Information Extraction

Le Anh Cuong

Reading

- Chapter 22 [1]

Outline

- Name entity recognition
- Relation detection and classification
- Temporal and event processing

IE tasks

- Named entity recognition
  - recognizes and classifies named expressions in text (such as person, company, location, protein names…)
- Noun phrase coreference resolution
  - phoric references (anaphoric and cataphoric)
- Cross-document coreference resolution

IE tasks

- Semantic role recognition
  - assignment of semantic roles to the (syntactic) contituents of a sentence
- Entity relation recognition
  - the relation between two or more entities is detected and the relation posibly is typed with a semantic role
- Timex recognition and resolution
  - temporal expression detection and recognition
    - absolute, relative, event anchored expressions

Named Entity Recognition

- What is NE?
- What isn't NE?
- Problems and solutions with NE task definitions
- Problems and solutions with NE task
- Some applications

Why do NE Recognition?

- Key part of Information Extraction system
- Robust handling of proper names essential for many applications
- Pre-processing for different classification levels
- Information filtering
- Information linking

NE Definition

- NE involves **identification** of *proper names* in texts, and **classification** into a set of predefined categories of interest.
- Three universally accepted categories: **person**, **location** and **organisation**
- Other common tasks: recognition of date/time expressions, measures (percent, money, weight etc), email addresses etc.
- Other domain-specific entities: names of drugs, medical conditions, names of ships, bibliographic references etc.

What NE is NOT

- NE is **not** event recognition.
- NE recognises **entities** in text, and classifies them in some way, but it does not create templates, nor does it perform co-reference or entity linking, though these processes are often implemented alongside NE as part of a larger IE system.
- NE is not just matching text strings with pre-defined lists of names. It only recognises entities which are being used as entities in a given context.
- NE is not easy!

Problems in NE Task Definition

- Category definitions are intuitively quite clear, but there are many grey areas.
- Many of these grey area are caused by **metonymy**.

Person vs. Artefact: "The **ham sandwich** wants his bill." vs "Bring me a **ham sandwich**."

Organisation vs. Location : "**England** won the World Cup" vs. "The World Cup took place in **England**".

Company vs. Artefact: "shares in **MTV**" vs. "watching **MTV**"

Location vs. Organisation: "she met him at **Heathrow**" vs. "the **Heathrow** authorities"

Solutions

- The task definition must be very clearly specified at the outset.
- The definitions adopted at the MUC conferences for each category listed guidelines, examples, counter-examples, and "logic" behind the intuition.
- MUC essentially adopted simplistic approach of disregarding metonymous uses of words, e.g. "England" was always identified as a location. However, this is not always useful for practical applications of NER (e.g. football domain).
- Idealistic solutions, on the other hand, are not always practical to implement, e.g. making distinctions based on world knowledge.

Basic Problems in NE

- Variation of NEs – e.g. John Smith, Mr Smith, John.
- Ambiguity of NE types
  - John Smith (company vs. person)
  - May (person vs. month)
  - Washington (person vs. location)
  - 1945 (date vs. time)
- Ambiguity with common words, e.g. “may”

More complex problems in NER

- Issues of style, structure, domain, genre etc.
  - Punctuation, spelling, spacing, formatting, ….all have an impact

Dept. of Computing and Maths
Manchester Metropolitan University
Manchester
United Kingdom

> Tell me more about Leonardo
> Da Vinci

List Lookup Approach

- System that recognises only entities stored in its lists (gazetteers).
- Advantages - Simple, fast, language independent, easy to retarget
- Disadvantages – collection and maintenance of lists, cannot deal with name variants, cannot resolve ambiguity

Shallow Parsing Approach

- Internal evidence – names often have internal structure. These components can be either stored or guessed.

location:

CapWord + {City, Forest, Center}

e.g. Sherwood Forest

Cap Word + {Street, Boulevard, Avenue, Crescent, Road}

e.g. *Portobello Street*

Shallow Parsing Approach

- External evidence - names are often used in very predictive local contexts

Location:

"to the" COMPASS "of" CapWord

e.g. *to the south of* ***Loitokitok***

"based in" CapWord

e.g. *based in* ***Loitokitok***

CapWord "is a" (ADJ)? GeoWord

e.g. ***Loitokitok*** *is a friendly city*

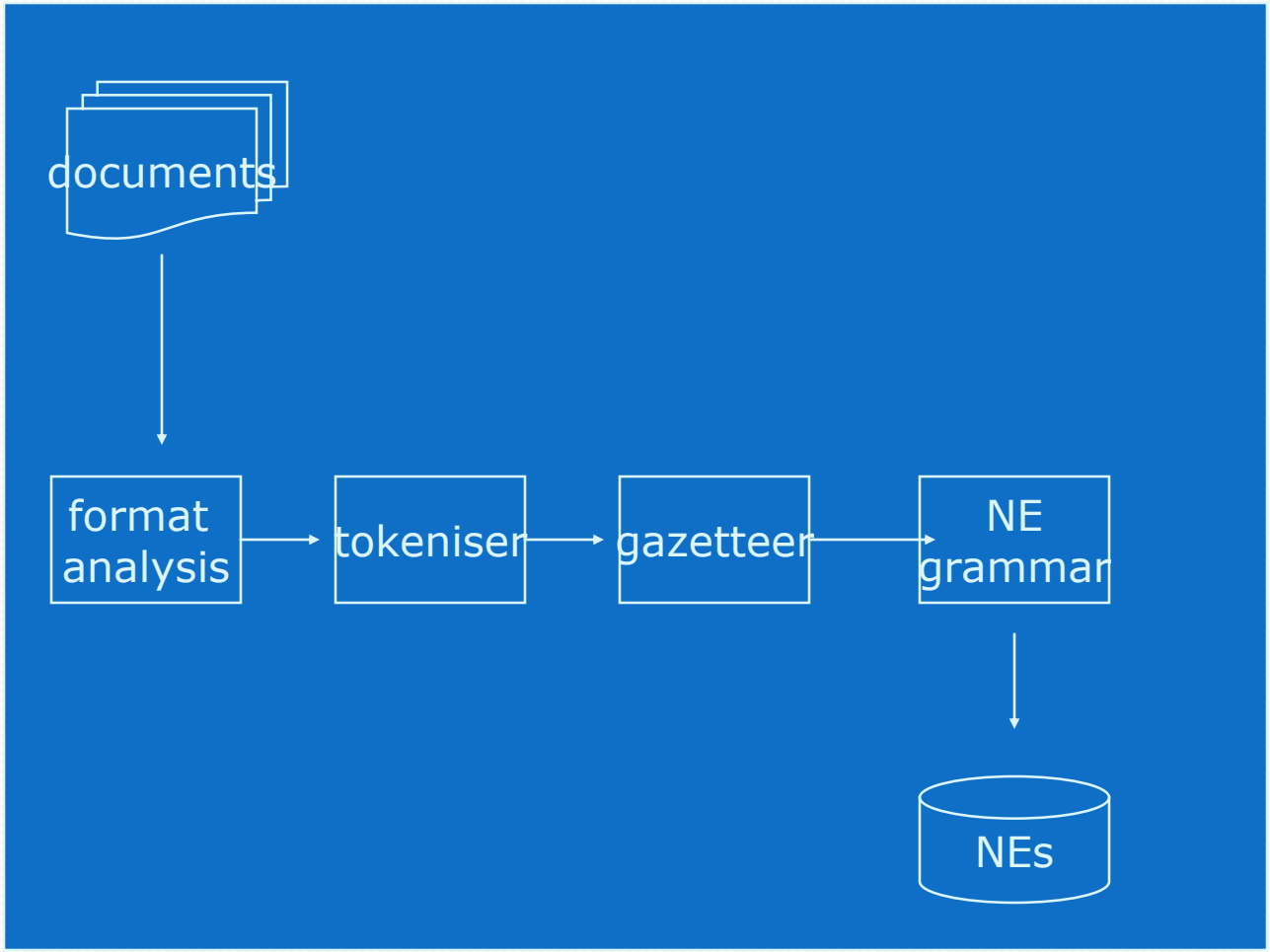
Difficulties in Shallow Parsing Approach

- **Ambiguously capitalised words** (first word in sentence)
  [All American Bank] vs. All [State Police]
- **Semantic ambiguity**
  "John F. Kennedy" = airport (location)
  "Philip Morris" = organisation
- **Structural ambiguity**
  [Cable and Wireless] vs. [Microsoft] and [Dell]
  [Center for Computational Linguistics] vs. message from [City Hospital] for
  [John Smith].

