

Named Entity Recognition

Fang Li
**Dept. of Computer Science &
Technology**

Contents

- **Problems on Entity Identification**
- **Features of Entity Identification**
- **Rule-based method**
- **Machine learning method**

Named Entity Recognition

*“TWA has not been a normal **company**,” said Robert Peiser, chief financial officer. We can not substantiate the claims.*

Entities include:

- **Named entities:** *TWA, Robert Peiser ...*
- **pronoun entities:** *we, ...*
- **nominal entities:** *the company, ...*

Named entities are the most important one among the 3 categories which is the anchor point for IE.

Difficulties of NE recognition

- ❑ Potential set of NE is **too large** to include in dictionaries/Gazetteers.
- ❑ Names changing constantly.
- ❑ Names appear in **many variant forms**. E.g. John Smith, Mr Smith or John
- ❑ Subsequent occurrences of names might be **abbreviated**.
- ❑ **Ambiguity** of NE types. E.g. John Smith is a person name or a company name? depends on:
 - Internal structure: **Mr.** John Smith
 - Context: The new **company**, John Smith will make....
 - Ambiguity: **Washington** is a person or a location?

Features of Named Entities

According to its position in the text:

- **Features** that occur in the **information unit itself**, such as the composition of letters and digits of an entity name.
- **Features** that **close neighborhood** or **context window** of the token string to be classified.
- **Features** that occur in the **complete document or document collection**.

Features of Named Entities

According to their **types**

- ❑ **Lexical**: variations concerning punctuation (USA versus U.S.A), capitalization (e.g., Citibank versus CITIBANK)
- ❑ **Syntactic**: The part-of-speech of a word.词性
- ❑ **Semantic**: refer to semantic classifications of single- or multi-word information units.
- ❑ **Discourse features** refer to features computed by using text fragments, larger than the sentence.

Use Lexical & Context Features

□ Set of **lexical features**

- Full-string=x (full-string=Maro Cooper)
- Contains(x) (contains(Maco))
- Allcap1 (IBM)
- Allcap2 (N.Y.)
- Nonalpha=x A.T.&T. nonalpha=.&.

□ Set of **context features**

- Context=x (context=president)
- Context-type=x (prep or apposition)

特征选择 与 特征表示

- *Feature selection*: use which features to identify NE (e.g.person name)
- *Different methods choose different features.*
- For rule-based method: to design **Patterns or Rules**
- For Machine Learning method:
Features → **features of models** → train → model

Features can have: 特征值类型

- ✓ **Numeric values: discrete or real values** (e.g: 字符个数)
- ✓ **Boolean value: true or false** (上下文窗口是否包含 Mr. Dr.)
- ✓ **Nominal values: certain words** (词性=名词, 形容词, 动词)
- ✓ **Ordinal values: 0=small, 1=medium, 2=large**
- ✓ **Interval or ratio scaled values** (1/10, 2/10)

Pattern vs. Rule

□ Pattern (like regular expressions)

Price pattern(**P**):

`\b[1-9][0-9]{[0-9]}\.[1-9]{0-6}\b`

Person name pattern (**P1**): 姓+名

姓属于（姓名库），名：任意的单字或双字

□ Rule

If x match P then x is a **price**.

If x match P1 then x is a **person**.

Steps for Rule-based Method

1. Build linguistic **patterns** or **rules** to identify Entities or Relations

"Dr.Yiming Yang was appointed as CEO of IBM at ..."

"Smith was appointed as chairman of the account board. →"

Pattern:

person be appointed as post of company

Steps for Rule-based Method (cont.)

2. Apply rules or patterns to text and extraction

"Smith was appointed as Akim of Akmola region" →

Person: Smith

Post: Akim (head of local government)

Company: Akmola region X

Steps for Rule-based Method (cont.)

3. ***Run over a corpus, find errors:***

- ✓ General Electric is a company, not a general
- ✓ Yesterday IBM Corp. announced ...

How to fix?

Dictionary or lexicon are needed.

Pattern Needs

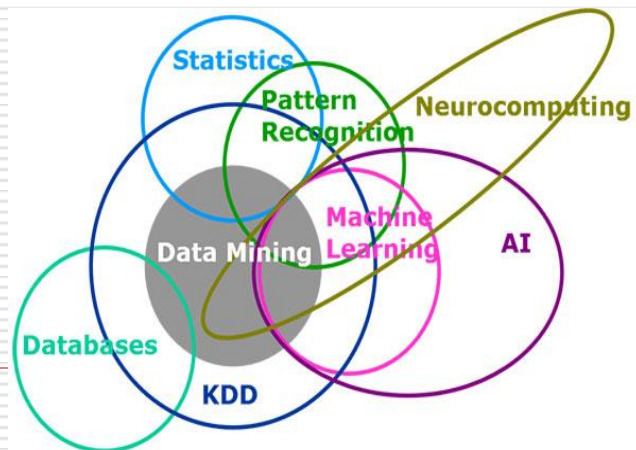
- ❑ **general enough:** to have a broad applicability.
- ❑ **specific enough:** to be consistently reliable over a large number of texts.
- ❑ For example:
 - "*Person, post convinced company*" → ***too general***
 - "*company named person to post*" → ***too specific***

Who build patterns?

□ Human experts



□ Machine automatically learned from data.



Use a lexicon

- Advantages - Simple, fast, language independent, easy to retarget.
- Disadvantages - collection and maintenance of lists, cannot deal with name variants, **cannot resolve ambiguity and include all.**

Some Dictionaries

- WordNet: 词的概念语义知识库
- FrameNet: 动词和它的属性
- Paraphrase Database(PPDB): 词和短语的复述数据库
- 公司命名库
- 地名库
- 。 。 。

For example:

Birthdate Extraction (正例与反例)

- ✓ George Washington was born in 1725.
- ✓ Washington was born on Feb. 12, 1725.
- ✓ Feb. 12 is Washington's birthday.
- ✓ Washington's birth date is Feb. 12, 1725.

— George Washington was born in America.

— Washington's standard was born by his troops in 1778.

