Named Entity Extraction

-

MEM & Co-training

PD Dr. Günter Neumann
DFKI and Saarland University

MENE [Borthwick et al 98]

- Combining rule-based and ML NE to achieve better performance
- Tokens tagged as: XXX_start, XXX_continue, XXX_end, XXX_unique, other (non-NE), where XXX is an NE category
- Uses Maximum Entropy
  - One only needs to find the best features for the problem
  - ME estimation routine finds the best relative weights for the features

From Cunningham & Bontcheva, 2003

Core idea of MEM

- Probability for a class Y and an object X depends solely on the features that are „active" for the pair (X,Y)
- Features are the means through which an experimenter feeds problem-specific information
- The importance of each feature is determined automatically by running a parameter estimation algorithm over pre-classified set of examples („training-set")
- Advantage: experimenter need only tell the model what information to use, since the model will automatically determine how to use it.

Maximum Entropy Modeling

- Random process
  - produces an output value y, a member from a finite set Y
  - Might be influenced by some contextual information x, a member from a finite set X
- Construct a stochastic model that accurately describes the random process
  - Estimate the conditional probability P(Y|X)
- Training data: $(x_1, y_1) , (x_2, y_2) , ..., (x_N, y_N)$

$$r(x, y) \equiv \frac{c(x, y)}{N}$$

Simple example

- Task: estimate a joint probability distribution p defined over $\{x,y\}\times\{0,1\}$
- Known facts (constraints) about p
  - $p(x,0)+p(y,0)=0.6$
  - $p(x,0)+p(y,0)+p(x,1)+p(y,1)=1$

P(a,b)	0	1	
X	?	?	
Y	?	?	
Total	.6		1

One way to satisfy constraints

P(a,b)	0	1	
X	.5	.1	
Y	.1	.3	
Total	.6		1

Is this also the most accurate one?

Simple Example

- Observed facts are constraints for the desired model p
- Observed fact $p(x,0)+p(y,0)=0.6$ is implemented as a constraint of feature f_1 of model p, $E_p f_1$, where

$$E_p f_1 = \sum_{a \in \{x,y\}, b \in \{0,1\}} p(a,b) f_1(a,b) \qquad f_1(a,b) = \begin{cases} 1 & if \;\; b = 0 \\ 0 & otherwise \end{cases}$$

Most uncertain way to satisfy constraints:

P(a,b)	0	1	
X	.3	.2	
Y	.3	.2	
Total	.6	.4	1

Histories, binary features & futures

- History b: information derivable from the corpus relative to a token:
  - text window around token w_i, e.g. $w_{i-2},\dots,w_{i+2}$
  - word features of these tokens
  - POS, other complex features
- Features:
  - yes/no-questions on history used by models to determine probabilities of
- Futures: what we are predicting (e.g., POS, name classes)

Features represent evidence

- a = what we are predicting (e.g., tags)
- b = what we observe (e.g., words)
- A feature f has the form

  $f_{y,q}(a,b)=1$ if $a=y$ & $q(b)$ = true

  0 otherwise

- E.g.,

  $f_{NNP,q1}(a,b)=1$ if $a=NNP$ & $q1(b)$ = true

  $f_{VBG,q2}(a,b)=1$ if $a=VBG$ & $q2(b)$ = true

Weight features with conditional probability model

$$P(a \mid b) = \frac{\prod_{j=1}^{k} \alpha_j^{f_j(a,b)}}{Z(b)} = \frac{\prod_{j=1}^{k} \alpha_j^{f_j(a,b)}}{\sum_a \prod_{j=1}^{k} \alpha_j^{f_j(a,b)}}$$

- Z(b) = normalization factor
- $\alpha_j > 0$: weights for feature f_j
- P(a|b): (normalized) product of weights of active feature on the (a,b) pair, i.e., those features f_j such that f_j (a,b)=1

MENE (2)

- Features
  - Binary features – "token begins with capitalised letter", "token is a four-digit number"
  - Lexical features – dependencies on the surrounding tokens (window ±2) e.g., "Mr" for people, "to" for locations
  - Dictionary features – equivalent to gazetteers (first names, company names, dates, abbreviations)
  - External systems – whether the current token is recognised as an NE by a rule-based system