Technology

- JAPE (Java Annotations Pattern Engine)
- Based on Doug Appelt's CPSL
- Reimplementation of NE recogniser from LaSIE

NE System Architecture



Modules

- **Tokeniser**
  - **segments text into tokens, e.g. words, numbers, punctuation**
- **Gazetteer lists**
  - **NEs, e.g. towns, names, countries, …**
  - **key words, e.g. company designators, titles, …**
- **Grammar**
  - **hand-coded rules for NE recognition**

JAPE

- Set of phases consisting of pattern /action rules
- Phases run sequentially and constitute a cascade of FSTs over annotations
- LHS - annotation pattern containing regular expression operators
- RHS - annotation manipulation statements
- Annotations matched on LHS referred to on RHS using labels attached to pattern elements

Tokeniser

- Set of rules producing annotations
- LHS is regular expression matched on input
- RHS describes annotations to be added to AnnotationSet

(UPPERCASE _LETTER) (LOWERCASE_LETTER)* >

Token; orth = upperInitial; kind = word

Gazetteer

- Set of lists compiled into Finite State Machines
- Each list has attributes MajorType and MinorType (and optionally, Language)

city.lst: location: city

currency_prefix.lst: currency_unit: pre_amount

currency_unit.lst: currency_unit: post_amount

Named entity grammar

- hand-coded rules applied to annotations to identify NEs
- annotations from format analysis, tokeniser and gazetteer modules
- use of contextual information
- rule priority based on pattern length, rule status and rule ordering

Example of JAPE Grammar rule

```
Rule:    Location1
Priority: 25
( ( { Lookup.majorType == loc_key,
      Lookup.minorType == pre}
    { SpaceToken} )?
  { Lookup.majorType == location}
  ( {SpaceToken}
    { Lookup.majorType == loc_key,
      Lookup.minorType == post} ) ?
)
 : locName -->
      :locName.Location = { kind = "gazetteer", rule = Location1
    }
```

MUSE

- **MUlti-Source Entity recognition**
- **Named entity recognition from a variety of text types, domains and genres.**
- **2 years from Feb 2000 – 2002**
- **Sponsors: GCHQ**

Supervised Classification

- Generative classifiers
  - learns a model of the joint probability $p(\boldsymbol{x},y)$ and makes predictions by using Bayes rule to calculate $p(y|\boldsymbol{x})$ and then selects the most likely label y
  - Hidden Markov model
- Discriminative classifiers
  - models the posterior probability $p(\boldsymbol{x}|y)$ directly and selects the most likely label y
  - Maximum entropy model
  - Support Vector Machine
  - Conditional Random Fields

Supervised Classification

- Support Vector Machines
  - advantage is that they can cope with many (sometimes) noisy features without being doomed by the course of dimensionality
  - succesfully employed in:
    - NERC (Isozaki and Kazawa, 2002)
    - Noun phrase coreferent resolution (Isozaki and Hirao, 2003)
    - Semantic role recognition (Zhang and Lee, 2003; Mehay et al., 2005)
    - Entity relation recognition (Culotto and Sorensen, 2004)

Supervised Classification

- Maximum Entropy Models
  - the classifier allows to model dependencies between features
  - succesfully employed in:
    - NERC (Chieu and Hwee, 2002)
    - Coreference resolution (Kehler, 1997)
    - Semantic role recognition (Fleischman et al., 2003; Mehay et al., 2005)

Supervised Classification

- Hidden Markov Models
  - Popular technique to detect and classify a linear sequence of information in text
  - Disadvantage is the need for large amounts of training data
  - System for extraction of gene names and locations from scientific abstracts (Leek, 1997)
  - NERC (Biker et al., 1997)
  - McCallum et al. (1999) extracted document segments that occur in a fixed or partially fixed order (title, author, journal)
  - Ray and Craven (2001) – extraction of proteins, locations, genes and disorders and their relationships

Supervised Classification

- Conditional Random Fields
  - Statistical method based on undirected graphical models
  - This metod can be thought of a generalization of both the maximum entropy model and the hidden Markov model
  - Best current approaches to IE in empirical evaluations
  - CRFs have been implemented for:
    - NERC (McCallum and Li, 2003)
    - Timex recognition and normalization (Ahn et al., 2005)

Supervised Classification

- Decision Rules and Trees
  - Popular and succesfull technique in coreference resolution (McCarthy and Lehnert 1995; Soon et al., 2001; Ng and Cardie, 2002)
  - Other applications to information extraction from semi-structured text (Soderland, 1999)

Unsupervised Classification Aids

- Clustering
  - noun phrase coferefence resolution
  - cross-document coreference resolution
  - relation recognition

- Expansion (bootstrapping)
  - Yarkowski algorithm – word sense disambiguation
  - NERC – context matching
  - question type recognition
  - dictionary construction

Unsupervised Classification Aids

- Self-training
  - incrementally learning a classifier based on a seed set of labeled examples and a set of unlabeled examples that are labeled with the current classifier
  - noun phrase coferefence resolution
- Co-training
  - multiple classifiers are trained using the same seed set of labeled examples, but each with a disjoint subset of features (conditionally independent)
  - NERC
  - noun phrase coferefence resolution

Unsupervised Classification Aids

- Active Learning
  - the algorithm starts with a seed set of labeled examples
  - at each iteration, a set of examples is selected and labeled by human and added to the training set
  - the selected examples are the ones the current classifier considers as most uncertain and thus most informative
  - NERC

Integration of Information Extraction in Retrieval Models

- Question answering systems
  - most of current QA systems are restricted to answering factual question
- Query by example
  - commonly used in multimedia information retrieval
- XML retrieval model
  - documents carry additional information (metadata)

Integration of Information Extraction in Retrieval Models

- IE offers the oportunity to semantically enrich the indexing representations made of the documents
- A same concept can be expresed using different syntactic structures
  - current research tries to solve this problem by identifying paraphrases (i.e., finding similar content that is expressed differently)
- Very little empirical studies on actual improvement of IR research by adding semantic information

Integration of Information Extraction in Retrieval Models

- Integrating the sematic annotations into the typical *bag-of-words*
  - o to *k* labels can be attached to single terms or phrases, combination of terms, passages…
  - *bed-of-words covered with different semantics*
  - query can be translated to a similar layered format as a document

Integration of Information Extraction in Retrieval Models

- Retrieval models:
  - Vector space model
  - Language model
    - documents represent a certain distribution of information content
    - ranking by the probability that the query is generated given a document
  - Inference network model
    - directed, acyclic dependency graph
      - nodes represent propositional (binary) variables or constants
      - edges represent dependence relations between propositions

Case studies

- Domains:
  - Information extraction from news texts
  - Biomedical domain
  - Intelligence gathering
  - Economic and business
  - Legal domain
  - Informal texts (transcripts, blogs, mails)

Information extraction from news texts

- Message Understanding Conferences (MUC)
  - Each conference operated in a specific domain
  - Finding relations between different entities that form the constituents of an event and that fill a template frame (e.g. time, location, instrument and actors in a terrorist attack)
- Automatic Content Extraction (ACE)
  - National Institute of Standards and Technology (NIST)
  - Automatic processing of human language in text form

IE from news texts (performance)

- NERC (persons, organizations, locations)
  - F1 > 95%
  - Methods:
    - Maximum entropy model, hidden Markov model…
  - Human performance: 94-96%

- Noun phrase coreference resolution
  - Decision tree algorithms:
    - F1 – 66.3% (MUC-6)
    - F1 – 61.2% (MUC-7)
  - C4.5:
    - F1 ~ 60%

IE from news texts (performance)

- Cross-document noun phrase coreference res.
  - detection of synonymous (alias) names
  - disambiguation of polysemious names
  - Li et al. (2004) – New York Times 300 documents
    - F1 – 73% (synonymous names)
    - F1 – 91% (polysemious names)
- Entity relation recognition
  - Culotta and Sorsen (2004) – ACE corpus 800 doc.
    - SVM using different dependency tree kernels
    - F1 – 61-63%
    - Precision ~ 80%, Recall ~ 50%

IE from Biomedical Texts

- Large amount of biological and medical literature
  - patient reports
  - scientific literature (MEDLINE – 15 mil. abstracts)
- NERC
  - very common task
  - complex task in this domain
  - problem with boundary detection

IE from Biomedical Texts (performance)