Some Rules for birthday extraction

- $\langle \text{Name} \rangle$ "was born" {"in"|"on"} $\langle \text{Date} \rangle$
=> extraction (Name, Date)
- $\langle \text{Date} \rangle$ "is" $\langle \text{Name, possessive} \rangle$
"birthday"
=> extraction (Name, Date)
- $\langle \text{Name, possessive} \rangle$ "birth" "date" "is"
 $\langle \text{Date} \rangle$
=> Extraction (Name, Date)

Machine Learning Method (idea)

□ Based on Probability:

"smith was appointed as CEO of IBM"

Category: ***Person, position, company, nn.***

If we know:

$$P(\text{smith} \mid \text{person}) = 0.8 \quad \checkmark$$

$$P(\text{smith} \mid \text{company}) = 0.1$$

$$P(\text{smith} \mid \text{position}) = 0.05$$

$$P(\text{smith} \mid \text{nn}) = 0.05$$

Machine learning method (idea)

□ Sequence labeling:

W: smith was appointed as CEO of IBM

NC : Per nn nn nn Pos nn CO

NC: CO Per nn nn Per nn Per

.....

$P(\text{NC sequences} \mid \text{W sequence})$

Which NC sequences has a larger probability?

Machine learning method (idea)

□ Classification

W: smith was appointed as CEO of IBM



Classification Model



Category: *Person, position, company, nn*

Machine learning for NE recognition

□ **Supervised Learning**

- Training is based on available very large **annotated** corpus.
 - Some models are
 1. *Bigram model*
 2. *HMM (马尔可夫) - MEMM (最大熵马尔可夫)*
 3. *CRF (conditional random field) (条件随机场)*
-

Supervised Machine Learning for NE Recognition

如何标注呢？

- 1. Construct a training corpus by manual annotation*
 - Extract necessary statistics from the corpus to build a statistical model which can automatically **estimate** $\Pr(\text{NC Sequence} \mid \text{W Sequence})$ for unseen data.*
- 2. For any unseen data, based on the statistical model to search the NC sequence which **maximizes** the probability $\Pr(\text{NC Sequence} \mid \text{W Sequence})$.*

Encoding classes for sequence labeling

C classes,
C+1 labels

□ **IO encoding**

Fred showed Sue Mengqiu Huang

Per nn Per Per Per

C classes,
2C+1 labels

□ **IOB encoding**

Fred showed Sue Mengqiu Huang

B-Per nn B-Per B-Per I-Per

IO encoding is simple, much faster than IOB encoding

Statistical Model for Named Entity Recognition

$$\operatorname{argmax}_{\text{nc sequence}} \Pr(\text{NC Sequence} \mid \text{W Sequence})$$

e.g, given word sequence :

it has set up a joint venture in Hong Kong

possible name-class sequence (LO: location OR: organization)

it has set up a joint venture in Hong Kong

NN NN NN NN NN NN NN NN LO LO

LO NN NN NN NN NN NN NN OR LO

...

...

Bigram Model for NE Recognition

- *Question: How to evaluate $Pr(\text{NC Sequence} / \text{Sentence})$ based on unigram and bigram information?*
- *One solution: transfer the conditional probability into $(\text{NC}, \text{Sentence})$ joint probability (**Bayes' equation**)*
- *Decouple a sentence into bigram sequences (**Markov assumption**)*

Bayes Equation

Based on Bayes equation:

$$\begin{aligned} & \operatorname{argmax}_{\text{nc sequence}} \Pr(\text{NC Sequence} | \text{W Sequence}) \\ &= \operatorname{argmax}_{\text{nc sequence}} \frac{\Pr(\text{W Sequence}, \text{NC Sequence})}{\Pr(\text{W Sequence})} \\ &= \operatorname{argmax}_{\text{nc sequence}} \Pr(\text{W Sequence}, \text{NC Sequence},) \end{aligned}$$

Markov Assumption

$\Pr(\text{NCSequence}, W \text{ Sequence})$

$$= \Pr(w_n, nc_n, w_{n-1}, nc_{n-1}, \dots, w_0, nc_0)$$

$$= \Pr(w_0, nc_0) \Pr(w_1, nc_1 \mid w_0, nc_0) \Pr(w_2, nc_2 \mid w_1, nc_1, w_0, nc_0)$$

$$\dots \Pr(w_n, nc_n \mid w_{n-1}, nc_{n-1}, \dots, w_0, nc_0)$$

Bigram Markov assumption:

$$\Pr(w_2, nc_2 \mid w_1, nc_1, w_0, nc_0) = \Pr(w_2, nc_2 \mid w_1, nc_1)$$

.....

$$\Pr(w_n, nc_n \mid w_{n-1}, nc_{n-1}, \dots, w_0, nc_0) = \Pr(w_n, nc_n \mid w_{n-1}, nc_{n-1})$$

Bigram-based NE Tagger

So the final formula is:

$$\begin{aligned} & \Pr(\text{NCSequence}, \text{W Sequence}) \\ &= \Pr(w_0, nc_0) \Pr(w_1, nc_1 \mid w_0, nc_0) \Pr(w_2, nc_2 \mid w_1, nc_1) \\ & \dots \Pr(w_n, nc_n \mid w_{n-1}, nc_{n-1}) \end{aligned}$$

*The size of the training corpus is **large enough** to provide fairly good bigram information.*

Parameter Estimation based on Bayesian Analysis

- Question: **how to estimate model parameters, *i.e.***

$$\Pr(w_n, n_{C_n} | w_{n-1}, n_{C_{n-1}})$$

- Parameter estimation based on Bayesian analysis: select parameters which **maximize**

$$\Pr(\text{parameter} | \text{training corpus})$$

- Based on Bayes equation, this is equivalent to **maximize**

$$\Pr(\text{parameter})\Pr(\text{training corpus} | \text{parameter})$$

Prior Probability

Maximum Likelihood Estimation (MLE)

- find the parameter value(s) that can predict the training corpus with the highest probability.

$$\operatorname{argmax}_{\text{parameter}} \Pr(\text{training corpus} \mid \text{parameter})$$

i.e. the prior probability is neglected. C is the count

- The MLE for the bigram statistical NE tagger:

$$\begin{aligned} & \Pr(w_n, nc_n \mid w_{n-1}, nc_{n-1}) \\ &= \frac{C(w_n, nc_n, w_{n-1}, nc_{n-1})}{C(w_{n-1}, nc_{n-1})} \end{aligned}$$

Smoothing (平滑技术)

- limited size of training corpus \rightarrow MLE suffers from training data over-fitting. MLE simply assign zero or even $\frac{0}{0}$ probabilities to unseen events.
- Smoothing: *add one* or modify MLE by taking the sampling space into consideration, *e.g. backing-off to estimations with larger sampling space*

$$\Pr(\mathbf{w}_n, \mathbf{nc}_n \mid \mathbf{w}_{n-1}, \mathbf{nc}_{n-1}) \\ = \frac{C(\mathbf{w}_n, \mathbf{nc}_n, \mathbf{w}_{n-1}, \mathbf{nc}_{n-1}) + 1}{C(\mathbf{w}_{n-1}, \mathbf{nc}_{n-1}) + 1}$$

