Information Extraction: Capabilities and Challenges

Ralph Grishman
New York University

What is information extraction?

- Information extraction (IE) is the process of identifying within text instances of specified classes of entities and of predications involving these entities

An example ("management succession")

- Fred Flintstone was named CTO of Time Bank Inc. in 2031.
- The next year he got married, left Time Bank, and became CEO of Dinosaur Savings & Loan.

Person	Company	Position	Year	In/out
Fred Flintstone	Time Bank Inc.	CTO	2031	In
Fred Flintstone	Time Bank Inc.	CTO	2032	Out
Fred Flintstone	Dinosaur Savings & Loan	CEO	2032	In

Characteristics of IE

- Only selected relationships are extracted
  - Ignore “got married”
- Different expressions for the same relationship are recognized
  - “was named”, “became”
- References to entities and dates are resolved
  - “he” → “Fred Flintstone”
  - “the next year” → 2032
- Information about individuals (no quantifiers)

Value of IE

- IE makes the information in text accessible for further computer processing …
  creating a data base with one table for each relationship of interest
- Makes it possible to answer questions such as "How many executives has D S&L hired in the last 10 years?"

Some history

- Zellig Harris
- Naomi Sager / Linguistic String Project
- Gerald DeJong / FRUMP

A History of Evaluations

Research in IE has been driven by a series of multi-site evaluations organized by the US Government …

- Message Understanding Conferences (MUC)
  - MUC-1 (1988) to MUC-7 (1998)
- Automatic Content Extraction (ACE)
  - Annually from 2000 to 2008
  - Trilingual (English / Chinese / Arabic)
  - Extensive annotated corpora
- Knowledge Base Population (KBP)
  - Since 2009
  - Large text corpus
  - Collect information about individuals across corpus
- These mostly involved 'general news'
  - Will discuss other extraction domains at the end

Learning to Extract

- There has been a gradual shift from hand-coded rules to systems which can learn from (partially) annotated corpora
    - Part of a general trend in NLP
- We will follow this trend for each type of extraction
    - And will begin with a quick review of relevant machine learning methods

Don't believe all you read

- IE technology has come a long way in 20 years (since MUC-1)
  - Techniques for some IE tasks are now well understood and commercially viable
- But many problems remain
  - Papers report results under very favorable conditions
  - Obscuring the limitations of current technology
  - Which offer the opportunity for many research projects
  - We will look at some of these limitations as part of our course

Course Outline

- Machine learning preliminaries
- Name extraction
- Entity extraction
- Relation extraction
- Event extraction
- Other domains

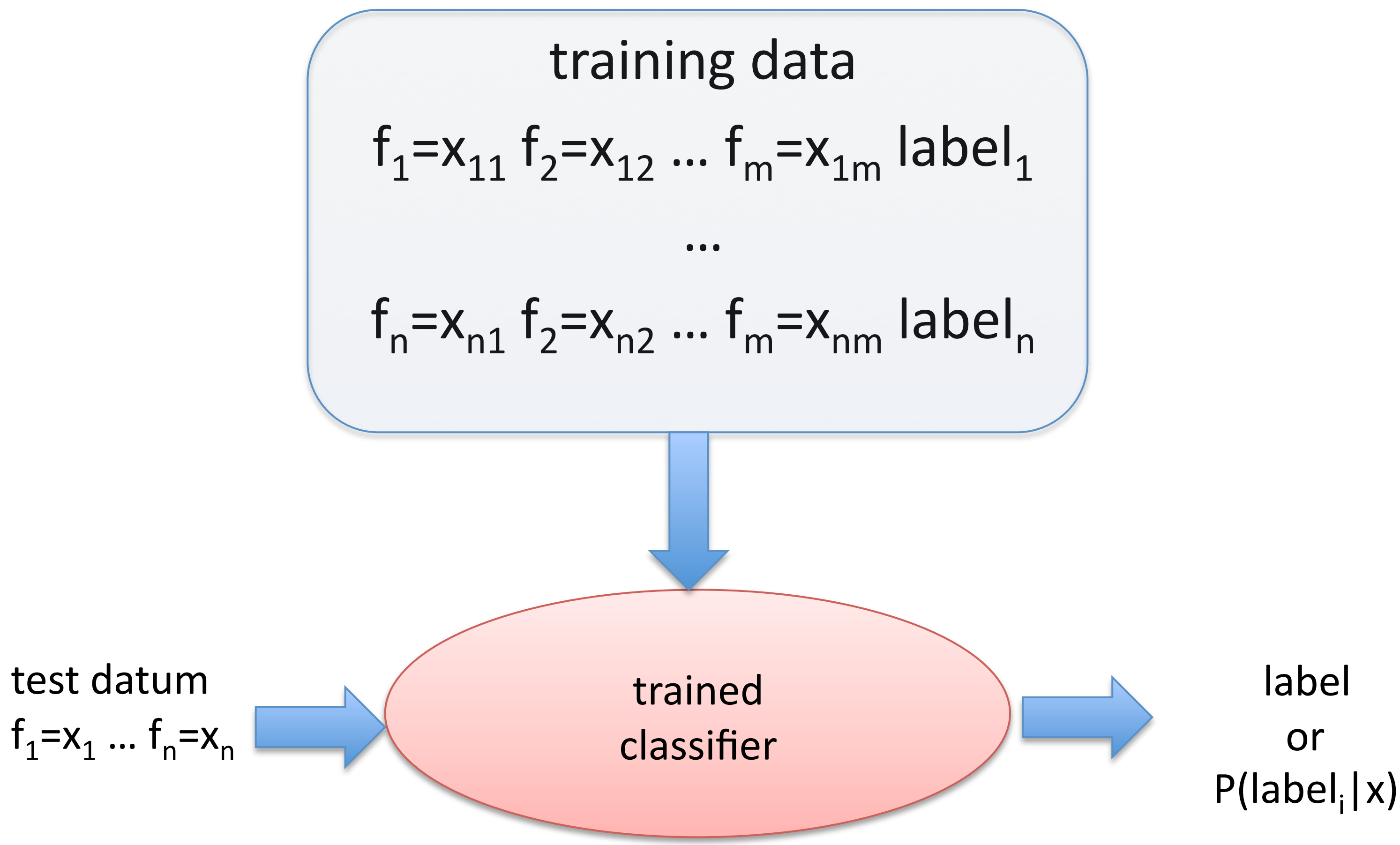
Course Outline

- Machine learning preliminaries
- Name extraction
- Entity extraction
- Relation extraction
- Event extraction
- Other domains

Classifiers

- A *classifier* assigns to a data item x one of a finite set of labels y
  - Two labels: binary classifier
  - More than two labels: n-ary classifier
  - In general, a data item will be viewed as a set of feature-value pairs
- A *trainable classifier* accepts a labeled training set $\{(x_1, y_1), \dots (x_n, y_n)\}$ and produces a classifier which can label any data item x

Trainable Classifier as a 'Black Box'



Popular trainable classifiers

- Maximum entropy classifier
- Support Vector Machine (SVM)

Maximum Entropy Classifier

General form

$$P(c \mid x) = \frac{1}{Z} \exp \sum_{j=0}^{N} w_j h_j(c, x)$$

where

Z = normalizing constant

h_j = j^{th} indicator function, of the form f_i=x_i AND c=label

w_j = weight assigned to j^{th} indicator function by training procedure

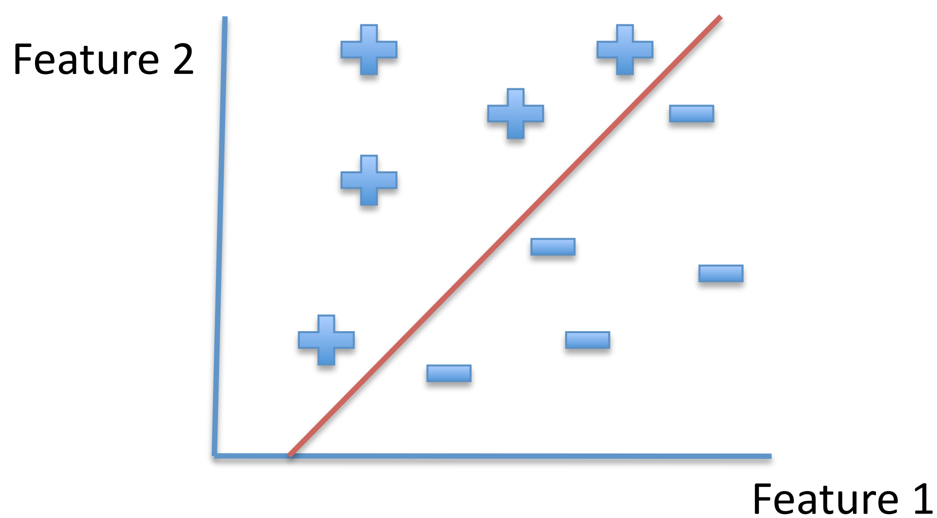
Maximum Entropy Classifier

- Positive w_j: feature makes class more likely
  - Ex: word ends in –ly and POS=adverb
- Negative w_j: feature makes class less likely
  - Ex: word ends in –ly and POS=adjective

- Characteristics
  - Effect of features combined multiplicatively
  - Produces label and its probability
  - Naturally handles n-way classification

Support Vector Machine

- Binary classifier
- Given linearly separable data, constructs a hyperplane separating positive from negative data
    - Chooses plane with maximal margin