- NERC
  - Zhang et al (2004) – 22 categories (GENIA ontology)
    - hidden Markov model: F1 – 66.5% (range 80 - 0%)
  - Kou et al (2005) – protein recognition GENIA corpus
    - maximum entropy model: F1 – 66%
    - conditional random fields: F1 – 71%
- Entity relation recognition
  - Leroy et al (2003) – cascaded finite state automata
    - on 26 abstracts: 90% precision

Intelligence Gathering

- Evidence Extraction and Link Discovery (DARPA)
  - system for scanning large amounts of heterogeneus, multi-lingual, open-source texts
  - detecting crime patterns, gangs and their organizational structure, suspect activity

- NERC
  - Chau and Xu (2005) – 36 Phoenix Police reports
    - persons: precision – 74%, recall – 73%
    - addresses: precision – 60%, recall – 51%
    - narcotic drugs: precision – 85%, recall – 78%
    - personal property: precision – 47%, recall – 48%

IE from Legal Texts

- Often combination of structured and unstructured data
- Recognition and tracking of named entities (mainly persons)
- Classification of sentences of court decisions according to their rethorical role
- Low interest in using IE in the law domain
  - Language problems:
    - vocabulary, syntax, semantics
    - disambiguation
    - POS tagging and parsing more dificult than in other texts

The Future of Information Extraction in a Retrieval Context

- Content recognition in multimedia
- Cascaded model – the output of one type of extraction forms the features of a more complex task of extraction
- Queries in spoken format

The Future of Information Extraction in a Retrieval Context

- Information synthesis (information fusion)
  - Selection of information
    - satisficing
    - suppresion
    - by veto
  - Integration of information
    - additive or subadditive accumulation
      - sum of information equal or smaller
    - cooperative integration
      - sum of information larger (new information not explicitly present in the sources)
    - disambiguation of one source with the other
      - often a first step before accumulation and coordination

Name Entity Recognition System

Name Entity Recognition:

- Identifying certain phrases/word sequences in a free text.
- Generally it involves assigning labels to noun phrases.
- Lets say: person, organization, locations, times, quantities, miscellaneous, etc.
- NER useful for information extraction, clever searching etc.

Name Entity Recognition System

Example: Bill Gates **(person name)** opened the gate **(thing)** of cinema hall and sat on a front seat to watch a movie named the Gate **(movie name)**.

- Simply retrieving any document containing the word Gates will not always help.
- It might confuse with other use of word gate.
- The good use of NER is to describe a model that could distinguish between these two items.

NER as sequence prediction

The basic NER task can be defined as:

- Let t_1, t_2, t_3,..... t_n be a sequence of entity types denoted by T.
- Let w_1, w_2, w_3,..... w_n be a sequence of words denoted by W.
- Given some W, find the best T.

Shared Data of CoNLL-2003

- Official web address

  http://cnts.uia.ac.be/conll2003/ner/
- Basically four different name entities:
  - Persons (I-Per)
  - Locations (I-Loc)
  - Organizations (I-Org)
  - Miscellaneous (I-Misc)

Data Format

- Data files contain four columns
- Separated by a single space
- First column is for words
- Second column is for Part of speech taggers
- Third column is for chunk tags
- Fourth column is for name entity tags
- Chunk tags and name entity tags are further classified as ***I-Type*** which means that a word is inside a phrase of type
- If two phrases are adjacent then ***B-Type*** is used distinguish two phrases by placing ***B-Type*** in front of first word of second phrase

Data Format

Word	POS Tag	Chunk Tag	Name Entity Tag
U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC

Encoding

- Suppose a random variable ***X*** can take ***y*** values.
- Each value can be described in ***log(y)*** bits.
- Log in base two.
  - Eight sided dice.
  - Eight possibilities, $2^3 = 8$.
  - $\log(8) = 3$ ($\log_b x = y$ $\quad$ $b^y = x$)

Entropy

- Entropy measures the amount of information in a random variable:

$$H(X) = -\sum P(X=x) \log(P(X=x))$$

- Entropy is the expected length of each out code.
- Entropy can be used as an evaluation metric.
- **In your assignment:** Random variable is name entity tag. And outcode is the probability of being different values of that tag.

$$H(NET) = -\{(P(per)*\log(P(per))) + (P(loc)*\log(P(loc))) + \ldots..\}$$

Probability P(x)	log(p(x)) = L	- (P(x)*L)
0	#NUM!	
0.1	-1	0.1
0.2	-0.6990	0.1398
0.3	-0.5229	0.1569
0.4	-0.3979	0.1592
0.5	-0.3010	0.1505
0.6	-0.2218	0.1331
0.7	-0.1549	0.1084
0.8	-0.0969	0.0775
0.9	-0.0458	0.0412
1	0	0.0000

Maximum Entropy

- Rough idea or generally:
  - Assume data is fully observed ***(R)***
  - Assume training data only partially determines the model ***(M)***
  - For undetermined issues, assume maximum ignorance
  - Pick a model M^* that minimises the distance between ***(R)*** and ***(M)***

$$M^* = \mathbf{argmin}_M D(R||M)$$

Kullback-Leibler Divergence

- The KL divergence measures the difference between two models.

$$D(R \,||\, M) = \sum_{x} P(x) \log\left(\frac{R(x)}{M(x)}\right)$$

- When R = M, D(R||M) = 0
- The KL divergence is used in maximum entropy.

Maximum Entropy Model

- Maximum Entropy Model (ME or Maxent) also known as Log-linear, Gibbs, Exponential, and Multinomial logit model used for machine learning.
- Based on Probability estimation technique.
- Widely used for classification problem like text-segmentation, sentence boundary detection, POS tagging, prepositional phrase attachment, ambiguity resolution, stochastic attributed-value grammar, and language modelling problems.

Maximum Entropy Model

Simple parametric equation of maximum entropy model:

$$P(c \mid s) = \frac{1}{Z(s)} \exp\left(\sum_i \lambda_i f_i(c, s) \right)$$

$$Z(s) = \sum_c \exp\left(\sum_i \lambda_i f_i(c, s) \right)$$

Here, ***c*** is the class from the set of labels ***C***. ***{I-Per, I-Org, …}***

s is a sample that we are interested in labelling. ***{word1, word2,…}***

λ is a parameter to be estimated and ***Z(s)*** is simply a normalising factor

Training Methods for Maxent

- There are many training methods. Complex mathematics and details can be found in literature
  - GIS (Generalised Iterative Scaling)
  - IIS (Improved Iterative Scaling)
  - Steepest Ascent
  - Conjugate Gradient
  - ……

Training Features

- Training data is used in terms of set of features.
- Deciding and extracting useful features is a major task in machine learning problem.
- Each feature is describing a characteristic of the data.
- For each feature, we measure its expected value using training data and set it as a constrain for the model.

Proposed Training Features for NER

- Current, previous and next Part of Speech Tags
- Current, previous, and next Chunk Tags
- Words start with capital letter
- Previous and next words start with capital letter
- Current, previous, and next word
- On adding each feature calculate performance
- ***Justify the purpose of adding each feature***

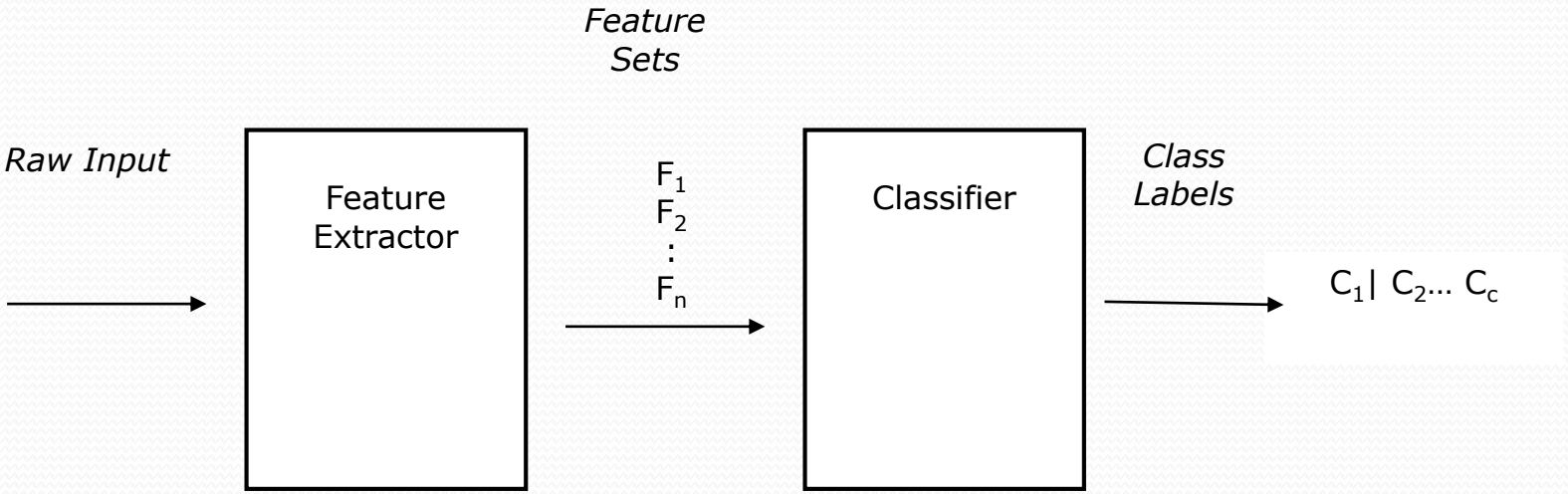
Feature Sets of training samples for Lee's maxent tool kit