Smoothing (平滑技術)

- Unseen bigrams.

e.g. Input sentence: Patt Gibbs

$$\Pr(Gibbs, nc_{Gibbs} | Patt, nc_{Patt})=0$$

- Smoothing: **modify MLE by taking the sampling space into consideration**, *e.g. backing-off to estimations with larger sampling space*

$$\begin{aligned} & \Pr(Gibbs, NC_{Gibbs} | Patt, NC_{Patt}) \\ & \approx \lambda \Pr(Gibbs, NC_{Gibbs} | NC_{Patt}) \end{aligned}$$

CRF model

- ❑ Lafferty, Pereira, and McCallum proposed this model in 2001.
- ❑ **A best model** for named entity recognition.
- ❑ A sequence model, the theory is complicated and omitted.
- ❑ Training is slow.

General Working Flow

□ Training

1. Collect representative training documents
2. Label each token for its entity or other (nn)
3. **Design feature extractors** (templates)
4. Train the sequence model

□ Testing

1. Input test documents
2. Run sequence model to predict labels for each token
3. Correctly output the recognized entities (match the output format)

Features for sequence labeling (CRF)

- Words
 - Current word (essentially like a learned dictionary)
 - Previous/next word (context)
- Other kinds of inferred linguistic classification
 - Part-of-speech tags
- Label context
 - Previous (and perhaps next) label

CRF Training corpus for NE identification

Word	POS	Chunk	Short shape	Label
American	NNP	B-NP	Xx	B-ORG
Airlines	NNPS	I-NP	Xx	I-ORG
,	,	O	,	O
a	DT	B-NP	x	O
unit	NN	I-NP	x	O
of	IN	B-PP	x	O
AMR	NNP	B-NP	X	B-ORG
Corp.	NNP	I-NP	Xx.	I-ORG
,	,	O	,	O
immediately	RB	B-ADVP	x	O
matched	VBD	B-VP	x	O
the	DT	B-NP	x	O
move	NN	I-NP	x	O
,	,	O	,	O
spokesman	NN	B-NP	x	O
Tim	NNP	I-NP	Xx	B-PER
Wagner	NNP	I-NP	Xx	I-PER
said	VBD	B-VP	x	O
.	,	O	.	O

Feature Notation in CRF

Feature Notation	Comment
w_0	Current word (token)
w_1	Next word
w_{-1}	Previous word
w_0w_1	Current and next
$w_{-1}w_0$	Previous and current
$w_{-1}w_1$	Previous and next
w_2	Next next
...	...

CRF Feature Template for chunk identification (组块识别)

- Each line is a template, special macro %x[row,col] is used to specify a token in the input file.

```
Input: Data
He      PRP    B-NP
reckons VBZ    B-VP
the     DT     B-NP  << CURRENT TOKEN
current JJ    I-NP
account NN    I-NP
```

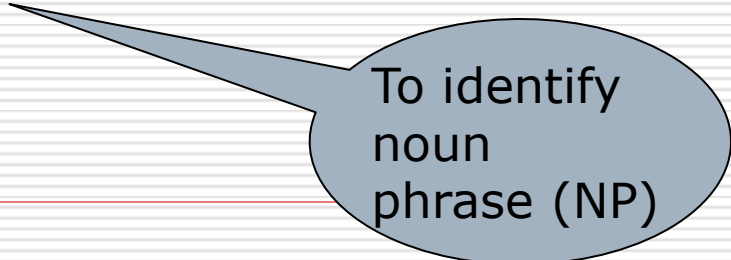
template	expanded feature
%x[0,0]	the
%x[0,1]	DT
%x[-1,0]	reckons
%x[-2,1]	PRP
%x[0,0]/%x[0,1]	the/DT
ABC%x[0,1]123	ABCDT123

The last column is the task :to identify noun phrase (NP) and verb phrase(VP)

Feature Template (cont.)

- When you give a template "U01:%x[0,1]", CRF++ automatically generates a set of feature functions (func1 ... funcN) like:

```
func1 = if (output = B-NP and feature="U01:DT") return 1 else return 0
func2 = if (output = I-NP and feature="U01:DT") return 1 else return 0
func3 = if (output = 0 and feature="U01:DT") return 1 else return 0
....
funcXX = if (output = B-NP and feature="U01:NN") return 1 else return 0
funcXY = if (output = 0 and feature="U01:NN") return 1 else return 0
....
```



To identify
noun
phrase (NP)

Example of a template

```
# Unigram
U00:%x[-2, 0]
U01:%x[-1, 0]
U02:%x[0, 0]
U03:%x[1, 0]
U04:%x[2, 0]
U05:%x[-1, 0]/%x[0, 0]
U06:%x[0, 0]/%x[1, 0]

U10:%x[-2, 1]
U11:%x[-1, 1]
U12:%x[0, 1]q
U13:%x[1, 1]
U14:%x[2, 1]
U15:%x[-2, 1]/%x[-1, 1]
U16:%x[-1, 1]/%x[0, 1]
U17:%x[0, 1]/%x[1, 1]
U18:%x[1, 1]/%x[2, 1]

U20:%x[-2, 1]/%x[-1, 1]/%x[0, 1]
U21:%x[-1, 1]/%x[0, 1]/%x[1, 1]
U22:%x[0, 1]/%x[1, 1]/%x[2, 1]

# Bigram
B
```

- Unigram template:
first character, 'U'
- Bigram template:
first character, 'B'
- U01:row-1,column 0
→previous word

Training command

❑ **crf_learn** *template_file train_file model_file*

❑ Parameters

-a CRF-L2 or CRF-L1: changing the regularization algorithm.

-c float: larger c, CRF tends to overfit to the given training corpus.

-f NUM: cut-off threshold. Use the features that occurs no less than NUM times in the given training data.

-p NUM: use multi-threading to faster the training step. NUM is the number of threads.

❑ **crf_learn -f 3 -c 1.5** *template_file train_file model_file*

Testing

❑ `crf_test -m model test_file`

❑ Parameter

-v sets verbose level. Default value is 0, Level 1 gives probabilities for each tag, and a conditional probability for the output.