Sequence models

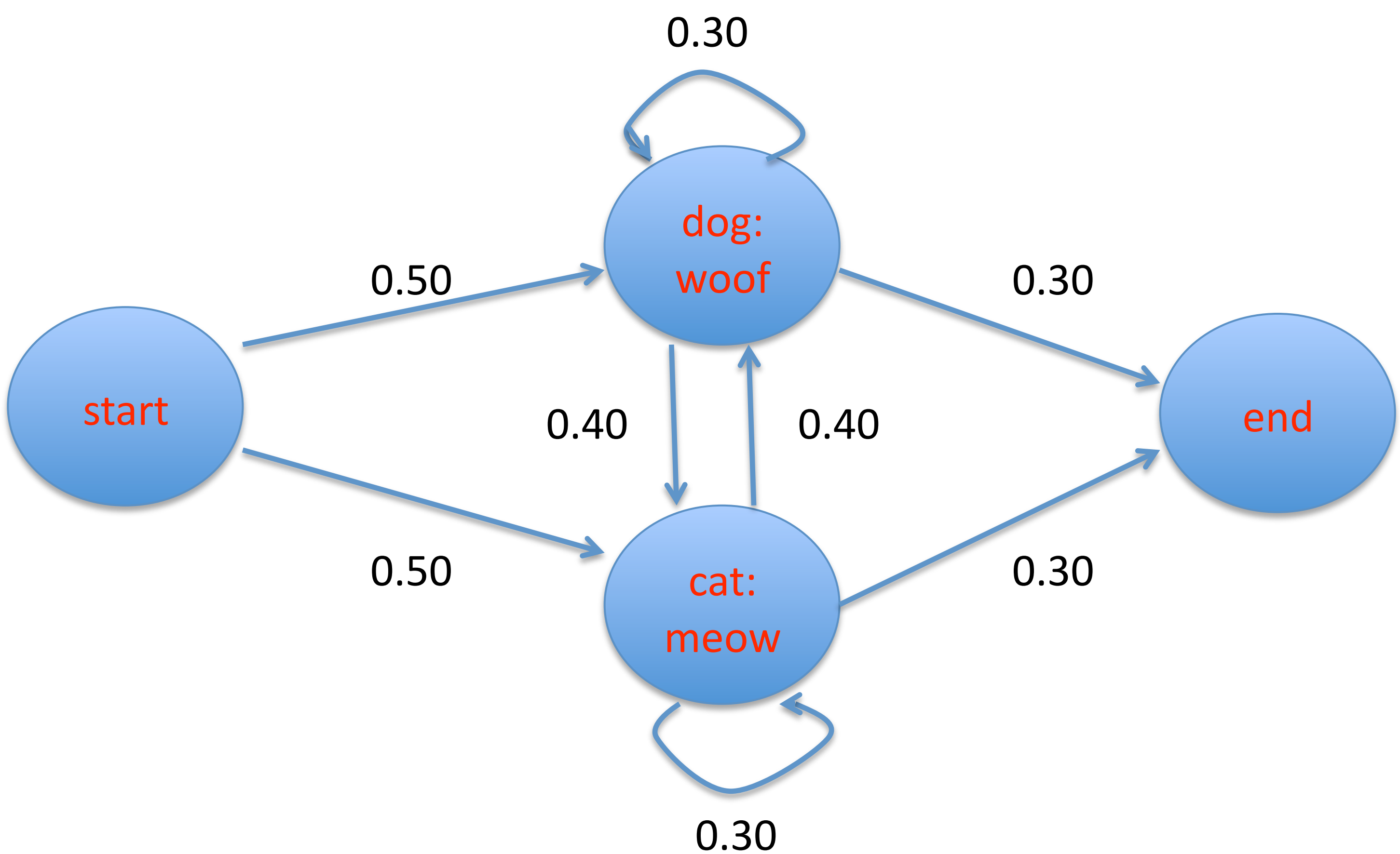
- Classifiers such as MaxEnt and SVM are fine when we have to classify items independently
    - E.g., classifying documents in a collection
- But often in NLP we have to classify every element in a sequence
    - E.g., part of speech tagging
    - Then decisions are not independent

Markov Model

- In principle each decision could depend on all the decisions which came before (the tags on all preceding words in the sentence)

- But we'll make life simple by assuming that the decision depends on only the immediately preceding decision
    - [first-order] Markov Model
    - representable by a finite state transition network
    - T_{ij} = probability of a transition from state i to state j

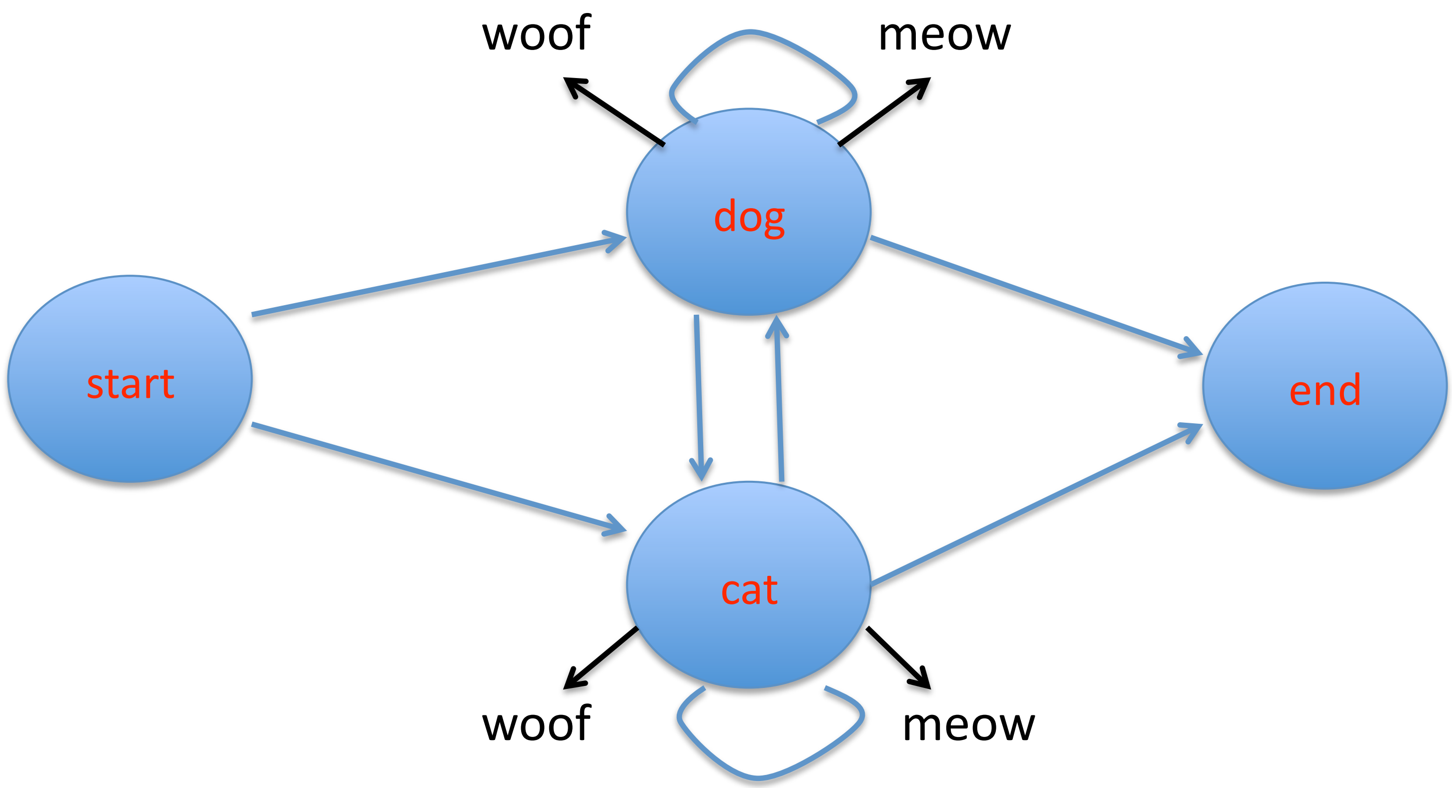
Finite State Network



Our bilingual pets

- Suppose our cat learned to say “woof” and our dog “meow”
- … they started chatting in the next room
- … and we wanted to know who said what

Hidden State Network



- How do we predict
  - When the cat is talking: t_i = cat
  - When the dog is talking: t_i = dog

- We construct a probabilistic model of the phenomenon

- And then seek the most likely state sequence S

$$S = \underset{t_1 \ldots t_n}{\arg\max}\, P(t_1 \ldots t_n \mid w_1 \ldots w_n)$$

Hidden Markov Model

- Assume current word depends only on current tag

$$S = \underset{t_1 \ldots t_n}{\arg\max} P(t_1 \ldots t_n \mid w_1 \ldots w_n)$$

$$= \underset{t_1 \ldots t_n}{\arg\max} P(w_1, \ldots, w_n \mid t_1, \ldots, t_n) P(t_1, \ldots, t_n)$$

$$= \underset{t_1 \ldots t_n}{\arg\max} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_i \mid t_{i-1})$$

Benefits of HMM

- Easy to train from a tagged corpus:
  - just count
    - frequency of state given prior state
    - frequency of word given state
- Fast and easy to apply (“decode”):
  - Viterbi algorithm (form of dynamic programming)
  - linear in length of input

Maximum Entropy Markov Model

$$S = \underset{t_1 \ldots t_n}{\arg\max} P(t_1 \ldots t_n \mid w_1 \ldots w_n)$$

$$= \underset{t_1 \ldots t_n}{\arg\max} \prod_{i=1}^{n} P(t_i \mid t_{i-1}, w_1, \ldots, w_n)$$

P is implemented by a MaxEnt model.

Note that P is conditioned only on the immediately prior state (Markov constraint) but can access the entire word sequence. This offers great flexibility in devising features for the MaxEnt model.

Flavors of learning

- Supervised learning
  - All training data is labeled
- Semi-supervised learning
  - Part of training data is labeled ('the seed')
  - Make use of redundancies to learn labels of additional data, then train model
  - Co-training
  - Reduces amount of data which must be hand-labeled to achieve a given level of performance
- Active learning
  - Start with partially labeled data
  - System selects additional 'informative' examples for user to label

Semi-supervised learning

L = labeled data

U = unlabeled data

1. L = seed
   -- repeat 2-4 until stopping condition is reached
2. C = classifier trained on L
3. Apply C to U.
   N = most confidently labeled items
4. L += N; U -= N

Confidence

How to estimate confidence?

- Binary probabilistic classifier
  - Confidence = | P – 0.5 | * 2
- N-ary probabilistic classifier
  - Confidence = $P_1 - P_2$

    where

    P_1 = probability of most probable label

    P_2 = probability of second most probable label
- SVM
  - Distance from separating hyperplane

Co-training

- Two 'views' of data (subsets of features)
  - Producing two classifiers $C_1(x)$ and $C_2(x)$
- Ideally
  - Independent
  - Each sufficient to classify data
- Apply classifiers in alternation (or in parallel)

1. L = seed
   -- repeat 2-7 until stopping condition is reached
2. C_1 = classifier trained on L
3. Apply C_1 to U.
   N = most confidently labeled items
4. L += N; U -= N
5. C_2 = classifier trained on L
6. Apply C_2 to U.
   N = most confidently labeled items
7. L += N; U -= N

Problems with semi-supervised learning

- When to stop?
  - U is exhausted
  - Reach performance goal using held-out labeled sample
  - After fixed number of iterations based on similar tasks

- Poor confidence estimates
  - Errors from poorly-chosen data rapidly magnified

Course Outline

- Machine learning preliminaries
- Name extraction
- Entity extraction
- Relation extraction
- Event extraction
- Other domains

Name Extraction

- Fred Flintstone was named CTO of Time Bank Inc. in 2031.

- The next year he got married, left Time Bank, and became CEO of Dinosaur Savings & Loan.

