll and KnowItAll-PL?**
  4. **What is the true recall of SRES?**

Training

1. **The patterns for a single model predicate are run over a small set of sentences (10000 sentences in our experiment), producing a set of extractions (between 150-300 extractions in our experiments).**
2. **The extractions are manually labeled according to whether they are correct or no.**
3. **For each pattern match *Mk*, the value of the feature vector *fk* = (*f*1,…*f*16) is calculated, and the label *Lk* = ±1 is set according to whether the extraction that the match produced is correct or no.**
4. **A regression model estimating the function *L*(*f*) is built from the training data {(*fk*, *Lk*)}. We used the BBR, but other models, such as SVM are of course possible.**

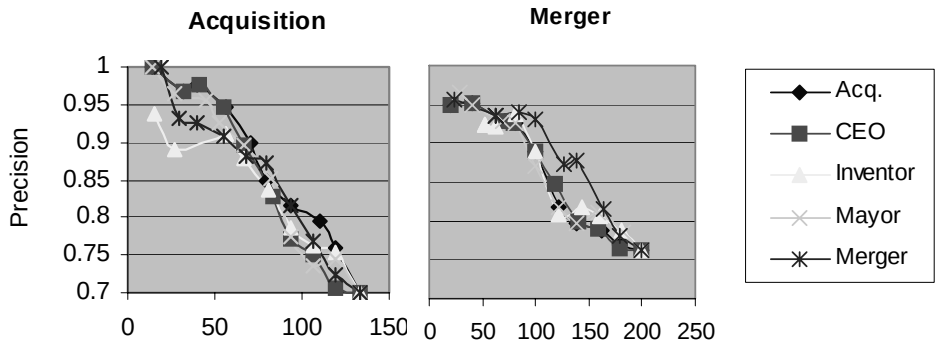
Testing

1. **The patterns for all predicates are run over the sentences.**
2. **For each pattern match *M*, its score *L*(*f*(*M*)) is calculated by the trained regression model. Note that we do not threshold the value of *L*, instead using the raw probability value between zero and one.**
3. **The final score for each extraction is set to the maximal score of all matches that produced the extraction.**

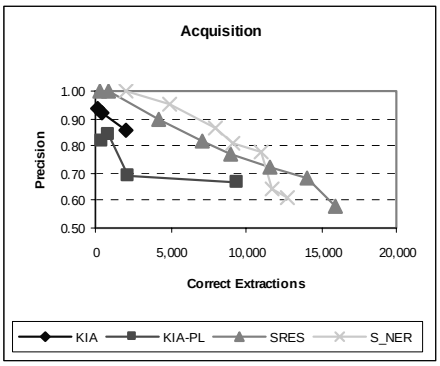
Sample Output

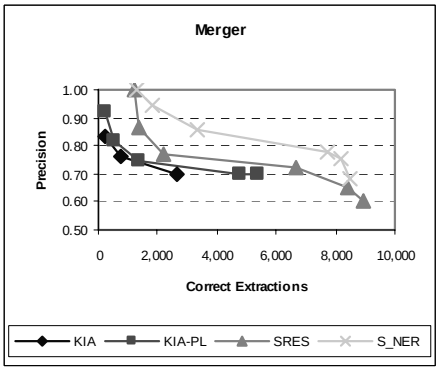
- **<e> <arg1>HP</arg1> <arg2>Compaq</arg2>**
  - **<s><DOCUMENT>Additional information about the <X>HP</X> -<Y>Compaq</Y> merger is available at www.VotetheHPway.com .</DOCUMENT></s>**
  - **<s><DOCUMENT>The Packard Foundation, which holds around ten per cent of <X>HP</X> stock, has decided to vote against the proposed merger with <Y>Compaq</Y>.</DOCUMENT></s>**
  - **<s><DOCUMENT>Although the merger of <X>HP</X> and <Y>Compaq</Y> has been approved, there are no indications yet of the plans of HP regarding Digital GlobalSoft.</DOCUMENT></s>**
  - **<s><DOCUMENT>During the Proxy Working Group's subsequent discussion, the CIO informed the members that he believed that Deutsche Bank was one of <X>HP</X>'s advisers on the proposed merger with <Y>Compaq</Y>.</DOCUMENT></s>**
  - **<s><DOCUMENT>It was the first report combining both <X>HP</X> and <Y>Compaq</Y> results since their merger.</DOCUMENT></s>**
  - **<s><DOCUMENT>As executive vice president, merger integration, Jeff played a key role in integrating the operations, financials and cultures of <X>HP</X> and <Y>Compaq</Y> Computer Corporation following the 19 billion merger of the two companies.</DOCUMENT></s>**

Cross-Classification Experiment

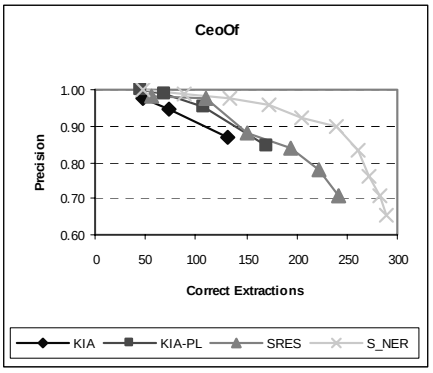


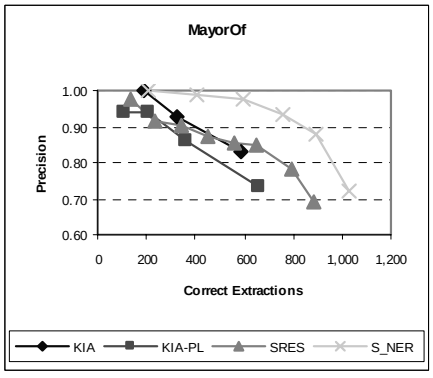
Results!