-n best outputs: get n-best results sorted by the conditional probability of CRF

`crf_test -m model_file test_files >> result_file`

Software – CRF++

- <http://code.google.com/p/crfpp/> , its homepage is now at:
<http://crfpp.googlecode.com/svn/trunk/doc/index.html> ,
- Easy to use input & output format
- <http://crfpp.sourceforge.net/>
- CRFmodelforORG (课程网站上下载)

A CRF template to identify Chinese Organization

	U08:%x[-3, 1]↵	<u>当前词</u> 前面第三个词词性↵
U01:%x[-3, 0]↵	U09:%x[-2, 1]↵	<u>当前词</u> 前面第二个词词性↵
U02:%x[-2, 0]↵	U10:%x[-1, 1]↵	<u>当前词</u> 前面第一个词词性↵
U03:%x[-1, 0]↵	U11:%x[0, 1]↵	<u>当前词</u> 词性↵
U04:%x[0, 0]↵	U12:%x[1, 1]↵	<u>当前词</u> 后面第一个词词性↵
U05:%x[1, 0]↵	U13:%x[2, 1]↵	<u>当前词</u> 后面第二个词词性↵
U06:%x[2, 0]↵	U14:%x[3, 1]↵	<u>当前词</u> 后面第三个词词性↵
U07:%x[3, 0]↵	U15:%x[-2, 1]/%x[-1, 1]/%x[0, 1]↵	<u>当前词</u> 前面第二个词词性+ <u>当前词</u> 前面第一个词词性+ <u>当前词</u> 词性↵
	U16:%x[-1, 1]/%x[0, 1]/%x[1, 1]↵	<u>当前词</u> 前面第一个词词性+ <u>当前词</u> 词性+ <u>当前词</u> 后面第一个词词性↵
	U17:%x[0, 1]/%x[1, 1]/%x[2, 1]↵	<u>当前词</u> 词性+ <u>当前词</u> 后面第一个词词性+ <u>当前词</u> 后面第二个词词性↵
	U18:%x[-1, 1]/%x[0, 1]↵	<u>当前词</u> 前面第一个词词性+ <u>当前词</u> 词性↵
	U19:%x[0, 1]/%x[1, 1]↵	<u>当前词</u> 词性+ <u>当前词</u> 后面第一个词词性↵

新华社 nt B
 北京 ns O
 12月 m O
 30日 m O
 电 n O
 西藏 ns B
 自治区 n I
 政府 n I
 。 w O
 西藏 ns O
 部分 n O
 地区 n O
 发生 n O
 特大 v O
 雪灾 b O
 后 n O
 ， f O
 党中央 w O
 、 nt B
 国务院 w O
 十分 nt B
 关心 d O
 西藏 v O
 的 ns O
 灾情 uj O
 和 n O
 救 c O
 灾 vn O
 工作 vn O
 ， w O
 指示 n O

Training corpus

Test corpus, last column is the prediction

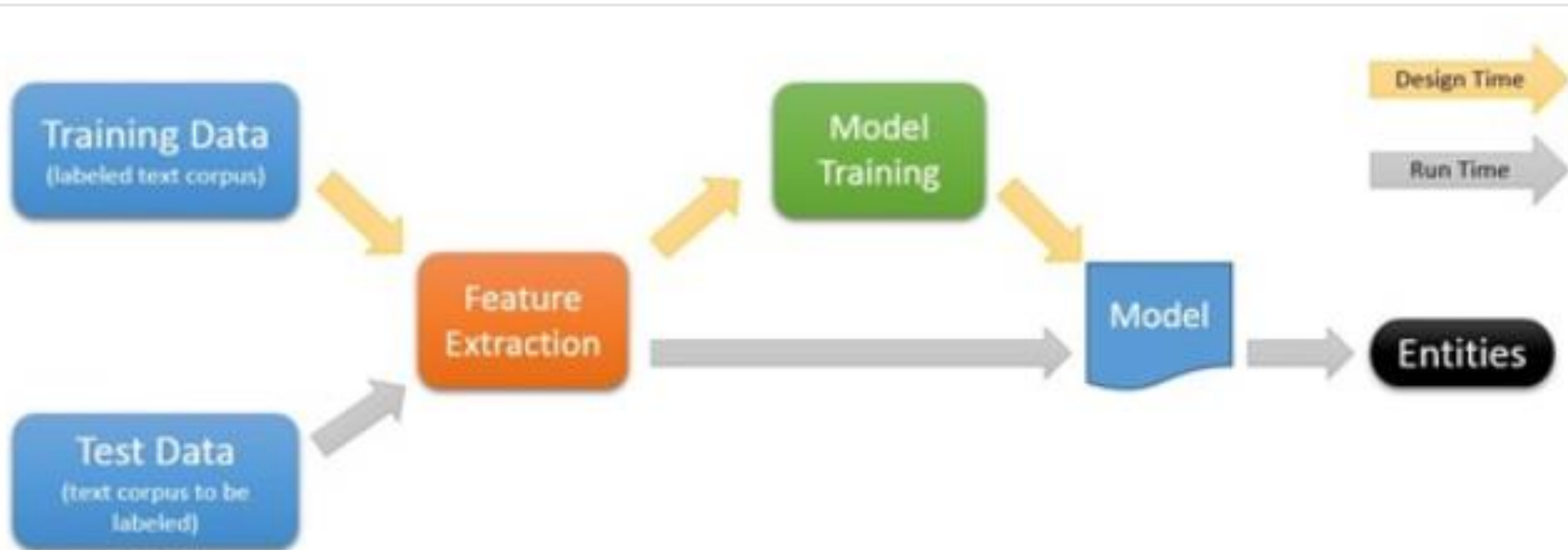
多	m	→	O	→	O
企业	n	→	O	→	O
校友	n	→	O	→	O
纷纷	d	→	O	→	O
上海	ns	→	B	→	B
交通	n	→	I	→	I
大学	n	→	I	→	I
明年	t	→	O	→	O
将	d	→	O	→	O
迎来	v	→	O	→	O
120周年	m	→	O	→	O
庆	vg	→	O	→	O
。	w	→	O	→	O
腾	v	→	B	→	O
讯	ng	→	I	→	O
公司	n	→	I	→	O
红海	ns	→	B	→	O
公司	n	→	I	→	O

Supervised Machine Learning and Knowledge **Bottleneck**

- *Requires considerable size of training corpus, hence facing serious knowledge bottleneck.*
- *cannot effectively support user-defined named entities which are important for open-domain IE*