Name Extraction

- Names are very common
  - Most news sentences have one or more
  - Want to treat names as a unit for most processing
  - `Rules' separate from those of general grammar
- Introduced as a separate task for MUC-6 (1995) for English news IE
  - Good name recognition seen as essential for IE
  - Rapidly extended to many other languages
  - MET, CoNLL multi-lingual tasks
- Now considered essential for QA, helpful for MT

Name Categories

- MUC started with 3 name categories: *person, organization, location*
- QA and some IE required much finer categories
  - Led to sets with 100-200 name categories
  - Hierarchical categories

Excerpt from a Detailed Name Ontology (Sekine 2008)

- Organization
- Location
- Facility
- Product
  - Product_Other, Material, Clothing, Money, Drug, Weapon, Stock, Award, Decoration, Offense, Service, Class, Character, ID_Number
  - Vehicle : Vehicle_Other, Car, Train, Aircraft, Spaceship, Ship
  - Food : Food_Other, Dish
  - Art : Art_Other, Picture, Broadcast_Program, Movie, Show, Music, Book
  - Printing : Printing_Other, Newspaper, Magazine
  - Doctrine_Method : Doctrine_Method_Other, Culture, Religion, Academic, Style, Movement, Theory, Plan
  - Rule : Rule_Other, Treaty, Law
  - Title : Title_Other, Position_Vocation
  - Language : Language_Other, National_Language
  - Unit : Unit_Other, Currency …

Systematic Name Polysemy

- Some names have multiple senses
  - Spain
    - Spain is south of France [geographic region]
    - Spain signed a treaty with France [the government]
    - Spain drinks lots of wine [the people]
  - McDonalds
    - McDonalds sold 3 billion Happy Means [the organization]
    - I'll meet you in front of McDonalds [the location]
- Designate a primary sense for each systematically polysemous name type
    - ACE introduced "GPE" = geo-political entity for regions with governments in recognition of this most common polysemy

Approaches to NER

- Hand-coded rules
- Supervised models
- Semi-supervised models
- Active learning

Hand-Coded Rules for NER

For people:

- title (capitalized-token)+
  - where title = “Mr.” | “Mrs.” | “Ms.” | …
- capitalized-token initial capitalized-token
- common-first-name capitalized-token
  - American first names available from census
- capitalized-token capitalized-token , 1-or-2-digit-number ,

For organizations

- (capitalized-token)+ corporate-suffix
  - where corporate-suffix = “Co.” | “Ltd.” | …

For locations

- capitalized-token , country

Burden of hand-crafted rules

- Writing a few rules is easy
- Writing lots of rules … capturing all the indicative contexts … is hard
    - ____ died
    - ____ was founded
- At some point additional rules may hurt performance
    - Need an annotated ‘development test’ corpus to check progress
- Once we have an annotated corpus, can we use it to automatically train an NER … *a supervised model*?

BIO Tags

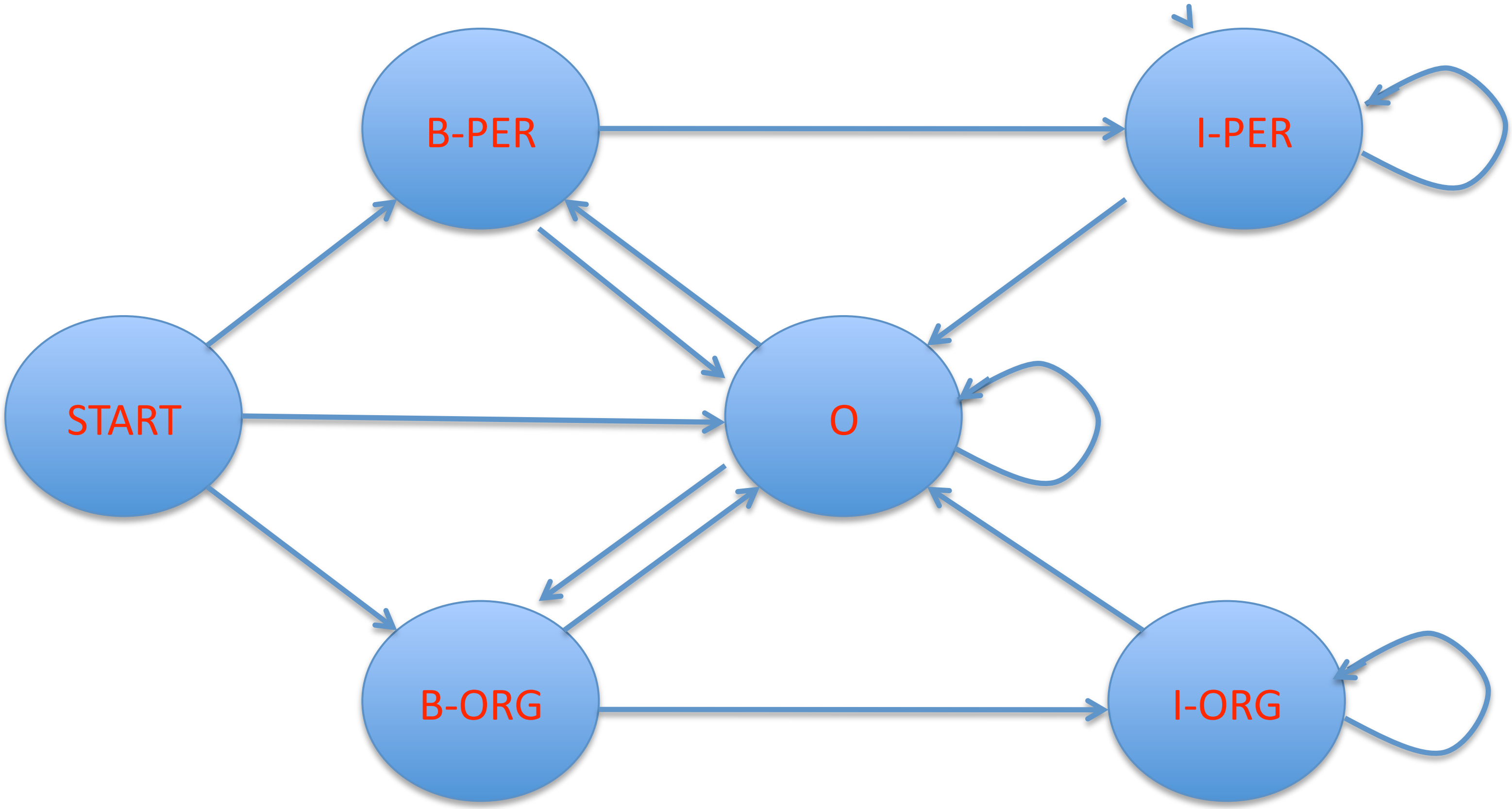
- How can we formulate NER as a standard ML problem?
- Use BIO tags to convert NER into a sequence tagging problem, which assigns a tag to each token:
  - For each NE category c_i, introduce tags B-c_i [beginning of name] and I-c_i [interior of name]
  - Add in category O [other]
  - For example, with categories per, org, and loc, we would have 7 tags B-per, I-per, B-org, I-org, B-loc, I-loc, and O
  - Require that I-c_i be preceded by B-c_i or I-c_i

Fred	lives	in	New	York
B-per	O	O	B-loc	I-loc

Using a Sequence Model

- Construct network with one state for each tag
    - 2n+1 states for n categories, plus start state
- Train model parameters using annotated corpus
    - HMM or MEMM model
- Apply trained model to new text
    - Find most likely path through network (Viterbi)
    - Assign tags to tokens corresponding to states in path
    - Convert BIO tags to names

A Minimal State Diagram for NER



Only two name classes; assumes two names are separated by at least one ‘O’ token.

Using a MEMM for NER

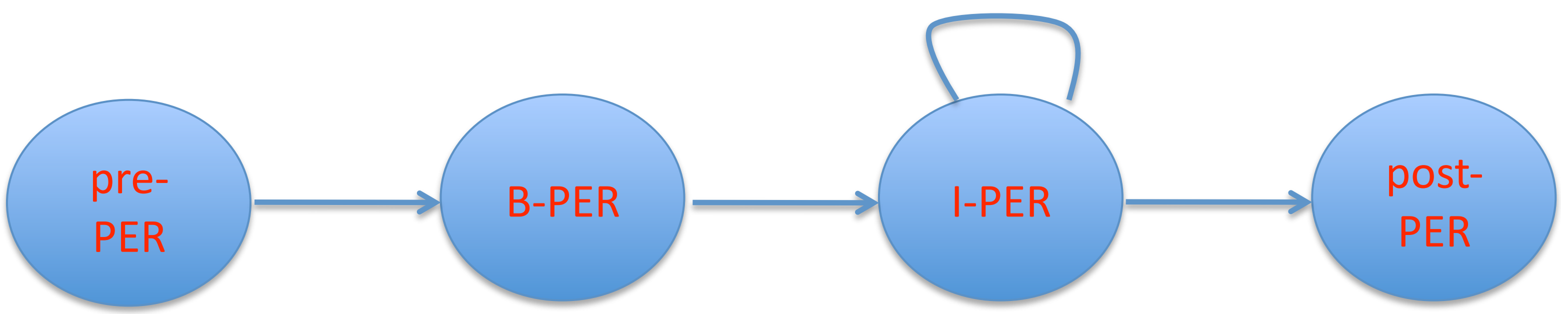
- Simplest MEMM …
  - $P(s_i \mid s_{i-1}, w_i)$
  - Have prior state, current word, (current word & prior state) as features
- Getting some context
  - Add prior word (w_{i-1}) as feature
  - Add next word (w_{i+1}) as feature

Adding States for Context

If we are using an HMM, can get context through pre-person and post-person states