Name Entity Tag	POS Tag	Chunk Tag	Word Start with Capital letter
I-ORG	NNP	I-NP	yes
O	NN	I-NP	no
I-PER	NNP	I-NP	yes
O	VBZ	I-VP	no
O	IN	I-PP	no
I-LOC	NNP	I-NP	Yes

Feature Sets of testing samples for Lee's maxent tool kit

POS Tag	Chunk Tag	Word Start with Capital letter	Name Entity Tag
NNP	I-NP	yes	I-ORG
NN	I-NP	no	O
NNP	I-NP	yes	I-PER
VBZ	I-VP	no	O
IN	I-PP	for	O
NNP	I-NP	yes	I-LOC

High Level Architecture of NER System



Steps to build a NER System

Step 1: You might require to do pre-processing of the data (e.g. eliminating empty lines, digits, punctuations)

Step 2: Extract features and format the training and testing samples required by Le's maxent toolkit.

Make a performance graph of NER system i.e. *F-Score vs number of samples*

Step 3: Pick **more** fixed percentage of samples (remember: each sample is in terms of feature set), lets say 5% of total samples.

Step 4: Train the maxent model using Le's maxent toolkit

```
maxent training_labelled_samples –m model1 –i 200
```

Steps to build a NER System

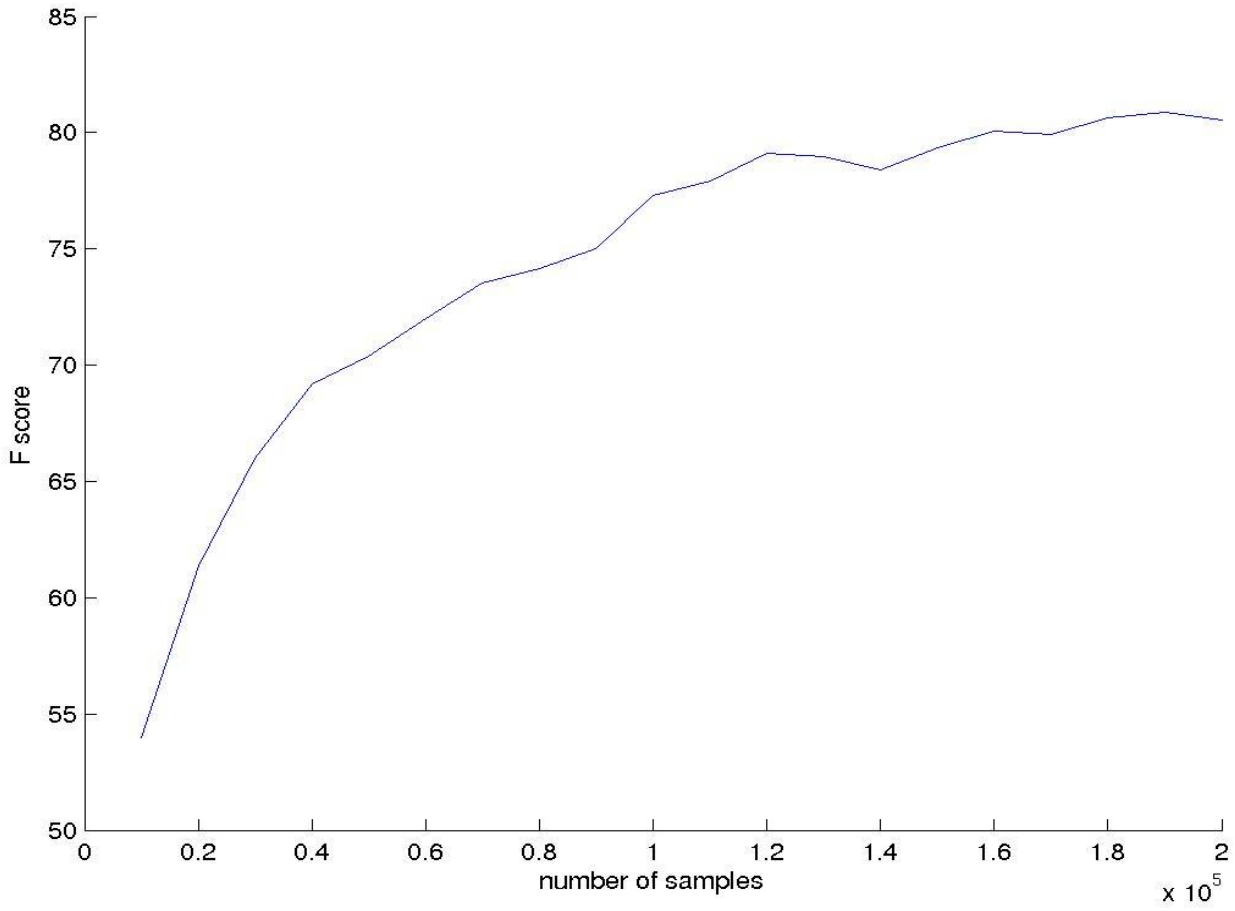
Step 5: Test the model (say model1 previously trained) using command

```
maxent –p –m model1 -o results.txt testing.unlabelled
```

Step 6: Calculate ***F-score*** using formula already discussed and make a graph labelling ***F-score*** along y-axis and ***number of samples*** along x-axis

Step 7: Reiterate from step 3

Example


85
80
75
70
65
60
55
50
F score
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
number of samples
x 10^5

Active Learning Method

- The goal of active learning method is to learn and improve the performance of the system from its experience.
- Error rate of a system can be reduced by minimising biasness of the model.
- The noise level can be decreased by selecting appropriate examples for training.

Active Learning Method

- Formally active learning method can be defined as:
  - Let $S=\{s_1, s_2, s_3, \ldots\}$ be a sample set
  - with labels $L=\{l_1, l_2, l_3, \ldots\}$
  - This sample set is to be extended by adding new labelled example after gaining some information from the previous experience.

Uncertainty Sampling Technique

- Uncertainty sampling technique measures the uncertainty of a model over a sample set.
- High uncertain means about which the learner is most uncertain.
- High uncertain examples will be more informative for the model and more likely to add into the sample set.
- Uncertainty can be estimated through Entropy.

Steps to Implement Active Learning Method

- Same steps as we followed to build sequential sampling NER system
- The only difference occurs when new samples will be added in the training data.
- The important issue is to decided which samples are select.
- Use entropy to calculate the amount information.
- Pick 5% more examples with highest entropy