Supervised Machine Learning Method (**summarization**)

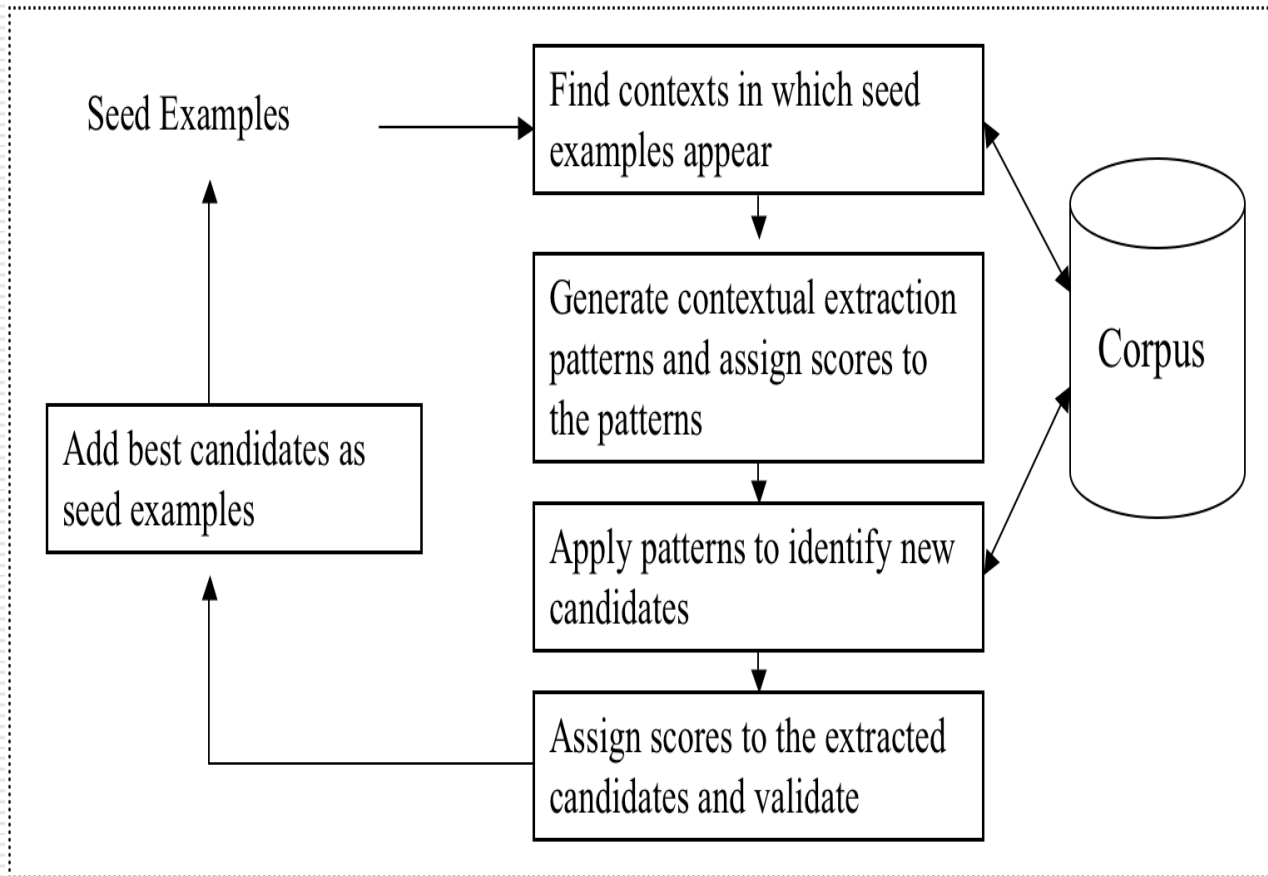
- ❑ Feature extraction: very important.
- ❑ Model selection:



Semi-supervised Method

- Training Corpus:
 - ✓ few seeds and large **un-annotated** corpus
- Training methods:
 - ✓ Bootstrapping
 - ✓ Others Expansion methods

Bootstrapping Algorithm



Bootstrapping refers to **a technology** that starts from a small initial effort and gradually grows into something larger and more significant.

For Example

- ❑ Seeds: 腾讯公司
- ❑ Find contexts in which seeds appear
 - ✓ 腾讯公司CEO马化腾说。。。
 - ✓ 7月21日, 腾讯宣布启动AI加速器
 - ✓ 腾讯宣布成立人工智能医学影像联合实验室
- ❑ Generate pattern based on the context
 - ✓ XX company CEO → XX is a company name
 - ✓ XX announced → XX is a company name
- ❑ Apply the pattern to find new one
- ❑ 虹华公司CEO在一个大会上... → Hong Hua is a company
- ❑ 华为宣布进军欧洲市场 → Huawei is a company
- ❑ 某公司CEO撤销了。。。 → Some is not a company

How to score the patterns?

In order to find **highly relevant** or **highly frequent patterns**:

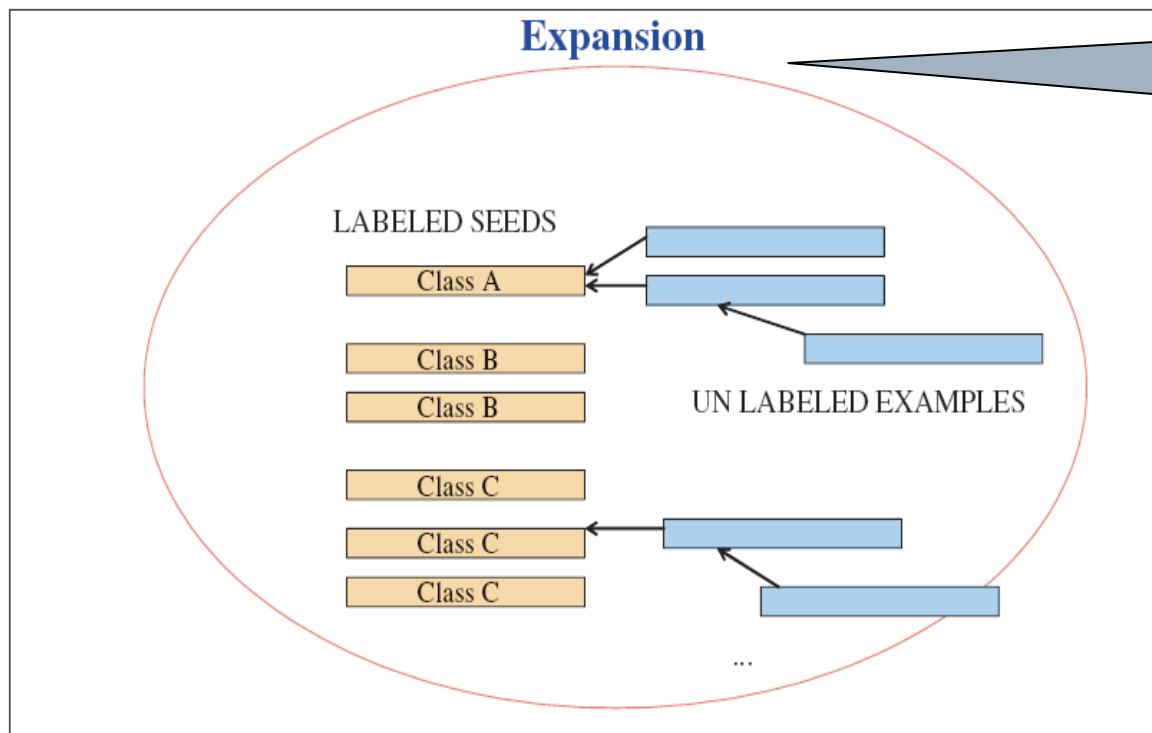
□ relevance rate: $R_i = F_i / N_i$

■ F_i : the number of instances of pattern i that were activated in the positive examples.

■ N_i : the total number of instances of pattern i activated in the **training corpus**

□ $\text{score}_i = R_i * \log (F_i)$

Other Learning Methods

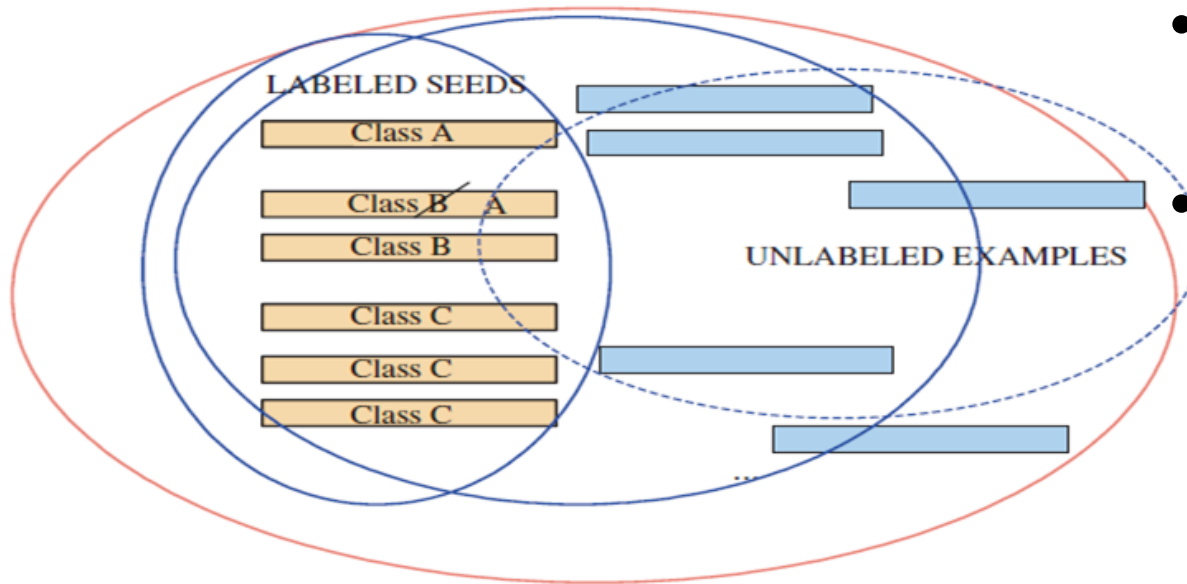


learn to classify unlabeled example to the closest seeds

For example:

- Self-learning
- Co-training
- Active learning
- ...

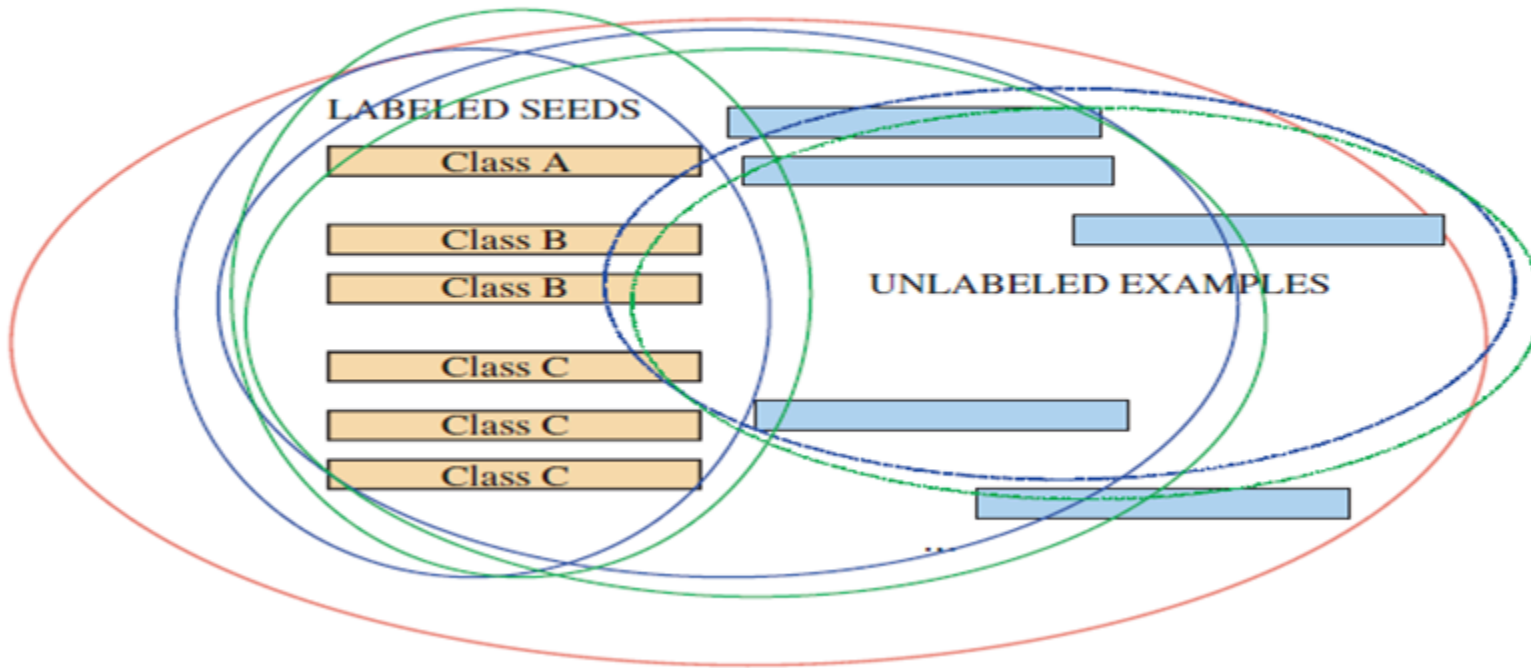
Self-training



- Incrementally
- supervised learning.
- Until reaches a certain level of accuracy