From Cunningham & Bontcheva, 2003

MENE (3)

- MUC-7 formal run corpus
  - MENE – 84.2% f-measure
  - Rule-based systems it uses – 86% - 91 %
  - MENE + rule-based systems – 92%
- Learning curve
  - 20 docs – 80.97%
  - 40 docs – 84.14%
  - 100 docs – 89.17%
  - 425 docs – 92.94%

From Cunningham & Bontcheva, 2003

Information Extraction

-

Bootstrapping NE lists

PD Dr. Günter Neumann
DFKI and Saarland University

Details of Bootstrapping approaches

- Bootstrapping classical NE types
  - Michael Collins and Yoran Singer, 1999

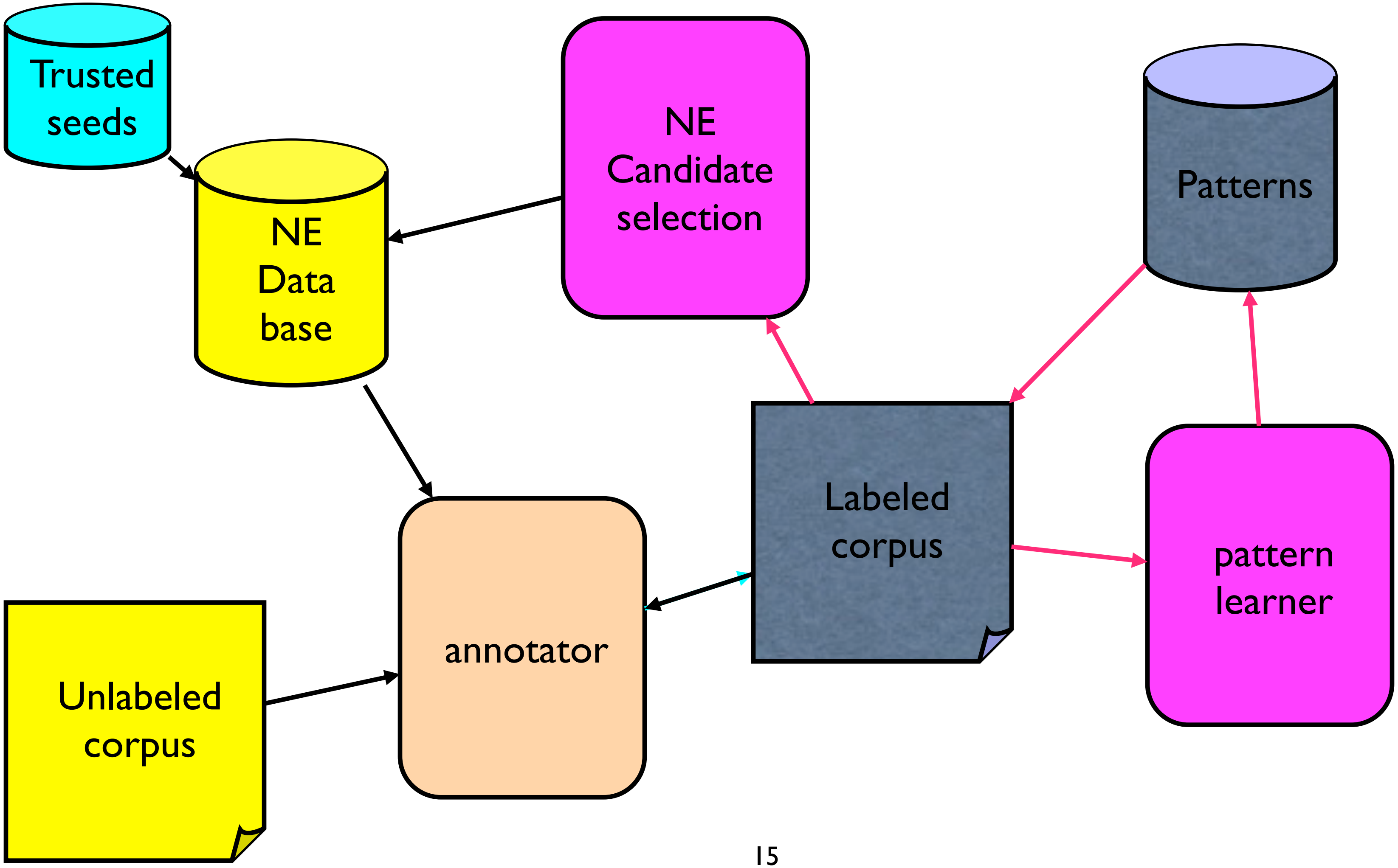
- Bootstrapping generalized names
  - Yangarber, Lin, Grishman, 2002
  - Lin, Yangarber, Grishman, 2003