Changing

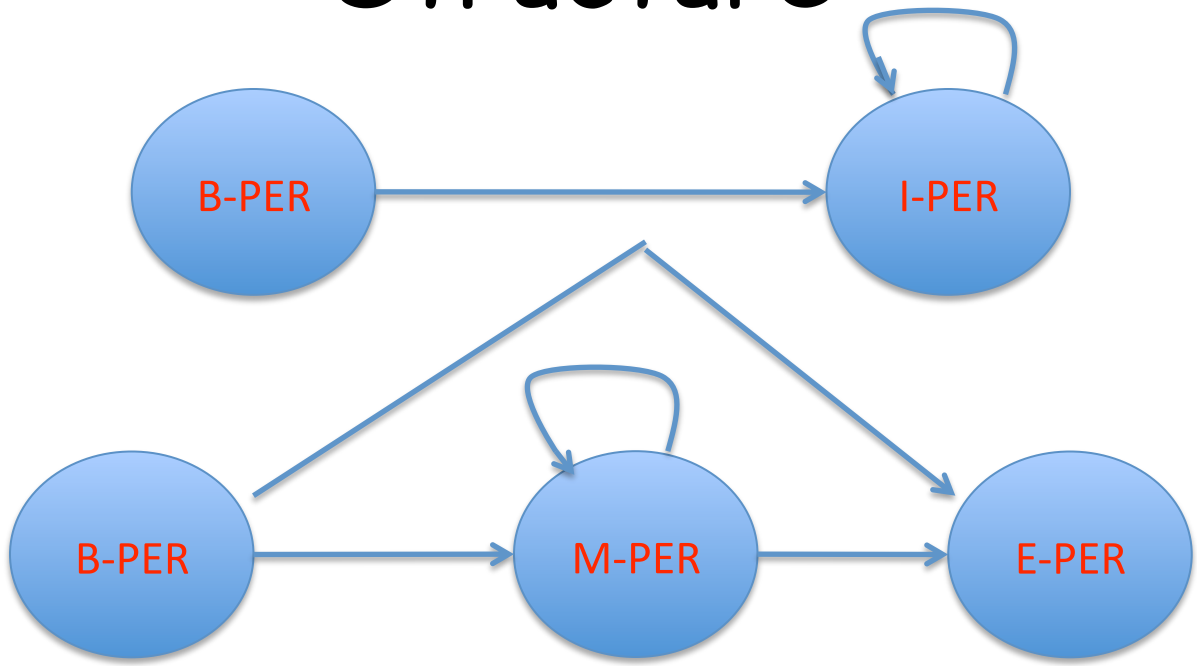
to



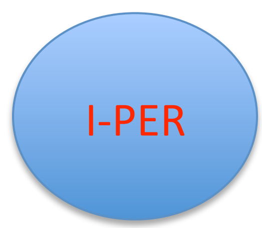
Adding States for Name Structure

Changing

to

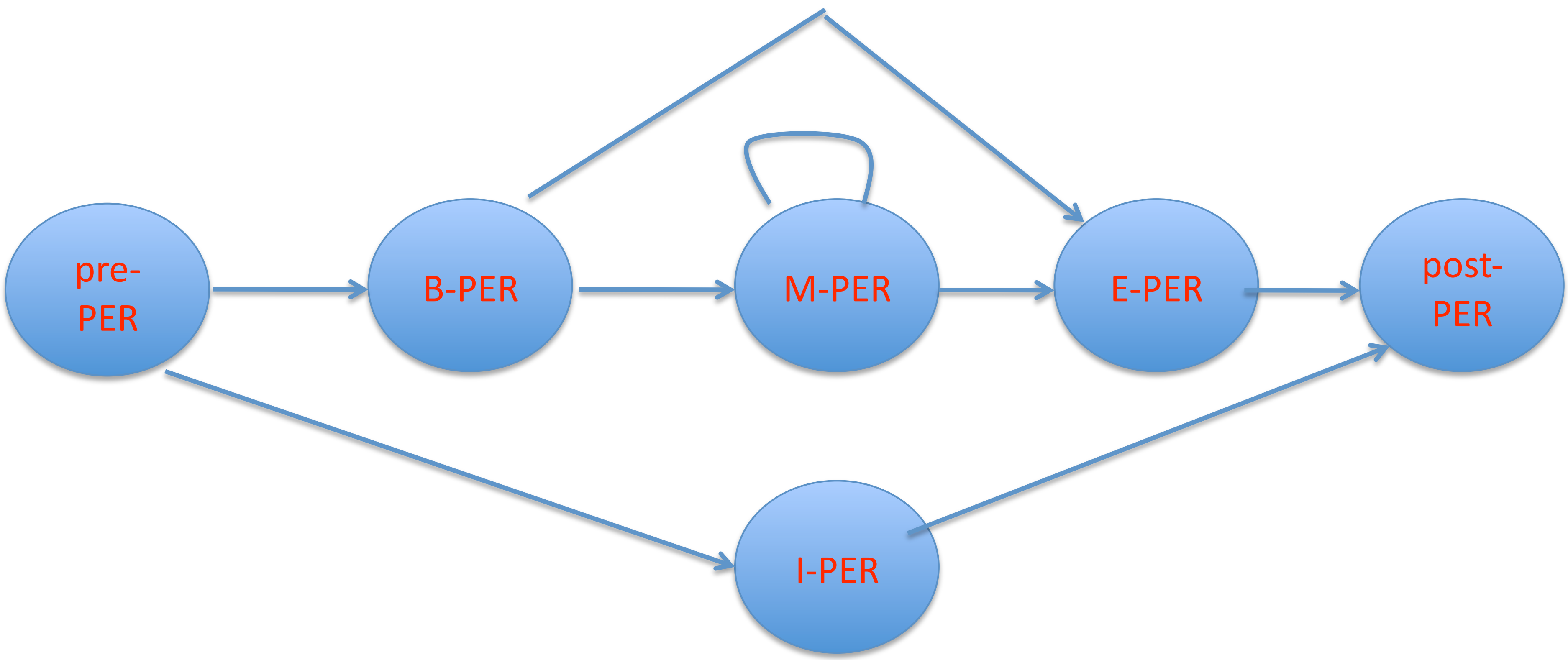


improves performance
by capturing more details
of name structure



Different languages have different name structure -- best recognized by language-specific states

Putting them together



More Local Features

- Lexical features
  - Whether the current word (prior word, following word) has a specific value
- Dictionary features
  - Whether the current word is in a particular dictionary
  - Full name dictionaries
    - For major organizations, countries, and cities
  - Name component dictionaries
    - Common first names
- Word clusters
  - Whether the current word belongs to a corpus-derived word cluster
- Shape features
  - Capitalized, all caps, numeric, 2-digit numeric, …
- Part-of-speech features
- Hand-coded NER rules as features

Long-range features [1]

- Most names represent the same name type (person / org / location) wherever they appear
  - Particularly within a single document
  - But in most cases across documents as well
- Some contexts will provide a clear indication of the name type, while others will be ambiguous
  - We would like to use the unambiguous contexts to resolve the ambiguity across the document or the corpus
- Ex:
  - On vacation, Fred visited Gilbert Park.
  - Mr. Park was an old friend from college.

Long-range features [2]

- We can capture this information with a two-pass strategy …
  - On the first pass, build a table ("name cache") which records each name and they type it is assigned
    - Possibly record only confident assignments
  - On the second pass, incorporate a feature reflecting the dominant name type from the first pass
- This can be done across an individual document or a large corpus [Borthwick 1999]

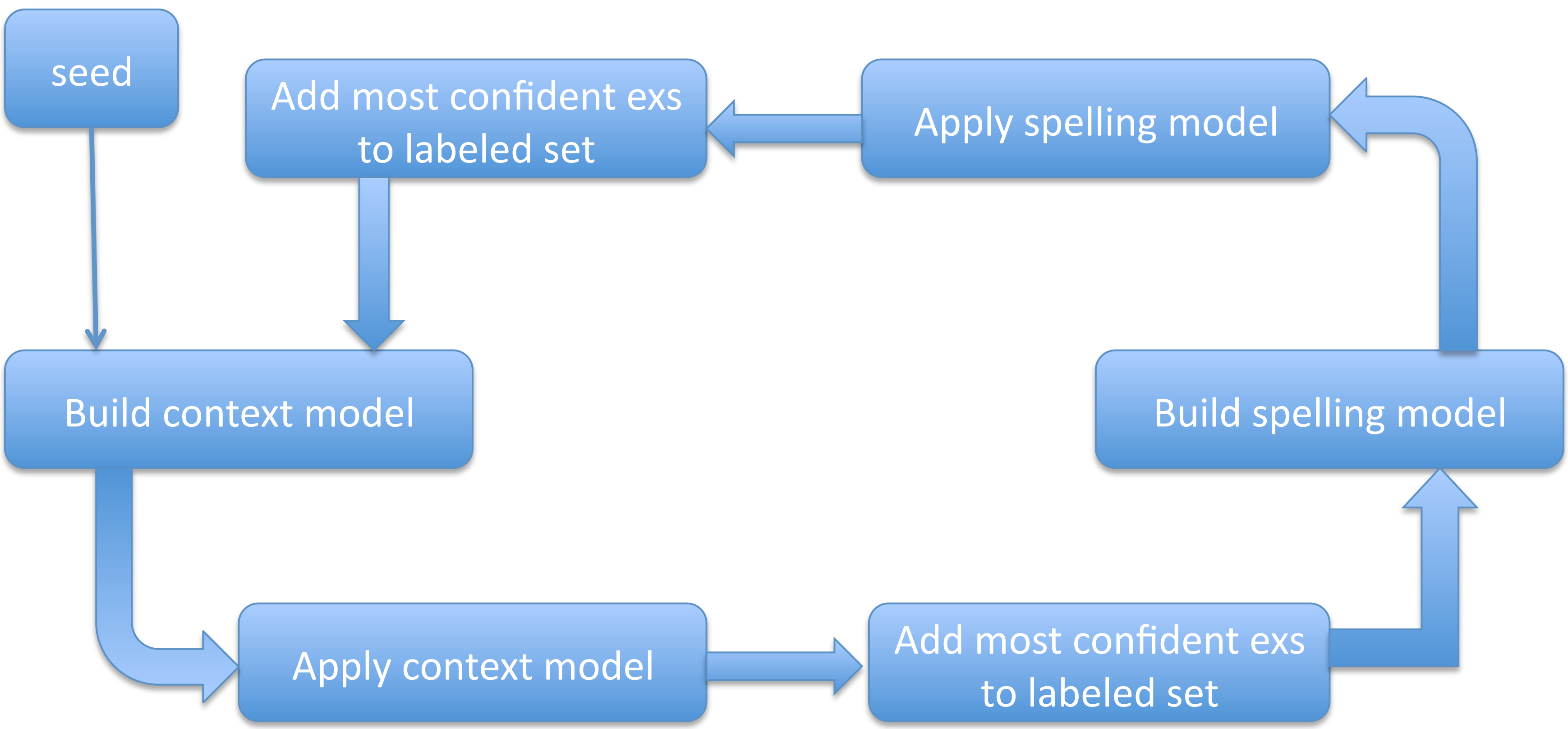
Semi-supervised NER

- Annotating a large corpus to train a high-performance NER is fairly expensive
- We can use the same idea (of name consistency across documents) to train an NER using
  - A smaller annotated corpus
  - A large unannotated corpus

Co-training for NER

- We can split the features for NER into two sets:
  - Spelling features
    (the entire name + tokens in the name)
  - Context features
    (left and right contexts + syntactic context)
- Start with a seed
  - E.g., some common unambiguous full names
- Iteratively grow seed, alternatively applying spelling and context models and adding most - confidently-labeled instances to seed

Co-training for NER



Name co-training: results

- 3 classes: person, organization, location (and 'other')
- Data: 1M sentences of news
- Seed:
    - New York, California, U.S. → location
    - contains(Mr.) → person
    - Microsoft, IBM → organization
    - contains(Incorporated) → organization
- Took names appearing with appositive modifier or as complement of preposition (88K name instances)
- Accuracy: 83%
- Clean accuracy (ignoring names not in one of the 3 categories): 91%

- (Collins and Singer 1999)

Semi-supervised NER: when to stop

- Semi-supervised NER labels a few more examples at every iteration
  - It stops when it runs out of examples to label
- This is fine if
  - Names are easily identified (e.g., by capitalization in English)
  - Most names fall into one of the categories being trained (e.g., people, organizations, and locations for news stories)

Semi-supervised NER: semantic drift

- Semi-supervised NER doesn't work so well if
  - The set of names is hard to identify
    - Monocase languages
    - Extended name sets including lower-case terms
  - The categories being trained cover only a small portion of the set of names
- The result is *semantic drift* and *semantic spread*
  - The name categories gradually grow to include related terms

Fighting Semantic Drift

- We can fight drift by training a larger, more inclusive set of categories
  - Including ‘negative’ categories
    - Categories we don’t really care about but include to compete with the original categories
  - These negative categories can be built
    - By hand (Yangarber et al. 2003)
    - Or automatically (McIntosh 2010)

Active Learning

- For supervised learning, we typically annotate text data sequentially
- Not necessarily the most efficient approach
  - Most natural language phenomena have a Zipfean distribution … a few very common constructs and lots of infrequent constructs
  - After you have annotated "Spain" 50 times as a location, the NER model is little improved by annotating it one more time
- We want to select the most *informative* examples and present them to the annotator
  - The data which, if labeled, is most likely to reduce NER error

How to select informative examples?