1. Labeled data → training → **model(1)**
2. Unlabeled data → model(1) → new labeled data
3. Labeled data + new labeled data → training → **model(2)**
4. Unlabeled data → model(2) → new labeled data
5. More data → training → **model(3)**
6. ...

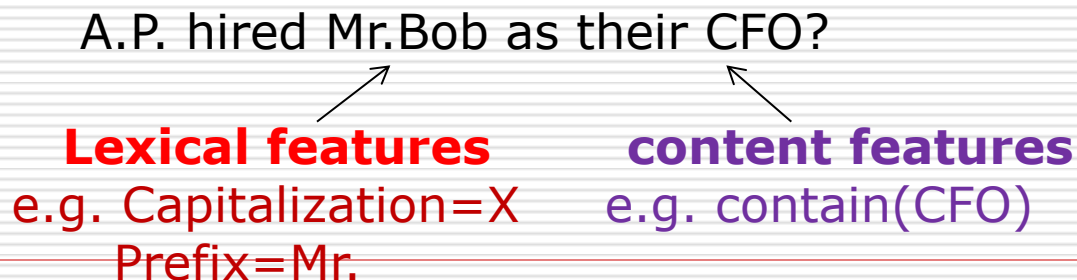
Co-training



1. Labeled data \rightarrow training \rightarrow **modelA (1) + modelB (1)**
2. Unlabeled data \rightarrow modelA & modelB \rightarrow labeled Data by modelA & modelB
3. Labeled data + new labeled data \rightarrow training \rightarrow **modelA(2), modelB(2)**
4. More unlabeled data \rightarrow modelA(2), modelB(2) \rightarrow labeled data by A,B
5. ...

Co-training (cont.)

- *two (or more) classifiers are trained using the same seed set of labeled examples, but each classifier trains with a disjoint subset of features.*
- These feature subsets are commonly referred to as **different views** that the classifiers have on the training examples.



A co-training algorithm

- features x can be separated into two types x_1, x_2
- either x_1 or x_2 is sufficient for classification – i.e. there exists functions f_1 and f_2 such that

$$f(x) = f_1(x_1) = f_2(x_2) \text{ has low error}$$

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x

Use L to train a classifier h_2 that considers only the x_2 portion of x

Allow h_1 to label p positive and n negative examples from U'

Allow h_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

Randomly choose $2p + 2n$ examples from U to replenish U'

Active Learning

- ❑ Human involved: all examples are labeled **by a human**
- ❑ Unlabeled data to be labeled: is **carefully selected** by the machine.

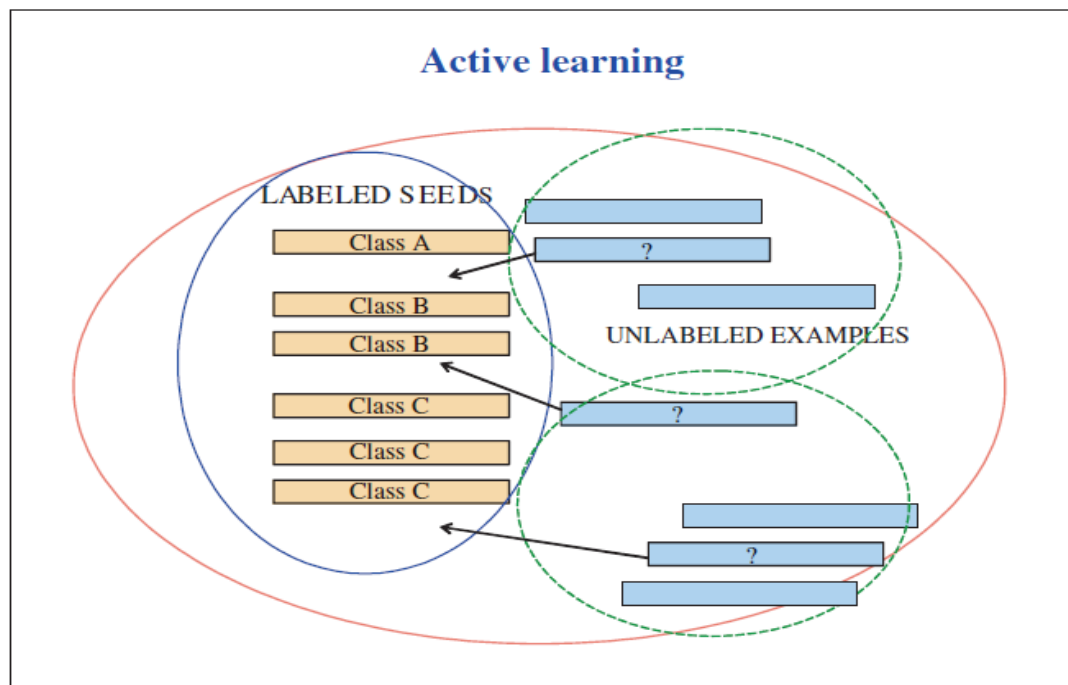


Fig. 6.5. Active learning: Representative and diverse examples to be labeled by humans are selected based on clustering.

Selective examples in active learning

- Most *uncertain* and most *informative*
- *Representative* or *diverse* with the other unlabeled examples

How to select those examples ?