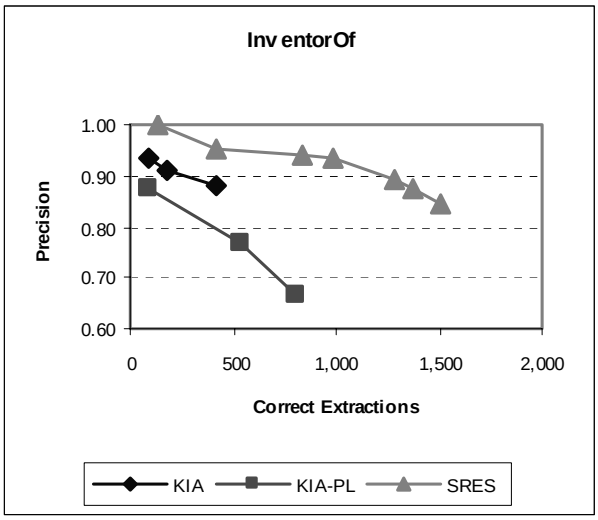


More Results





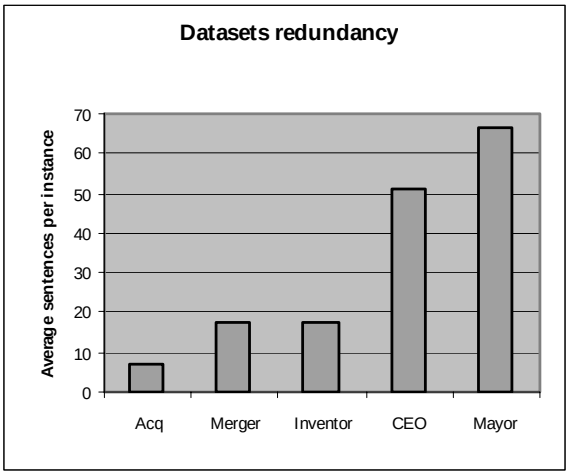
Inventor Results



When is SRES better than KIA?

- **KnowItAll extraction works well when redundancy is high and most instances have a good chance of appearing in simple forms that KnowItAll is able to recognize.**
- **The additional machinery in SRES is necessary when redundancy is low.**
- **Specifically, SRES is more effective in identifying low-frequency instances, due to its more expressive rule representation, and its classifier that inhibits those rules from overgeneralizing.**

The Redundancy of the Various Datasets



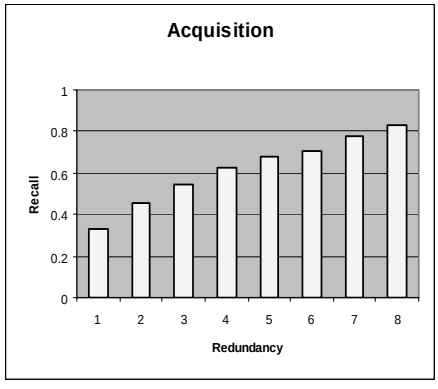
True Recall Estimates

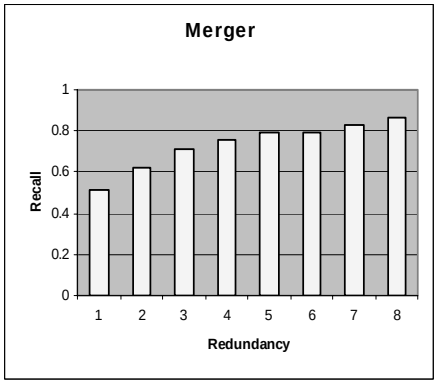
- **It is impossible to manually annotate all of the relation instances because of the huge size of the input corpus.**
- **Thus, indirect methods must be used. We used a large list of known acquisition and merger instances (that occurred between 1/1/2004 and 31/12/2005) taken from the paid service subscription SBC Platinum.**
- **For each of the instances in this list we identified all of sentences in the input corpus that contained both instance attributes and assumed that all such sentences are true instances of the corresponding relation.**

Under Estimation of the recall

- **This is of course an overestimate since in some cases the appearance of both attributes of a true relation instance is just a chance occurrence and does not constitute a true mention of the relation.**
- **Thus, our estimates of the true recall are pessimistic, and the actual recall is higher.**

True Recall Estimates





Conclusions

- **We have presented the SRES system for autonomously learning relations from the Web.**
- **SRES solves the bottleneck created by classic information extraction systems that either relies on manually developed extraction patterns or on manually tagged training corpus.**
- **The system relies upon a pattern learning component that enables it to boost the recall of the system.**

Future Work

- **In our future research we want to try to improve the precision values even at the highest recall levels.**
- **One of the topics we would like to explore is the complexity of the patterns that we learn. Currently we use a very simple pattern language that just has 3 types of elements, slots, constants and skips. We want to see if we can achieve higher precision with more complex patterns.**
- **In addition we would like to test SRES on n-ary predicates, and to extend the system to handle predicates that are allowed to lack some of the attributes.**