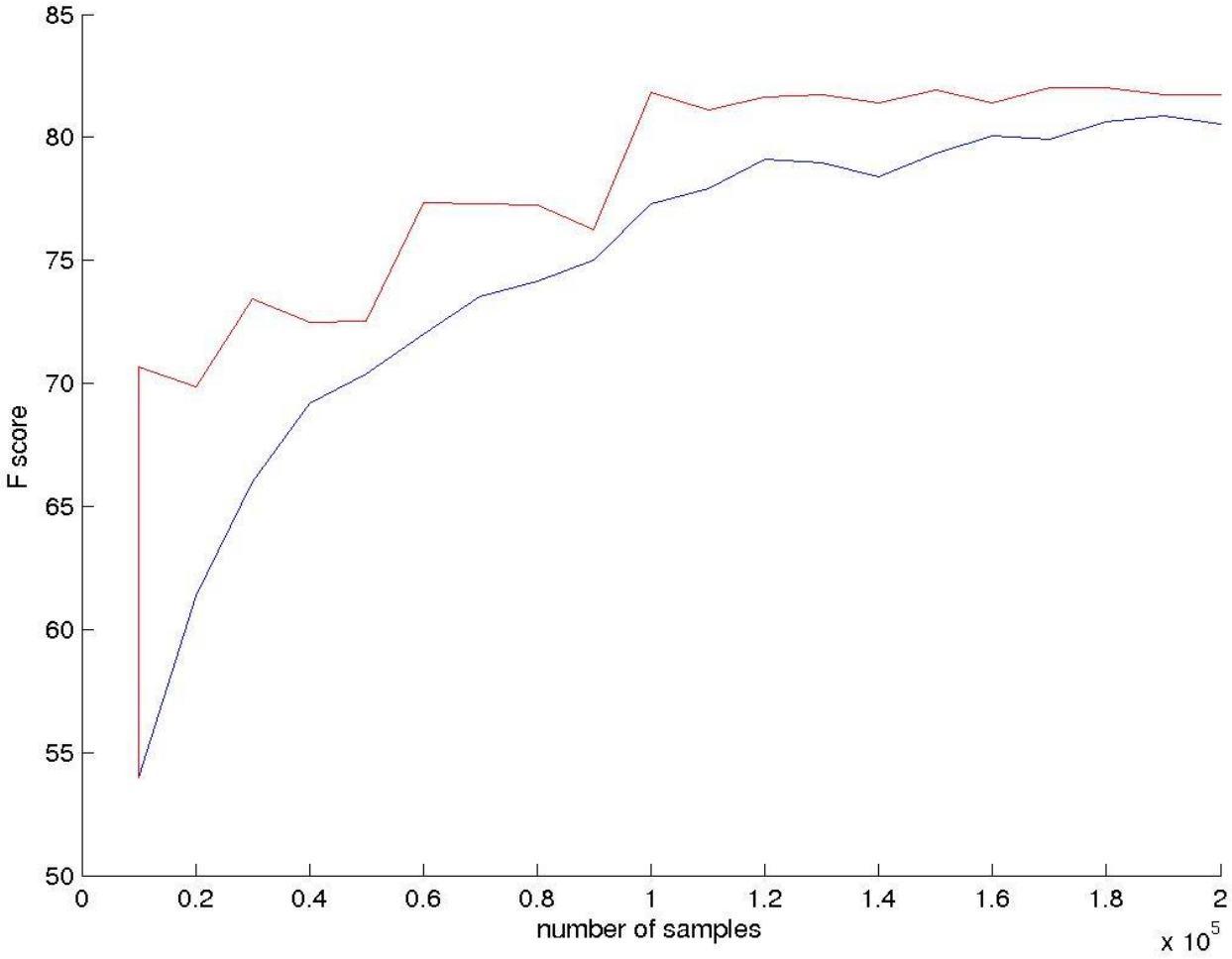
Steps to Implement Active Learning Method

- Divide a training pool into two categories.
  - Labeled_training_samples
  - Unlabeled_training_samples
- Pick few initial samples from labeled training data and train the model.
- Test the model on unlabeled_training_samples using below maxent command and calculate the entropy.

```
maxent -p -m model1 -detail -o results.txt testing.unlab
```

- On next iteration, pick 5% more samples with highest entropy and append it with the previous training samples.
- Also, on each iteration, test the model on testing data and make the performance graph.

Example of Active Learning Method



Named Entity Recognition

Germany's representative to the European Union's veterinary committee Werner Zwingman said on Wednesday consumers should ...

IL-2 gene expression and NF-kappa B activation through CD28 requires reactive oxygen production by 5-lipoxygenase.

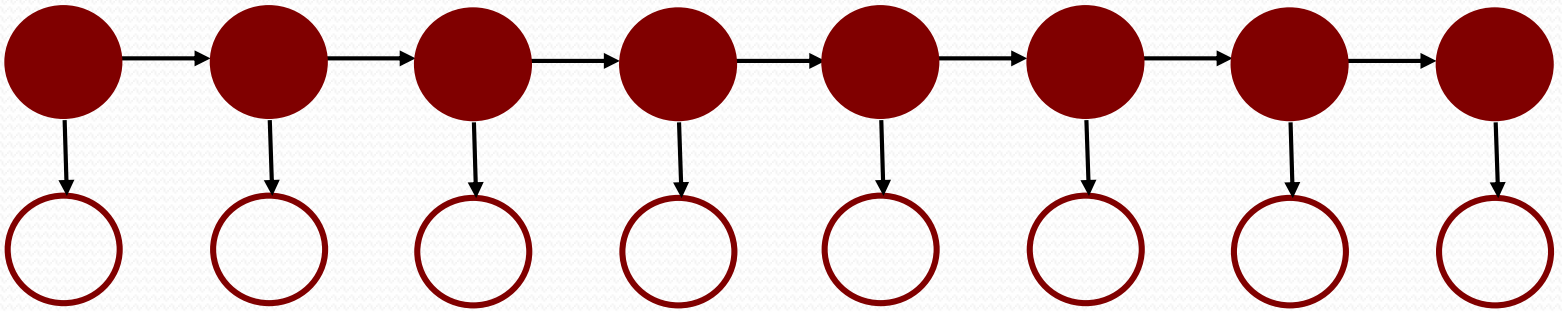
Why NER?

- Question Answering
- Textual Entailment
- Coreference Resolution
- Computational Semantics
- …

NER Data/Bake-Offs

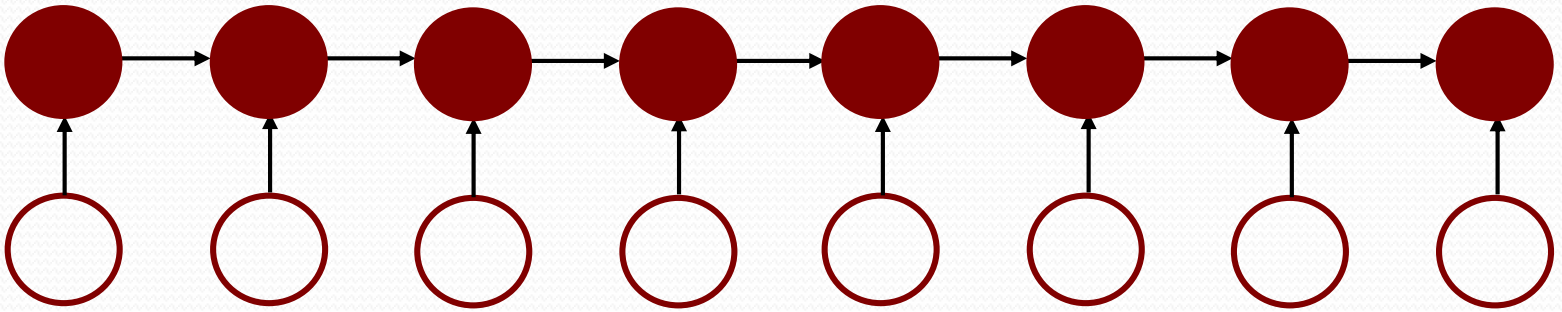
- CoNLL-2002 and CoNLL-2003 (British newswire)
  - Multiple languages: Spanish, Dutch, English, German
  - 4 entities: Person, Location, Organization, Misc
- MUC-6 and MUC-7 (American newswire)
  - 7 entities: Person, Location, Organization, Time, Date, Percent, Money
- ACE
  - 5 entities: Location, Organization, Person, FAC, GPE
- BBN (Penn Treebank)
  - 22 entities: Animal, Cardinal, Date, Disease, …

Hidden Markov Models (HMMs)



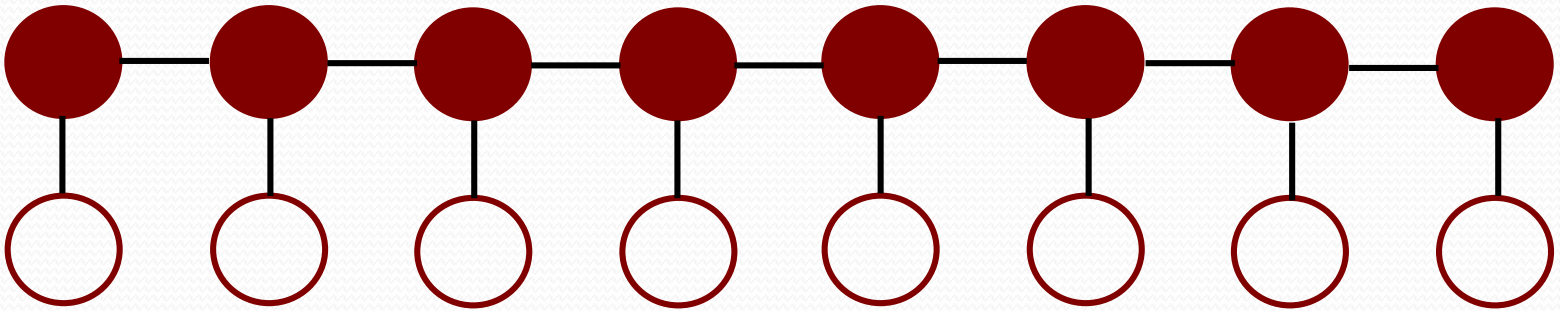
- Generative
  - Find parameters to maximize P(X,Y)
- Assumes features are independent
- When labeling X_i future observations are taken into account (forward-backward)

MaxEnt Markov Models (MEMMs)



- Discriminative
  - Find parameters to maximize P(Y|X)
- No longer assume that features are independent
- Do not take future observations into account (no forward-backward)

Conditional Random Fields (CRFs)



- Discriminative
- Doesn't assume that features are independent
- When labeling Y_i future observations are taken into account

➔ The best of both worlds!