- The probability , **Entropy**-based measure → *uncertain*
- Similarity calculation, clustering, outliers in the clusters → *representative*

Supervised vs. semi-supervised method

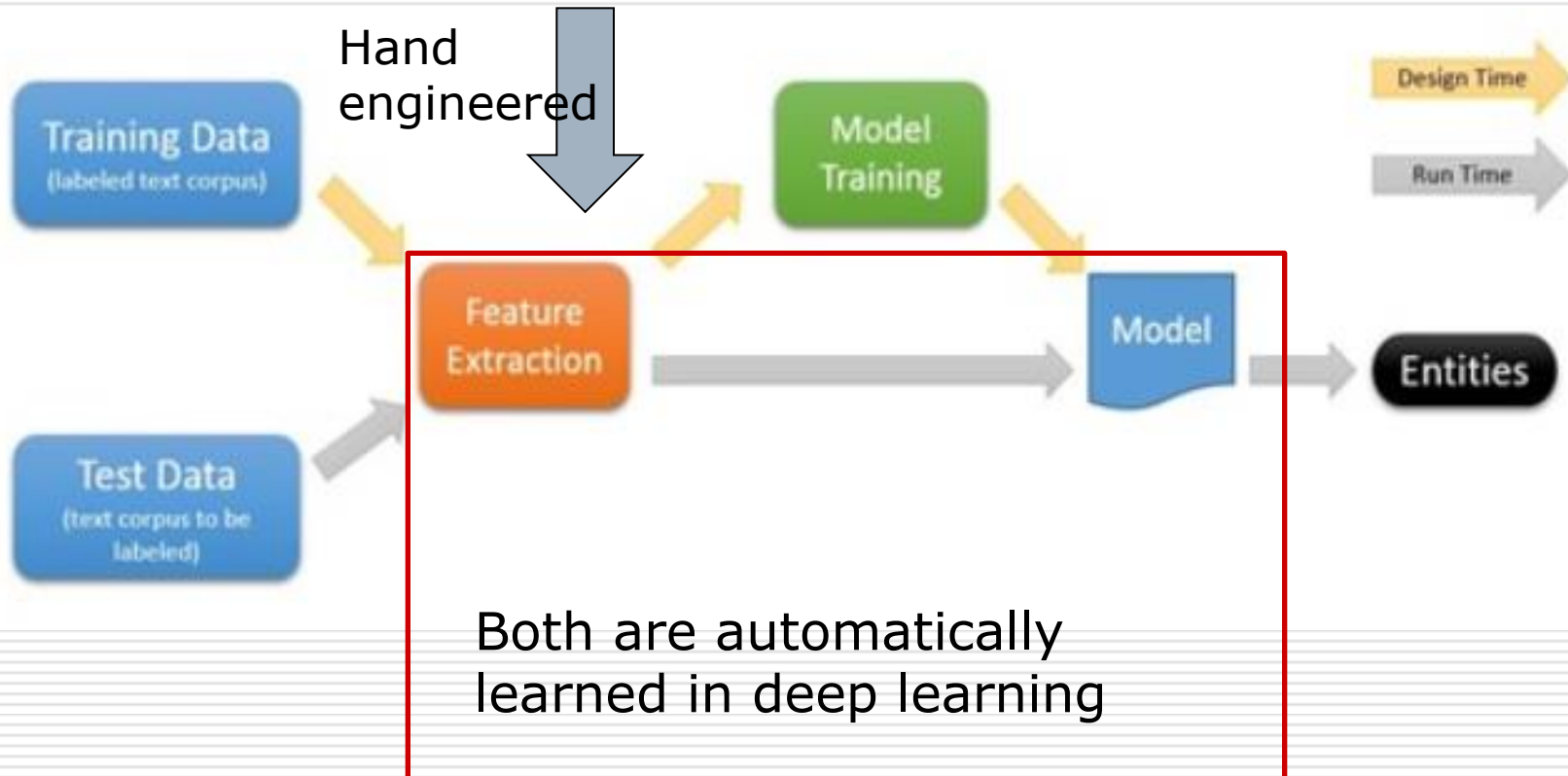
- **Chinese NP chunking:** Experiments with Supervised and semi-supervised learning 国立台湾大学 2008年的文章

	Tag accuracy	Precision	Recall	F-rate
封閉測試：测试语料和训练语料同种类型，70%训练，30%测试				
supervised	92.06%	84.65%	86.28%	85.46%
supervised II	91.76%	81.71%	86.05%	83.82%
semi-supervised	92.19%	84.85%	86.64%	85.73%
開放測試：测试语料与训练语料完全不同				
supervised	89.03%	67.31%	72.92%	70%
supervised II	83.83%	63.06%	69.23%	66%
semi-supervised	91.61%	76.47%	81.25%	78.79%

Unsupervised vs. Supervised NE Benchmarking (from Chen Niu)

	Supervised NE			Unsupervised NE			D
type	P	R	F	P	R	F	F
Person	92.3%	93.1%	92.7%	86.6%	88.9%	87.7%	5%
Location	89.0%	87.7%	88.3%	82.9%	81.7%	82.3%	6%
Org	85.7%	87.8%	86.7%	57.1%	48.9%	52.7%	34%

Deep Learning vs. Machine Learning



NE using DL

results of **Novel** and **Emerging** Entity
Recognition(WNUT2017)

Identify:

- Person
- Location
- Corporation
- Product
- Creative work (song,movie,book,...)
- Group (sports team, music band)

Deep learning model they used

- The multi-task neural network architecture learns higher order feature representations from word and character sequences along with basic Part-of-Speech tags and gazetteer information.
- Two modifications of a basic neural network architecture for sequence tagging.

Dataset statistics

Metric	Dev	Test
Documents	1,008	1,287
Tokens	15,734	23,394
Entities	835	1,040
person	470	414
location	74	139
corporation	34	70
product	114	127
creative-work	104	140
group	39	150

Results

Team	F1 (entity)	F1 (surface)
Arcada (Jansson and Liu, 2017)	39.98	37.77
Drexel-CCI (Williams and Santia, 2017)	26.30	25.26
FLYTXT (Sikdar and Gambäck, 2017)	38.35	36.31
MIC-CIS	37.06	34.25
SJTU-Adapt (Lin et al., 2017)	40.42	37.62
SpinningBytes (von Däniken and Cieliebak, 2017)	40.78	39.33
UH-RiTUAL (Aguilar et al., 2017)	41.86	40.24

F1(entity):

F1(surface): measures how good systems are at correctly recognizing a diverse range of entities, rather than just the very frequent surface forms. 识别正确的实体只计算一次。

Summarization

- *Rule based & machine learning method*
- *Supervised vs. semi-supervised methods*
- *Core learning engines are language independent*
- *Feature extraction relies on language specific properties*