- Uncertainty-based sampling
  - For binary classifier
    - For MaxEnt, probability near 50%
    - For SVM, data near separating hyperplane
  - For n-ary classifier, data with small margin
- Committee-based sampling
  - Data on which committee members disagree
  - (co-testing … use two classifiers based on independent views)

Representativeness

- It's more helpful to annotate examples involving common features
    - Weighting these features correctly will have a larger impact on error rate

- So we rank examples by frequency of features in the entire corpus

Batching and Diversity

- Each iteration of active learning involves running classifier on (a large) unlabeled corpus
  - This can be quite slow
  - Meanwhile annotator is waiting for something to annotate
- So we run active learning in batches
  - Select best n examples to annotate each time
  - But all items in a batch are selected using the same criteria and same system state, and so are likely to be similar
- To avoid example overlap, we impose a diversity requirement with a batch: limit maximum similarity of examples within a batch
  - Compute similarity based on example feature vectors

Simulated Active Learning

- True active learning experiments are
  - Hard to reproduce
  - Very time consuming
- So most experiments involve *simulated active learning*:
  - "unlabeled" data has really been labeled, but the labels have been hidden
  - When data is selected, labels are revealed
  - Disadvantage: "unlabeled" data can't be so bit
- This leads us to ignore lots of issues of true active learning:
  - An annotation unit of one sentence or even one token may not be efficient for manual annotation
  - So reported speed-ups may be optimistic
    (typical reports reduce by half the amount of data to achieve a given NER accuracy

Evaluating NER

- Systems are evaluated using an annotated test corpus
  - Ideally dual annotated and adjudicated
- Name tags in system output are classified as correct, spurious, or missing:

	Cervantes	wrote	Don Quixote	in	Tarragona.
System:	person		person		
Reference:	person				location
	correct		*spurious*		*missing*

Metrics

- Systems are measured in terms of:

$$recall = \frac{correct}{correct + missing}$$

$$precision = \frac{correct}{correct + spurious}$$

$$F = \frac{2 \times recall \times precision}{recall + precision}$$

Typical Performance

- News corpora
  - Training and test from same source
- 3 categories: person, organization, location
- Based on CoNLL 2002 and 2003 multi-lingual, multi-site evaluations
  - English F = 89
  - Spanish F = 81
  - Dutch F = 77
  - German F = 72

Limitations

- Cited performance is for well matched training and test
    - Same domain
    - Same source
    - Same epoch
  - Performance deteriorates rapidly if less matched
    - NER trained on Reuters (F=91),
      tested on Wall Street Journal (F=64) [Ciaramita and Altun 2003]
  - Work on NER adaptation is vital
- Adding rarer classes to NER is difficult
  - Supervised learning inefficient
  - Semi-supervised learning is subject to semantic drift

Course Outline

- Machine learning preliminaries
- Name extraction
- Entity extraction
- Relation extraction
- Event extraction
- Other domains

Names, mentions, and entities

- Information extraction gathers information about discrete <u>entities</u> such as people, organizations, vehicles, books, cats, etc.
- Texts contain <u>mentions</u> of these entities; these mentions may take the form of
    - Names ("Sarkozy")
    - Noun phrases headed by nouns ("the president")
    - Pronouns ("he")

Reference and co-reference

- Data base entries filled with nouns or pronouns are not very useful …
  - At a minimum, entries should be names
- But even names may be ambiguous
  - So we may want to create a data base of entities with unique ID's
  - And express relations and events in terms of these ID's

In-document coreference

- The first step is in-document coreference – linking all mentions in a document which refer to the same entity
  - If one of these mentions is a name, this allows us to use the name in the extracted relations
- Coreference has been extensively studied independently of IE
  - Typically by constructing statistical models of the likelihood that a pair of mentions are coreferential
  - We will not review these models here

Cross-document [co]reference

- Cross-document coreference links together the entities mentioned by individual documents
  - Generally limited to entities which are named in both documents
- Entity linking links an entity named in one document to an entity in a data base

Cross-document [co]reference

- Studied mainly in an IE setting
    - ACE 2008
    - KBP 2009-2010-2011
    - WePS
- Involves modeling
    - Possible spelling / name variation
        - William Jefferson Clinton ←→ Bill Clinton
        - Osama bin Laden ← → Usama bin Laden
    - Probable coreference based on
        - Shared / conflicting attributes
        - Co-occurring terms / names

Course Outline

- Machine learning preliminaries
- Name extraction
- Entity extraction
- Relation extraction
- Event extraction
- Other domains

Relation

- A *relation* is a predication about a pair of entities:
  - Rodrigo works for UNED.
  - Alfonso lives in Tarragona.
  - Otto's father is Ferdinand.
- Typically they represent information which is permanent or of extended duration.

History of relations

- Relations were introduced in MUC-7 (1997)
  - 3 relations
- Extensively studied in ACE (2000 – 2007)
  - lots of training data
- Effectively included in KBP

ACE Relations

- Several revisions of relation definitions
    - With goal of having a set of relations which can be ore consistently annotated
- 5-7 major types, 19-24 subtypes
- Both entities must be mentioned in the same sentence
    - Do not get a parent-child relation from
        - Ferdinand and Isabella were married in 1481.
          A son was born in 1485.
    - Or an employee relation for
        - Bank Santander replaced several executives. Alfonso was named an executive vice president.
- Base for extensive research
    - On supervised and semi-supervised methods

2004 Ace Relation Types

Relation type	Subtypes
Physical	Located, Near, Part-whole
Personal-social	Business, Family, Other
Employment / Membership / Subsidiary	Employ-executive, Employ-staff, Employ-undetermined, Member-of-group, Partner, Subsidiary, Other
Agent-artifact	User-or-owner, Inventor-or-manufacturer, Other
Person-org affiliation	Ethnic, Ideology, Other
GPE affiliation	Citizen-or-resident, Based-in, Other
Discourse	-

KBP Slots

- Many KBP slots represent relations between entities:
    - Member_of
    - Employee_of
    - Country_of_birth
    - Countries_of_residence
    - Schools_attended
    - Spouse
    - Parents
    - Children …

- Entities do not need to appear in the same sentence
- More limited training data
    - Encouraged semi-supervised methods

Characteristics

- Relations appear in a wide range of forms:
  - Embedded constructs (one argument contains the other)
    - within a single noun group
      - John’s wife
    - linked by a preposition
      - the president of Apple
  - Formulaic constructs
      - Tarragona, Spain
      - Walter Cronkite, CBS News, New York
  - Longer-range (‘predicate-linked’) constructs
    - With a predicate disjoint from the arguments
      - Fred lived in New York
      - Fred and Mary got married

Hand-crafted patterns

- Most instances of relations can be identified by the types of the entities and the words between the entities
    - But not all: Fred and Mary got married.
- So we can start by listing word sequences:
    - Person lives in location
    - Person lived in location
    - Person resides in location
    - Person owns a house in location
    - …

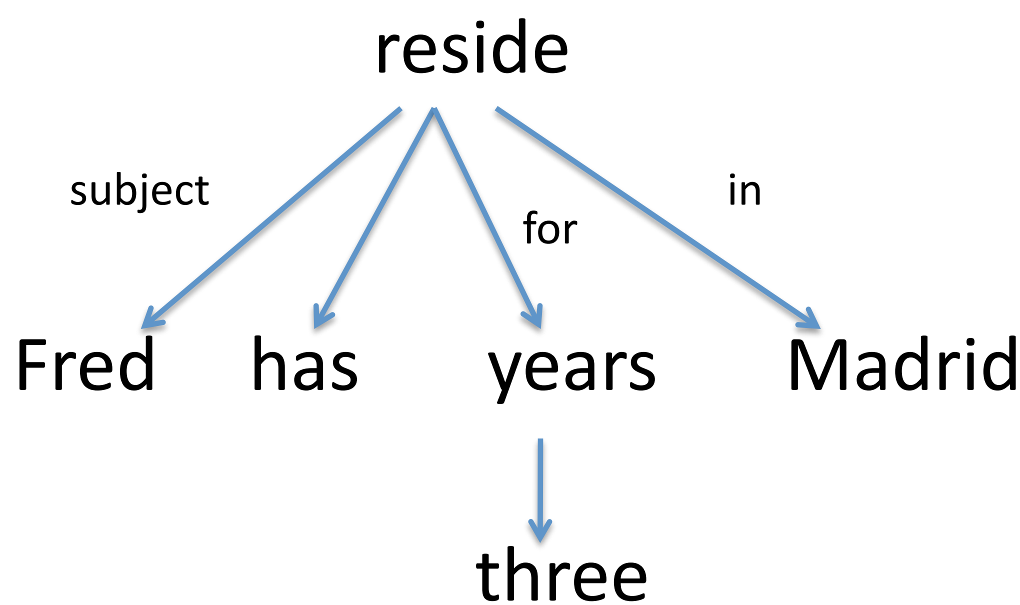
Generalizing patterns

- We can get better coverage through syntactic generalization:
  - Specifying base forms
    - Person <v base=reside> in location
  - Specifying chunks
    - Person <vgroup base=reside> in location
  - Specifying optional elements
    - Person <vgroup base=reside> [<pp>] in location

Dependency paths

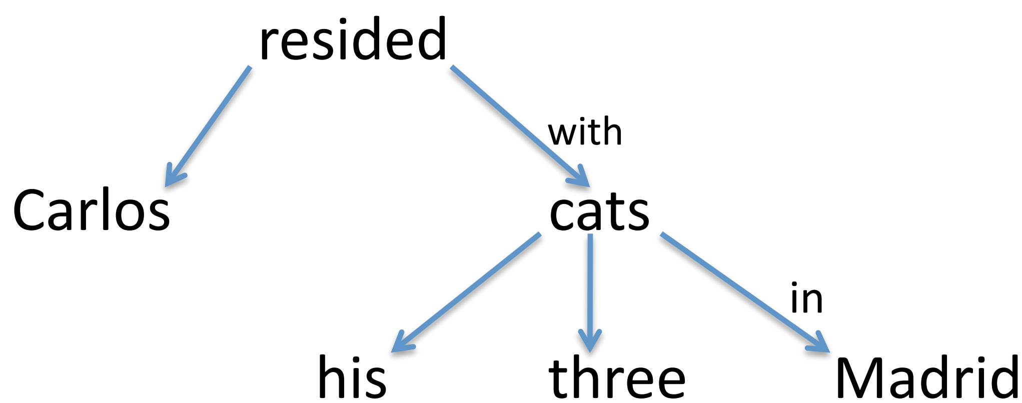
- Generalization can also be achieved by using paths in labeled dependency trees:

  person – subject^{-1} – reside – in -- *location*



Pattern Redundancy

- Using a combination of sequential patterns and dependency patterns may provide extra robustness
  - Dependency patterns can handle more syntactic variation but are more subject to analysis errors: "Carlos resided with his three cats in Madrid."