- Context Pattern Induction method
  - Talukdar, Brants, Liberman, Pereira, 2006

Bootstrapping NE: idea

- Define manually only a small set of trusted seeds
- Training then only uses un-labeled data
- Initialize system by labeling the corpus with the seeds
- Extract and generalize patterns from the context of the seeds
- Use the patterns to further label the corpus and to extend the seed set (bootstrapping)
- Repeat the process until no new terms can be identified

Bootstrapping NE-learning: idea



Bootstrapping NE classification

based on Michael Collins and Yoran Singer, EMNLP 1999

- The task: to learn a decision list to classify strings as person, location or organization

The learned decision list is an *ordered* sequence of if-then rules

… says Mr. Gates, founder of Microsoft …

R_1 : if <u>features</u> then person
R_2 : if <u>features</u> then location
R_3 : if <u>features</u> then organization
…
R_n : if <u>features</u> then person

… says Mr. Gates, founder of Microsoft …

Outline of Bootstrapping Co-Training

- Parse an unlabeled document set
- Extract each NP, whose head is tagged as proper noun
- Define a set of relevant features, which can be applied on extracted NPs
- Define two separate types of rules on basis of feature space
- Determine small initial set of seed rules
- Iteratively extend the rules through co-training

Two Categories of Rules

- The key to the method is redundancy in the two kind of rules.

…says Mr. Cooper, a vice president of…

Paradigmatic or spelling

Syntagmatic or contextual

Huge amount of unlabeled data gives us these hints!

The Data

- 971,746 New York Times sentences were parsed using full sentence parser.

- Extract consecutive sequences of proper nouns (tagged as NNP and NNPS) as named entity examples if they met one of following two criterion.

- Note: thus seen, NNP(S) functions as a generic NE-type, and the main task is now to sub-type them.

Kinds of Noun Phrases

1. There was an appositive modifier to the NP, whose head is a singular noun (tagged NN).

   …says [Maury Cooper], [a vice president]…

2. The NP is a complement to a preposition which is the head of a PP. This PP modifies another NP whose head is a singular noun.

   … fraud related to work on [a federally funded sewage plant] [in [Georgia]].

(spelling, context) pairs created

- *…says Maury Cooper, a vice president…*
    - (Maury Cooper, president)
- *… fraud related to work on a federally funded sewage plant in Georgia.*

    (Georgia, plant_in)

Features

representing examples for the learning algorithm

- Set of spelling features
  - Full-string=x (full-string=Maury Cooper)
  - Contains(x) (contains(Maury))
  - Allcap1 IBM
  - Allcap2 N.Y.
  - Nonalpha=x A.T.&T. (nonalpha=..&.)
- Set of context features
  - Context = x (context = president)
  - Context-type = x appos or prep

It is strongly assumed that the features can be partitioned into two types such that each type alone is sufficient for classification

Examples of named entities and their features

Sentence	Entities(Spelling/Context)	(Active) Features
But Robert Jordan, a partner at Steptoe & Johnson who took …	Robert Jordon/partner	Full-string=Robert_Jordan, contains(Robert), contains (Jordan), context=partner, context-type=appos
	Steptoe & Johnson/partner_at	Full-string=Steptoe_&_Johnson, contains(Steptoe), contains(&), contains(Johnson), nonalpha=& , context=partner_at, context-type=prep
By hiring a company like A.T.&T. …	A.T.&T./company_like	Full-string= A.T.&T., allcap2, nonalpha=..&. , context=company_like, context-type=prep
Hanson acquired Kidde Incorporated, parent of Kidde Credit, for …	Kidde Incorporated/parent	Full-string=Kidde_Incorporated, contains(Kidde), contains(Incorporated), context=parent, context-type=appos
	Kidde Credit/parent_of	Full-string=Kidde_Credit, contains(Kidde), contains (Credit), context=parent_of, context-type=prep