Model Trade-offs

	Speed	Discrim vs. Generative	Normalization
HMM	very fast	generative	local
MEMM	mid-range	discriminative	local
CRF	kinda slow	discriminative	global

Stanford NER

- CRF
- Features are more important than model
- How to train a new model

Our Features

- Word features: current word, previous word, next word, all words within a window
- Orthographic features:
  - Jenny Xxxx
  - IL-2 XX-#
- Prefixes and Suffixes:
  - Jenny <J, <Je, <Jen, …, nny>, ny>, y>
- Label sequences
- Lots of feature conjunctions

Distributional Similarity Features

- Large, unannotated corpus
- Each word will appear in contexts - induce a distribution over contexts
- Cluster words based on how similar their distributions are
- Use cluster IDs as features
- Great way to combat sparsity
- We used Alexander Clark's distributional similarity code (easy to use, works great!)
- 200 clusters, used 100 million words from English gigaword corpus

Training New Models

Reading data:

- edu.stanford.nlp.sequences.DocumentReaderAndWriter
  - Interface for specifying input/output format
- edu.stanford.nlp.sequences.ColumnDocumentReaderAndWriter:

```
Germany         LOCATION
's              O
representative  O
to              O
The             ORGANIZATION
European        ORGANIZATION
Union
```

Training New Models

- Creating features
  - edu.stanford.nlp.sequences.FeatureFactory
    - Interface for extracting features from data
    - Makes sense if doing something very different (e.g., Chinese NER)
  - edu.stanford.nlp.sequences.NERFeatureFactory
    - Easiest option: just add new features here
    - Lots of built in stuff: computes orthographic features on-the-fly
- Specifying features
  - edu.stanford.nlp.sequences.SeqClassifierFlags
    - Stores global flags
    - Initialized from Properties file

Training New Models

- Other useful stuff
  - useObservedSequencesOnly
    - Speeds up training/testing
    - Makes sense in some applications, but not all
  - window
    - How many previous tags do you want to be able to condition on?
  - feature pruning
    - Remove rare features
  - Optimizer: LBFGS

Distributed Models

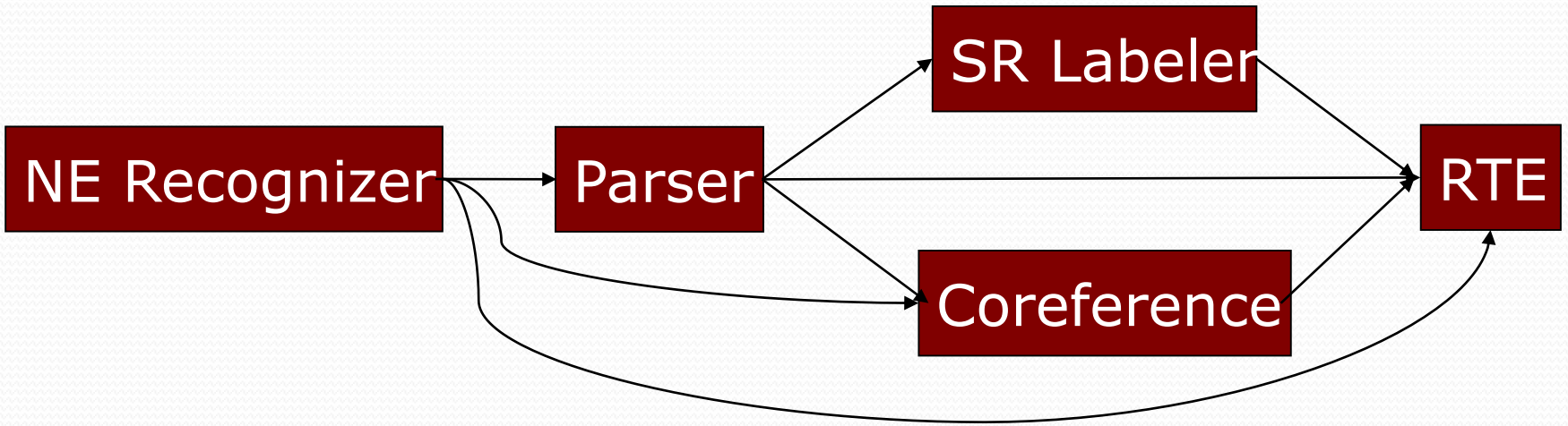
- Trained on CoNLL, MUC and ACE
- Entities: Person, Location, Organization
- Trained on both British and American newswire, so robust across both domains
- Models with and without the distributional similarity features

Incorporating NER into Systems

- NER is a component technology
- Common approach:
  - Label data
  - Pipe output to next stage
- Better approach:
  - Sample output at each stage
  - Pipe sampled output to next stage
  - Repeat several times
  - Vote for final output
- Sampling NER outputs is fast