Supervised learning

- Collect training data
  - Annotate corpus with entities and relations
  - For every pair of entities in a sentence
    - If linked by a relation, treat as positive training instance
    - If not linked, treat as a negative training instance
- Train model
  - For *n* relation types, either
    - Binary (identification) model + n-way classifier model or
    - Unified *n+1*-way classifier
- On test data
  - Apply entity classifier
  - Apply relation classifier to every pair of entities in same sentence

Supervised relation learner: features

- Heads of entities
- Types of entities
- Distance between entities
- Containment relations
- Word sequence between entities
- Individual words between entities
- Dependency path
- Individual words on dependency path

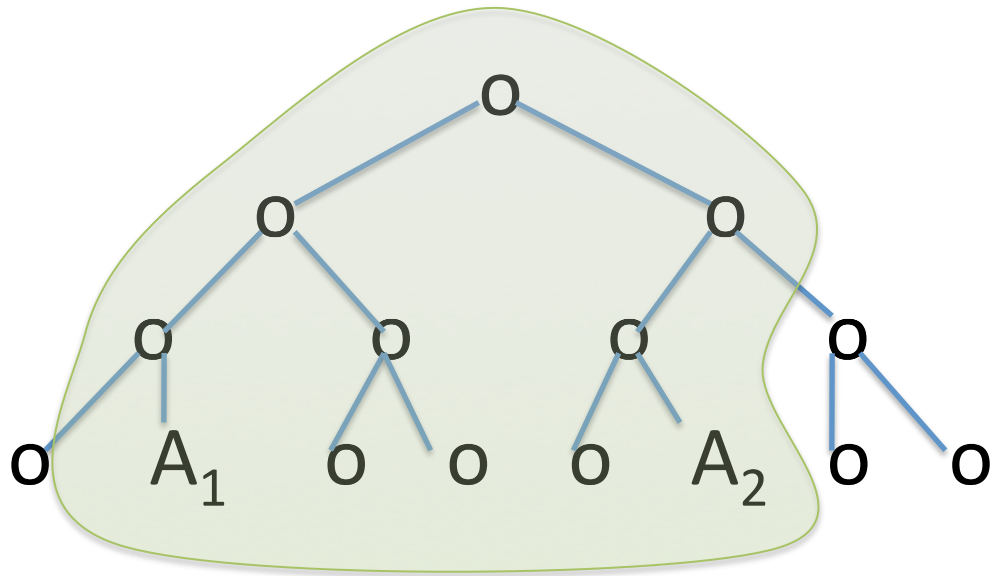
Kernel Methods

- Goal is to find training examples similar to test case
  - Similarity of word sequence or tree structure
  - Determining similarity through features is awkward
  - Better to define a similarity measure directly: a kernel function
- Kernels can be used directly by
  - SVMs
  - Memory-based learners (k-nearest-neighbor)
- Kernels defined over
  - Sequences
  - Parse or Dependency Trees

Tree Kernels

- Tree kernels differ in
  - Type of tree
    - Partial parse
    - Parse
    - Dependency
  - Tree spans compared
    - Shortest path-enclosed tree
    - Conditionally larger context
  - Flexibility of match

Shortest-path-enclosed Tree



- For predicate-linked relations, must extend shortest-path-enclosed tree to include predicate

Composite Kernels

- Can combine different levels of representation
- Composite kernel can combine sequence and tree kernels

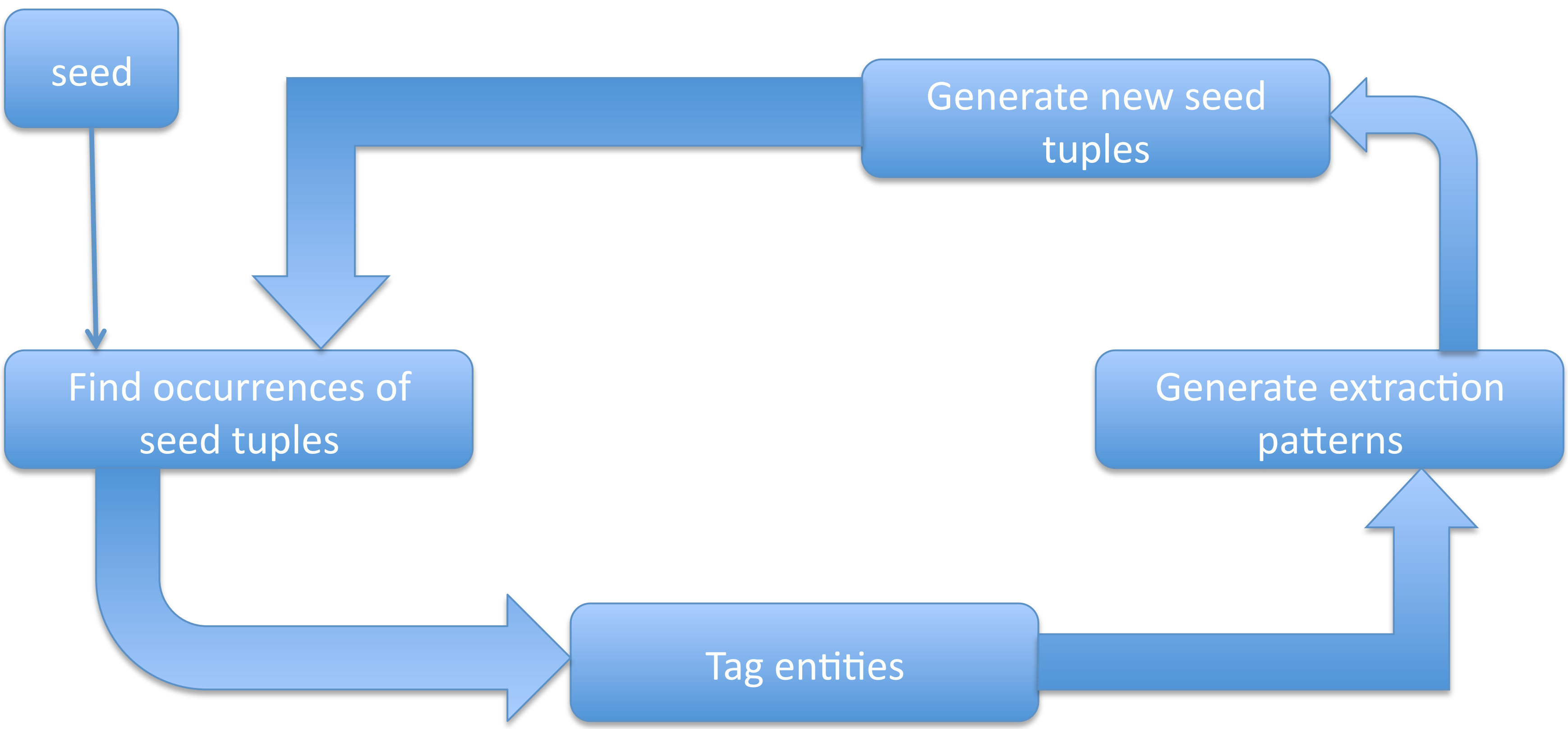
Semi-supervised methods

- Preparing training data is more costly than for names
  - Must annotate entities and relations
- So there is a strong motivation to minimize training data through semi-supervised methods
- As for names, we will adopt a co-training approach:
  - Feature set 1: the two entities
  - Feature set 2: the contexts between the entities
- We will limit the bootstrapping
  - to a specific pair of entity types
  - and to instances where both entities are named

Semi-supervised learning

- Seed:
  - [*Moby Dick,* Herman Melville]
- Contexts for seed:
  - … wrote …
  - … is the author of …
- Other pairs appearing in these contexts
  - [*Animal Farm*, George Orwell]
  - [*Don Quixote,* Miguel de Cervantes]
- Additional contexts …

Co-training for relations



Ranking contexts

- If relation R is functional,
  and [X, Y] is a seed,
  then [X, Y'], Y'≠Y, is a negative example
- Confidence of pattern P

$$Conf(P) = \frac{P.positive}{P.positive + P.negative}$$

- where
  P.positive = number of positive matches to pattern P
  P.negative = number of negative matches to pattern P

Ranking pairs

- Once a confidence has been assigned to each pattern, we can assign a confidence to each new pair based on the patterns in which it appears
  - Confidence of best pattern
  - Combination assuming patterns are independent

$$Conf(X,Y) = 1 - \prod_{P \in contexts\_of\_(X,Y)} (1 - Conf(P))$$

Semantic drift

- Ranking / filtering quite effective for functional relations (book → author, company → headquarters)
  - But expansion may occur into other relations generally implied by seed ('semantic drift')
    - Ex: from governor → state governed to
      person → state born in
- Precision poor without functional property

Distant supervision

- Sometimes a large data base is available involving the type of relation to be extracted
  - A number of such public data bases are now available, such as FreeBase and Yago
- Text instances corresponding to some of the data base instances can be found in a large corpus or from the Web
- Together these can be used to train a relation classifier

Distant supervision: approach

- Given:
    - Data base for relation R
    - Corpus containing information about relation R
- Collect <X, Y> pairs from data base relation R
- Collect sentences in corpus containing both X and Y
    - These are positive training examples
- Collect sentences in corpus containing X and some Y'with the same entity type as Y such that <X,Y'> is not in the data base
    - These are negative training examples
- Use examples to train classifier which operates on pairs of entities

Distant supervision: limitations

- The training data produced through distant supervision may be quite noisy:
- If a pair <X, Y> is involved in multiple relations, R<X, Y> and R'<X, Y> and the data base represents relation R, the text instance may represent relation R', yielding a false positive training instance
  - If many <X, Y> pairs are involved, the classifier may learn the wrong relation
- If a relation is incomplete in the data base … for example, if resides_in<X, Y> contains only a few of the locations where a person has resided … then we will generate many false negatives, possibly leading the classifier to learn no relation at all

Evaluation

- Matching relation has matching relation type and arguments
  - Count correct, missing, and spurious relations
  - Report precision, recall, and F measure
- Variations
  - Perfect mentions vs. system mentions
    - Performance much worse with system mentions – an error in either mention makes relation incorrect
  - Relation type vs. relation subtype
  - Name pairs vs. all mentions
    - Bootstrapped systems trained on name-name patterns
- Best ACE systems on perfect mentions: F = 75

Course Outline

- Machine learning preliminaries
- Name extraction
- Entity extraction
- Relation extraction
- Event extraction
- Other domains

Events and Scenarios

- Event extraction: most general task
  - Multiple arguments and modifiers
  - Most arguments are optional
- MUC task … scenarios
  - Focus on a single topic (terrorist attack, plane crash, union negotiation)
  - Look for larger structure which may include several sub-events
  - Capture connection between these sub-events
- ACE 2005 task … events
  - Seek broad coverage of major news stories
  - Use relatively fine-grained individual events
  - No connections between events