Rules

Two separate types of rules:
Spelling rules
Context rules

h(x,y): the strength of a rule, defined as

Feature → NE-type, h(Feature,NE-type)

$$\arg\max_{x,y} \frac{Count(x,y)+\alpha}{Count(x)+k\alpha}$$

Is an estimate of the conditional probability of the NE-type given the feature, P(y|x)

where

$$Count(x) = \sum_{y \in Y} Count(x,y)$$

α is a smoothing parameter

$k = \#NE\text{-}types$

The rules ordered according to their strengths h form a decision list: the sequence of rules are tested in order, and the answer to the ***first*** satisfied rule is output.

7 SEED RULES

Note: only one type of rules used as seed rules, and all NE-types should be covered

- Full-string = New York → Location
- Full-string = California → Location
- Full-string = U.S. → Location
- Contains(Mr.) → Person
- Contains(Incorporated) → Organization
- Full-string=Microsoft → Organization
- Full-string=I.B.M. → Organization

The Co-training algorithm

1. Set N=5 (max. # of rules of each type induced in each iteration)
2. **Initialize**: Set the spelling decision list equal to the set of seed rules. Label the training set using these rules.
3. Use these to get contextual rules. (x = feature, y = label)

   Compute h(x,y), and induce at most N * K rules

   all must be above some threshold p_{min}=0.95
4. Label the training set using the contextual rules.
5. Use these to get N*K spelling rules (same as step 3.)
6. Set spelling rules to seed plus the new rules.
7. If N < 2500, set N=N+5, and goto step 3.
8. Label the training data with the combined spelling/contextual decision list, then induce a final decision list from the labeled examples where all rules (regardless of strength) are added to the decision list.

Example

- (IBM, company)

  …IBM, the company that makes…

- (General Electric, company)

  ..General Electric, a leading company in the area,…

- (General Electric, employer)

  … joined General Electric, the biggest employer…

- (NYU, employer)

  NYU, the employer of the famous Ralph Grishman,…

Why Separate Spelling, Context Features?

Can use theory behind co-training to explain how algorithm works

f_i must correctly classify labeled examples, and

must agree with each other on unlabeled ex.

Open question: best similarity function?

3. & 4. Say that features can be partitioned.

Requirements:

- Classification problem $f: X \rightarrow Y$
  - $f_1(x_{1,i}) = f_2(x_{2,i}) = y_i$ for $i = 1 \ldots m$
  - $f_1(x_{1,i}) = f_2(x_{2,i})$ for $i = m+1 \ldots n$
  - (softer criteria requires f_1 and f_2 to minimize their disagreement similarity)
- Can partition features X into 2 types of features $x = (x_1,x_2)$
- Each type is sufficient for classification
- x_1,x_2 not correlated too tightly (e.g., no deterministic function from x_1 to x_2)

The Power of the Algorithm

- Greedy method
  - At each iteration method increases number of rules
  - While maintaining a high level of agreement between spelling & context rules

For n= 2500:

The two classifiers give both labels on 49.2% of the unlabeled data
And give the same label on 99.25% of these cases

➢ The algorithm maximizes the number of unlabeled examples on which the two decision lists agree.

Evaluation

- 88,962 (spelling, context) pairs.
    - 971,746 sentences
- 1,000 randomly extracted to be test set.
- Location, person, organization, noise (items outside the other three)
- 186, 289, 402, 123 (- 38 temporal noise).
- Let N_c be the number of correctly classified examples
    - Noise Accuracy: N_c / 962
    - Clean Accuracy: N_c /(962-85)

Results

Algorithm	Clean Accuracy	Noise Accuracy
Baseline	45.8%	41.8%
EM	83.1%	75.8%
Yarowsky 95	81.3%	74.1%
Yarowsky Cautious	91.2%	83.2%
DL-CoTrain	91,3 %	83,3 %
CoBoost	91.1%	83.1%

Remarks

- Needs full parsing of unlabeled documents
  - Restricted language independency
  - Need linguistic sophistication for new types of NE
- Slow training
  - In each iteration, full size of training corpus has to be re-labeled