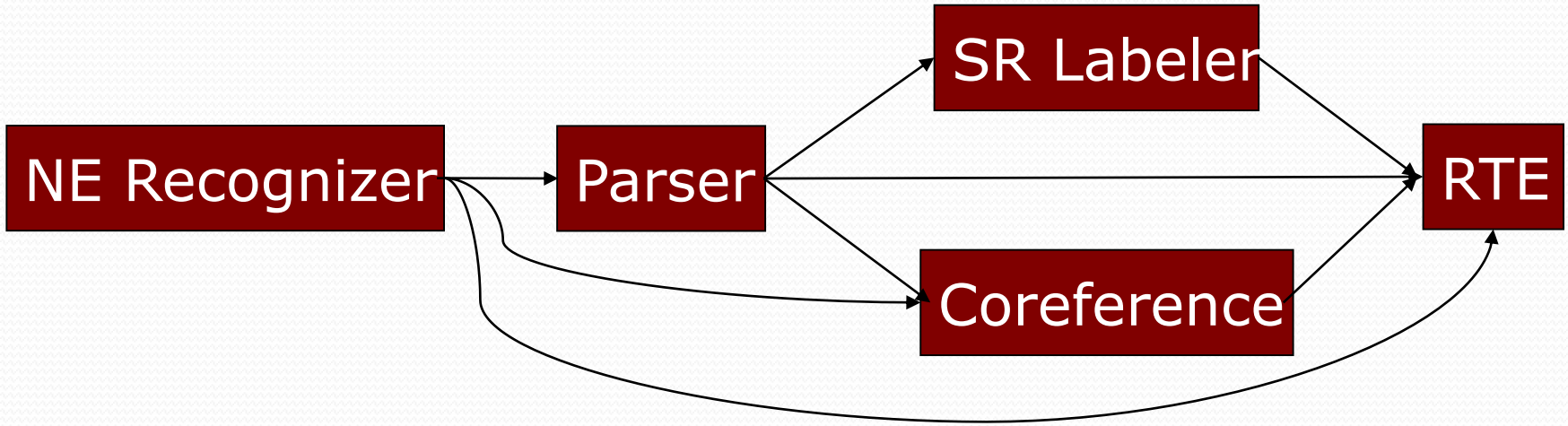
Textual Entailment Pipeline

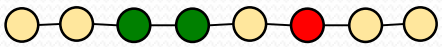
- Topological sort of annotators

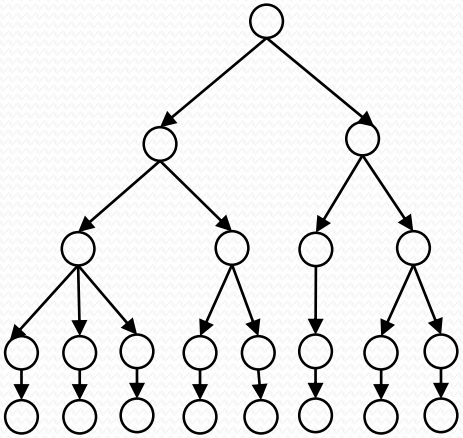


<NER, Parser, SRL, Coreference, RTE>

Sampling Example

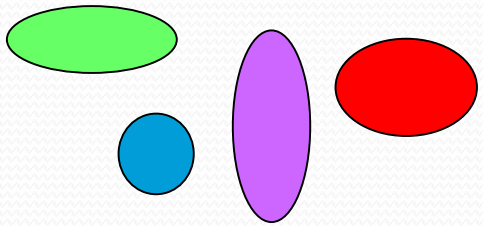




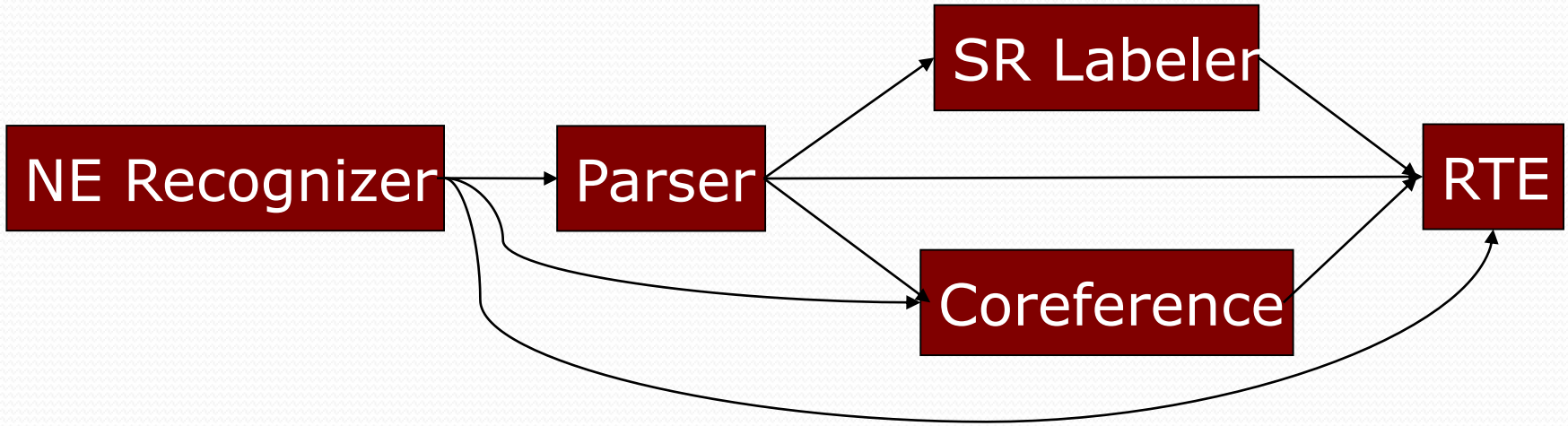


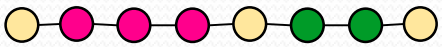
ARG0[…]
ARG1[…]
ARG-TMP[…]

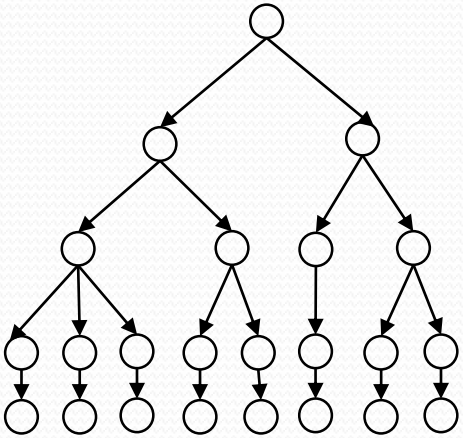
Yes



Sampling Example



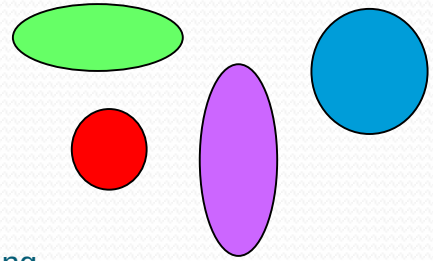




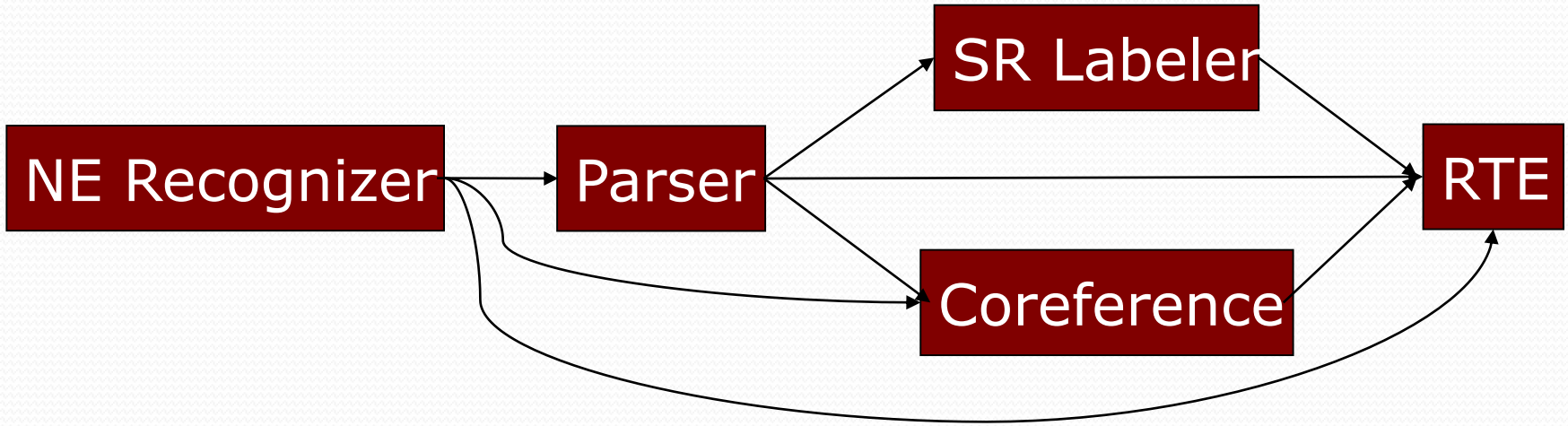
ARG0[…]
ARG1[…]
ARG-LOC[…]

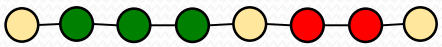
Yes

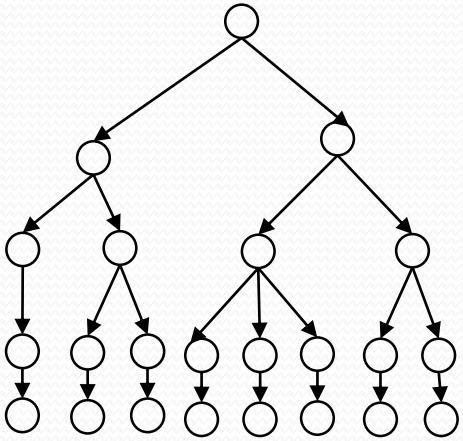
No



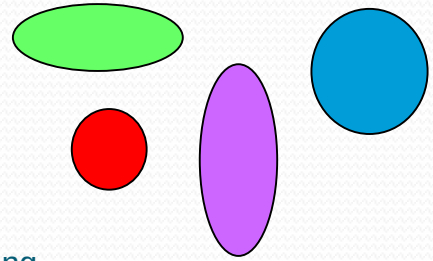
Sampling Example







ARG0[…]
ARG1[…]
ARG-TMP[…]