MUC-3 Template (Terrorist incident)

0. MESSAGE ID	TST1-MUC3-0099
1. TEMPLATE ID	1
2. DATE OF INCIDENT	24 OCT 89 - 25 OCT 89
3. TYPE OF INCIDENT	BOMBING
4. CATEGORY OF INCIDENT	TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)	"THE MAOIST SHINING PATH GROUP"
6. PERPETRATOR: ID OF ORG(S)	"SHINING PATH" "TUPAC AMARU REVOLUTIONARY MOVEMENT (MRTA)" "THE SHINING PATH"
7. PERPETRATOR: CONFIDENCE	POSSIBLE: "SHINING PATH" POSSIBLE: "TUPAC AMARU REVOLUTIONARY MOVEMENT (MRTA)" POSSIBLE: "THE SHINING PATH"
8. PHYSICAL TARGET: ID(S)	"THE EMBASSIES OF THE PRC AND THE SOVIET UNION"
9. PHYSICAL TARGET: TOTAL NUM	1
10. PHYSICAL TARGET: TYPE(S)	DIPLOMAT OFFICE OR RESIDENCE: "THE EMBASSIES OF THE PRC AND THE SOVIET UNION"
11. HUMAN TARGET: ID(S)	-
12. HUMAN TARGET: TOTAL NUM	-
13. HUMAN TARGET: TYPE(S)	-
14. TARGET: FOREIGN NATION(S)	PRC: "THE EMBASSIES OF THE PRC AND THE SOVIET UNION"
15. INSTRUMENT: TYPE(S)	*
16. LOCATION OF INCIDENT	PERU: SAN ISIDRO (TOWN): LIMA (DISTRICT)
17. EFFECT ON PHYSICAL TARGET(S)	-
18. EFFECT ON HUMAN TARGET(S)	-

ACE Events

Event type	Event subtype
Life	Be-born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-ownership, Transfer-money
Business	Start-org, Merge-org, Declare-bankruptcy, End-org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-write
Personnel	Start-position, End-position, Nominate, Elect
Justice	Arrest-jail, Release-parole, Trial-hearing, Charge-indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Two Tasks

- Slot filling
  - Find values of individual template slots or arguments
- Consolidation
  - Identify slots associated with the same event / template

Hand-crafted patterns

- For terrorist incident
  - Killing of <HumanTarget>
  - Bomb was placed by <Perp> on <PhysicalTarget>
  - <Perp> attacked <HumanTarget>’s <PhysicalTarget> with <Device>
  - <HumanTarget> was injured
- Pattern must specify slot(s) filled
- Pattern may also specify type of filler in cases of ambiguity
  - Target was <person:HumanTarget>

Hand-crafted patterns (2)

- Must allow for syntactic variation
  - Intervening modifiers (between subject and verb)
  - conjunction
- FASTUS approach: syntactic patterns
  - express patterns in terms of noun and verb groups
  - for prepositional phrases:
    - Subject {Preposition NounGroup}* VerbGroup
  - for relative clauses
    - Subject Relative-pronoun {NounGroup | Other} VerbGroup {NounGroup | Other}* VerbGroup
- Parsing approach: build dependency parse,
  state patterns in terms of dependency relations

Supervised Event Extraction

- Multiple classifiers
- Trigger classifier
    - Applied to each noun / verb / adjective
    - Determine if word is a trigger
    - Determine its event type and subtype
    - Typical features: lexical, WordNet, other entities in sentence, their dependency relation to the trigger and their semantic types
- Argument classifier
    - Applied to <trigger word, entity in same sentence>
    - Determine if word is an argument
    - Determine its role
    - Typical features: trigger, event type, dependency relation of entity to trigger

Using Non-local Information

- Local clues may not be sufficient for event classification:
    - He left Microsoft that afternoon.
  - A trip? A resignation?
- Information from broader scope can help
  - Use bag-of-words classifier applied to sentence as feature
  - Use other events in document as feature
  - Run document topic classifier, use document topics as features

Consolidation

- For individual ACE event mentions, consolidation is a form of coreference
  - Construct similarity mention based on
    - Trigger words
    - Shared or conflicting arguments
    - Distance
  - Cluster event mentions
  - Unfortunately tagging of event mentions is not reliable enough to support effective coreference
- For larger templates
  - If components are largely contiguous, can treat consolidation as a text segmentation task
  - Label sentences as BIO-segment
  - Based on
    - Slots already filled in a segment
    - Shared or conflicting slots

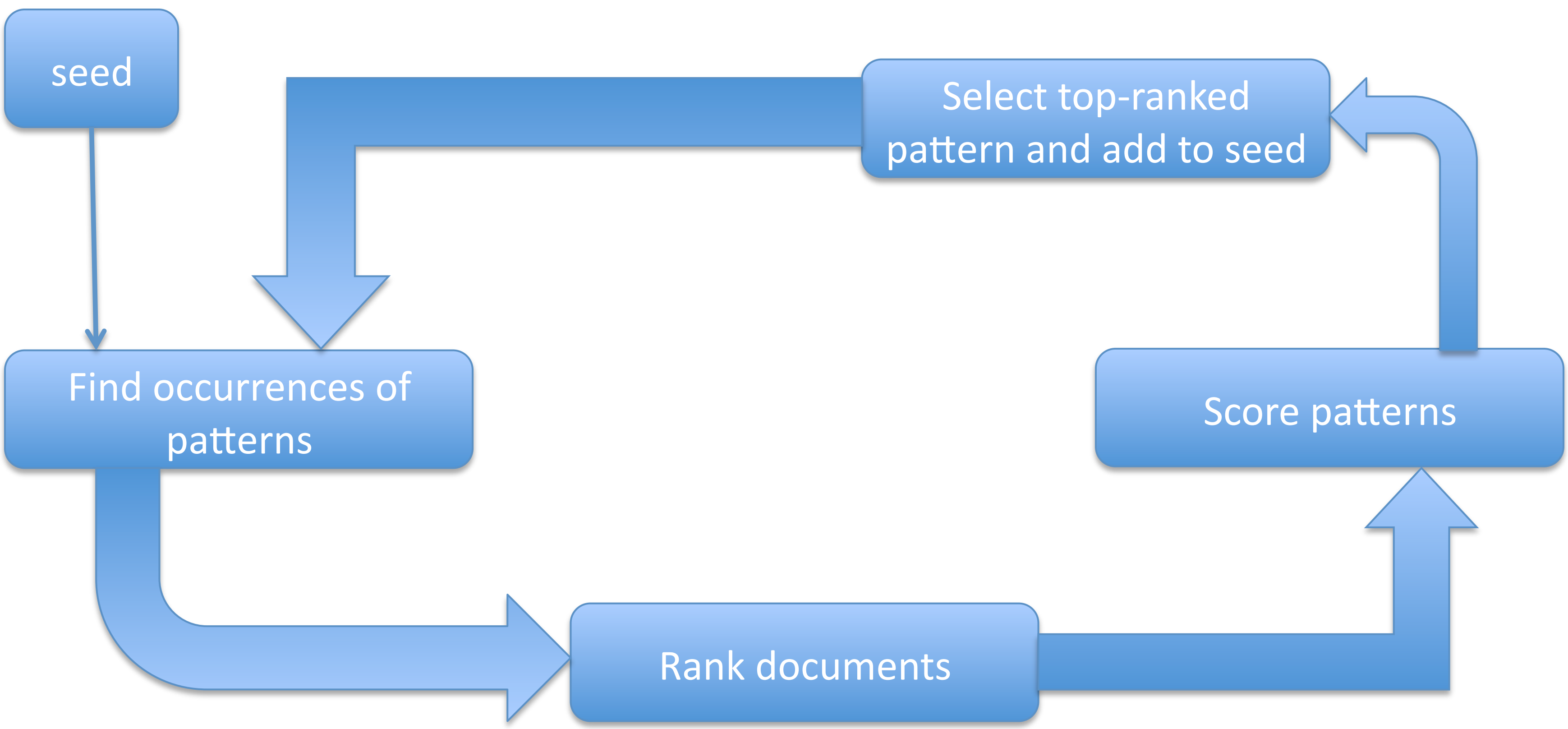
Semi-supervised models (1)

- Goal:
  - find event patterns relevant to a specific topic
- Approach:
  - mark relevant documents in corpus
  - extract all single-slot patterns in corpus
  - for each pattern P compute score

$$\frac{frequency\_in\_relevant\_documents}{frequency\_in\_corpus} \times \log(frequency\_in\_relevant\_documents)$$

  - patterns with high score are good candidates: top 5 for the MUC terrorist corpus …
    - (subj) exploded
    - murder of (np)
    - assassination of (np)
    - (sujb) was killed
    - (subj) was kidnapped

Semi-supervised models (2)



Semi-supervised models (3)

To make this into a bootstrapping procedure:

- Start with seed patterns
- Mark documents containing patterns as ‘relevant’

Repeat

- Score patterns
  - » Based on (relev. freq / total freq) * log(relev. freq)
- Add top-ranked pattern to seed
- Recompute relevance of documents
  - » Relevance graded … between 0 and 1

Semi-supervised models (3)

- Problems:
  - Semantic drift
    - documents containing event type X also contain event type Y
  - Stopping point
    - Eventually all documents are marked relevant
- Solution: competitive bootstrapping
  - Identify all major topics in corpus
  - Create seed for each topic
  - Train patterns for all topics concurrently
    - Assume topics are mutually exclusive

Semi-supervised models (4)

- Using co-training:
  - Treat this as a document classification task with two classifiers
    - C_1 = pattern-based classifier
    - C_2 = bag-of-words-based classifier
  - Yields consistent improvement over using pattern-based classifier alone [Surdeanu et al. 2006]

Evaluation

- Multiple events with multiple arguments
  - Many possible alignments
- Unified evaluation score
  - Penalties for each type of mismatch
    - Missing event / spurious event / event type error
    - Missing argument / spurious argument / role error
  - Search for best alignment
    - Potentially large search
- Separate scores for events and arguments
  - Score events based on <trigger word, event type> pairs
  - Score arguments based on <event type, role, argument> triples
    - Scores for both based on recall / precision / F-measure

Course Outline

- Machine learning preliminaries
- Name extraction
- Entity extraction
- Relation extraction
- Event extraction
- Other domains

Good candidates for IE

- Large volume of text

- Common set of high-frequency semantic relations

- Strong incentive for
    - Search
    - Data base construction
    - Data mining

which involves entity attributes or relations between entities

Good candidates for IE

- General and business news
- Medical records
  - Hospitals generate a large number of text documents
    - Some of narrow scope, such as radiology reports
    - Some of wide scope, such as discharge summaries
- Scientific papers
  - Rapid growth of medical and biomedical literature
    - PubMed adds 500,000 entries per year
  - Focus of NLP for last decade on genomics literature
    - Large resources assembled (e.g., GENIA project in Tokyo)

News IE Demos

- Europe Media Monitor NewsExplorer
  - http://emm.newsexplorer.eu/NewsExplorer/home/en/latest.html

- OpenCalais
  - http://viewer.opencalais.com/

Medical Record IE

- A critical application
  - timely access to patient information
  - collect diagnosis / treatment / outcome statistics
  - currently much info is encoded by hand
  - encouraged by push for Electronic Health Records
- Impediments
  - data is sensitive, must be anonymized
  - hospitals build their own electronic records
    - » makes sharing difficult
  - standard test sets & evaluations only in last few years
    - » medication extraction in 2009
    - » discharge summary analysis in 2010

Sample Discharge Summary analysis

The patient is a 64-year-old male with a long standing history of peripheral vascular disease who has had multiple vascular procedures in the past including a fem-fem bypass , a left fem pop as well as bilateral TMAs and a right fem pop bypass who presents with a nonhealing wound of his left TMA stump as well as a pretibial ulcer that is down to the bone . The patient was admitted to obtain adequate pain control and to have an MRI / MRA to evaluate any possible bypass procedures that could be performed .