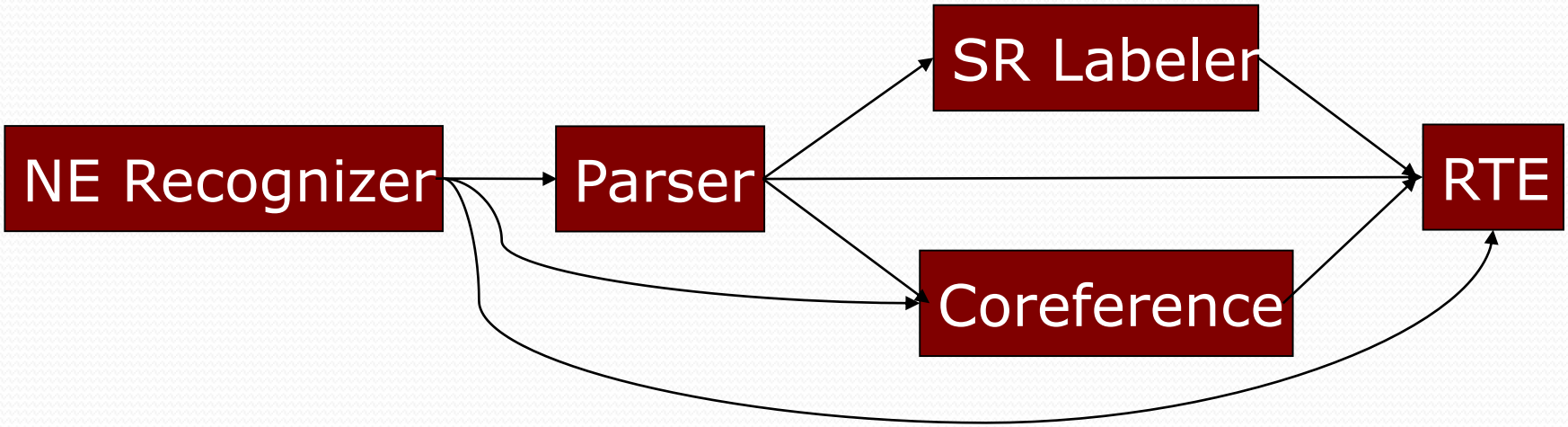


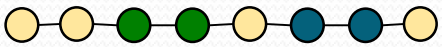
Yes

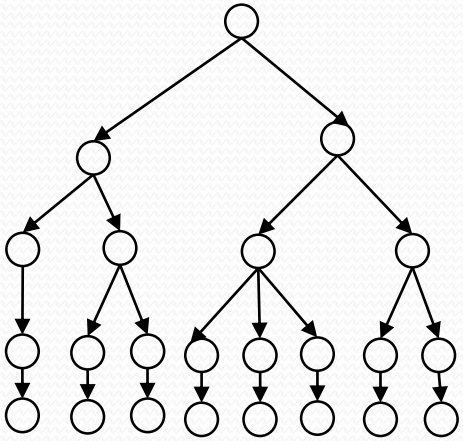
No

Yes

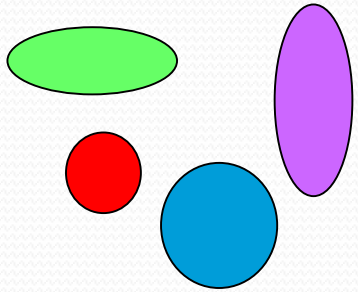
Sampling Example





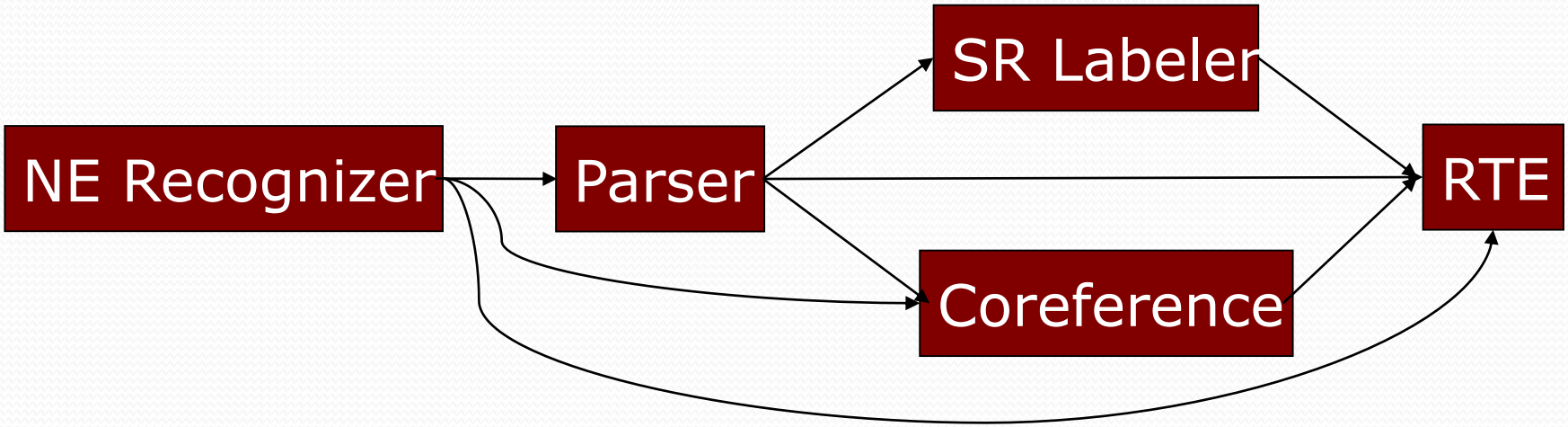


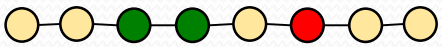
ARG0[…]
ARG1[…]
ARG2[…]

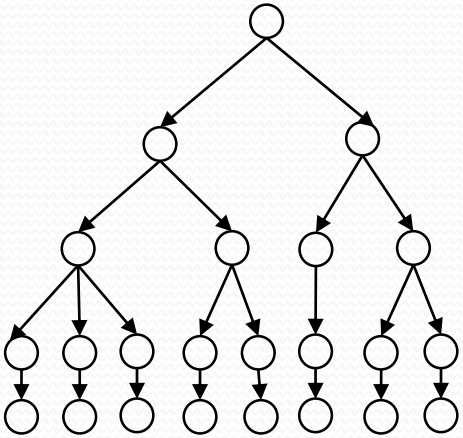


Yes
No
Yes
Yes

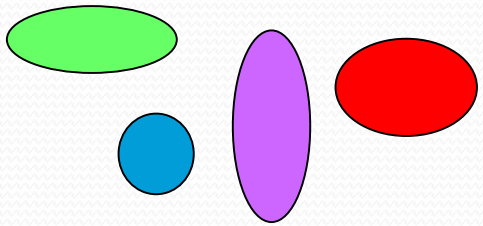
Sampling Example





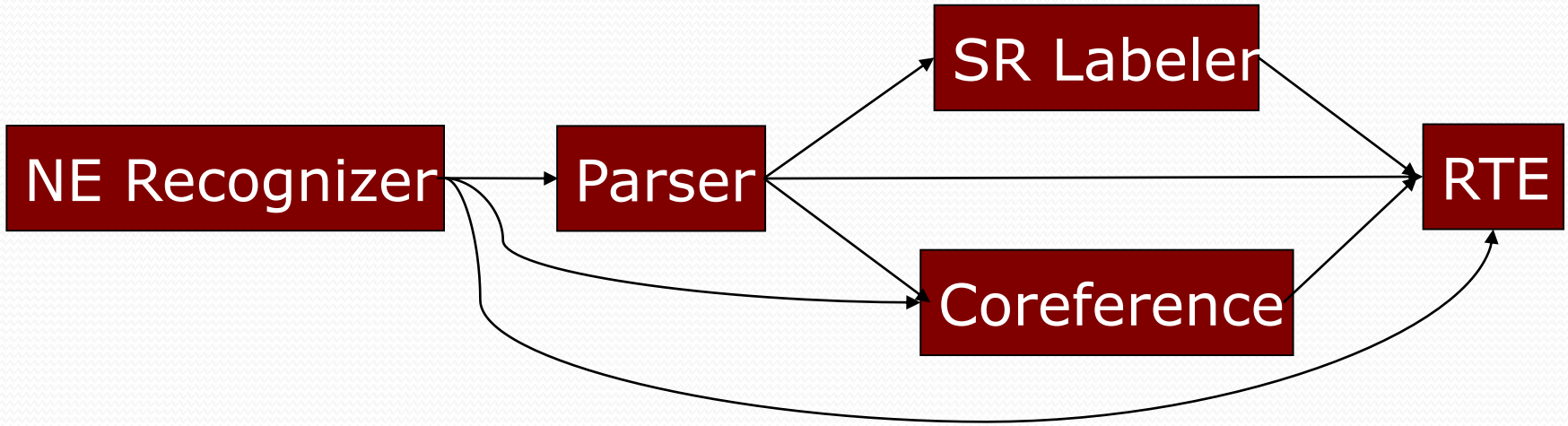


ARG0[…]
ARG1[…]
ARG-TMP[…]



Yes
No
Yes
Yes
No

Sampling Example



Yes

No

Yes

Yes

No