- c="peripheral vascular disease" 1:12 1:14||t="problem"
- c="multiple vascular procedures" 1:18 1:20||t="treatment"
- c="a fem-fem bypass" 1:25 1:27||t="treatment"
- c="a left fem pop" 1:29 1:32||t="treatment"
- c="bilateral tmas" 1:36 1:37||t="treatment"
- c="a right fem pop bypass" 1:39 1:43||t="treatment"
- c="a pretibial ulcer" 1:58 1:60||t="problem"
- c="adequate pain control" 2:6 2:8||t="treatment"
- c="an mri / mra" 2:12 2:15||t="test"
- c="a nonhealing wound of his left tma stump" 1:47 1:54||t="problem"
- c="bypass procedures" 2:20 2:21||t="treatment"

Medical IE Demo

- Extracting information about medication (2009 shared task)

  – http://code.google.com/p/lancet

Bio-IE

- Bio-NER: challenging named entity tasks for proteins, genes, chemicals, etc.
  - Large variation in name structures
  - Difficulty of identifying name boundaries
  - Feature set quite different from names in the news
    - prefix and suffix strings
    - 'shape' features
  - Multiple names for same gene or protein
  - Ambiguous abbreviations (context-dependent)
  - Now F in 80's for protein names (JNLPBA task)

- Sample sentence for JNLPBA task

We have shown that
<cons sem=”G#protein”>interleukin-1</cons>
(<cons sem=”G#protein”>IL-1</cons>) and
<cons sem=”G#protein”>IL-2</cons> control
<cons sem=”G#DNA”>IL-2 receptor alpha (IL-2R alpha)
gene</cons> transcription in
<cons sem=”G#cell line”>CD4-CD8- murine T lymphocyte
precursors</cons>.

Bio-IE (2)

- Bio-IE tasks are motivated by the databases which are currently being curated by hand from journal articles
- PPI – protein-protein interaction
  - cellular processes generally involve interaction of two or more proteins
  - large and rapidly growing database
    - MINT: 240,000 interactions of 35,000 proteins
  - first Bio-IE shared tasks aimed to capture these interactions (LLL (2005), BioCreative (2007))
  - intensively studied by Bio-NLP groups using methods described for relation extraction (feature & kernel-based methods)
- More recent Bio-NLP tasks are aimed at more detailed event information involving proteins

Biomedical IE Demo

- Biomedical NER
  - http://nlp.i2r.a-star.edu.sg/demo_bioner.html

Closing Thoughts

- Unsupervised learning
- Estimating confidence
- Variations in corpora
- Obstacles and performance limits

Unsupervised learning

- Until now we have assumed that we have a specific extraction goal: to identify a specific relation or fill a predefined template
- But when we get texts in a new domain we may be explorers:
  we want to know what the major relations (or larger semantic structures) are for the new domain

Unsupervised extraction

- Unsupervised relation extraction
  - Essentially a clustering procedure [Hasegawa et al 2002]
    - For a given pair of argument types
    - Group triples <arg1, context, arg2> based on lexical similarity of contexts and shared argument pairs
    - Efficient clustering for web-scale tasks
    - Identify argument classes
- Unsupervised template construction
  - Gather documents about same event, and then about same type of event; collect shared predicates [Shinyama et al. 2006]

Evaluating unsupervised extraction

- Compare against "gold standard"
  - problem: there may be several 'right answers'
  - problem: gold standard may be very large
- Evaluate manually the clusters produced by the system
  - judge consistency (precision) and completeness (recall) of clusters
  - problem: must repeat after each system revision
  - problem: hard to judge recall … find everything the system missed
- Use clusters as features for supervised training
  - result depends on final task

The Unsupervised and the Semi-supervised

Unsupervised search can play another role …

- The results of unsupervised search can inform semi-supervised search
    - For word classes [McIntosh 2010]
    - For relations [Sun 2010]
  - Gives structure to the space being searched

Estimating Confidence

- A crucial part of semi-supervised extraction is confidence estimation
- Is this information useful directly?
  - Can we create a probabilistic data base?

Variations in Corpora

- IE components may be much more sensitive to changes in corpora than one expects
    - test scores are really test scores *on a particular corpus*
    - a name tagger which gets mid-80's F-score on general news may drop to mid-60's on terrorist reports
    - an event tagger trained on news stories will do very poorly on the sports section
  - need (semi-supervised) methods to adapt to new sources and topics
  - need topic models to capture broad context

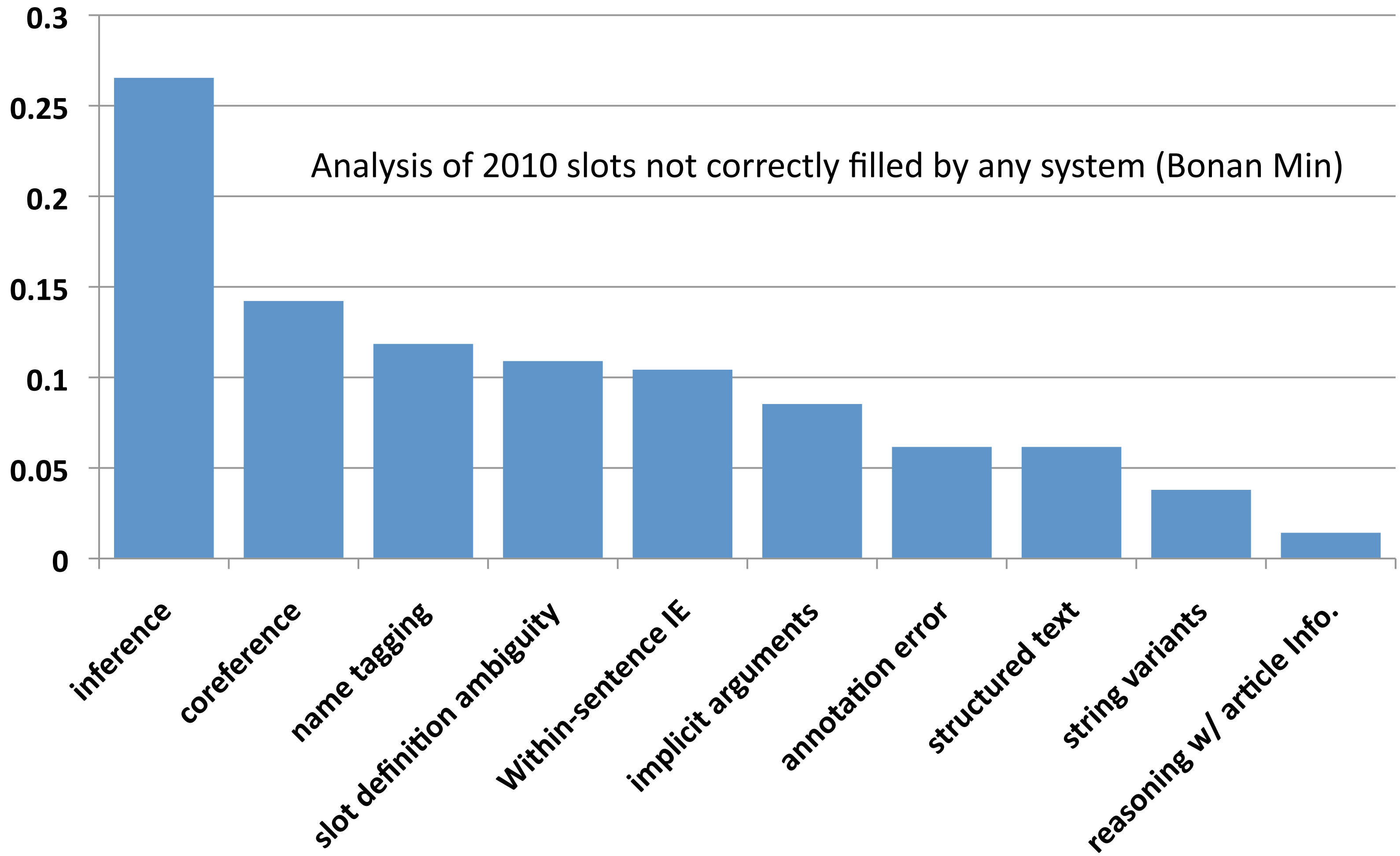
Obstacles to better performance

- Coreference and implicit relations
- The pipeline problem
- Need for deep reasoning

- In our course, we have emphasized the problem of coverage (paraphrase discovery)

- This is important, but not necessarily the dominant problem in an IE system

Many Sources of Error in KBP Slot Filling task



Coreference

- As we have discussed, the mention directly involved in a relation or event is often not the name mention we need to report
- So coreference errors are a major limitation on extraction performance
  - Particularly errors from nominal anaphors
- Implicit reference is also common and not frequently handled

Some coreference examples

Nominal coreference

- *A woman charged with running a prostitution ring in the U.S. capital city made….In court records, prosecutors estimate that her business, Pamela Martin and Associates, generated more...*
- *the alleged prostitution outfit, known as Pamela Martin and Associates, that she is accused of running by phone out of her homes in Vallejo and Escondido, Calif. ...The operation, …*

Implicit argument

- *National Museum of Women in the Arts*
  … Judy L. Larson, formerly of the Art Museum of Western Virginia, has served as a director
  [of ____] since 2002.

The Pipeline Problem

- IE systems are generally organized as pipelines …
    - Name recognition
    - Parsing
    - Coreference
    - Relation and event extraction
  - simple, efficient, modular structure
- Each may be quite good, but each depends on all its predecessors
  - If each introduces 10% error, we may have 40-50% error at the end of the pipeline
- Effect can be mitigated by *joint inference*
  - For example, joint inference of name and relation extraction
    - Prefer name types consistent with relations
  - Reduces errors somewhat but at cost of large search space

Deep reasoning

- Our general strategy has been to address the wide variety of ways in which a relation or event may be expressed by gathering evermore patterns or features
- But at some point there is a remnant for which such shallow matching does not suffice … deeper reasoning is needed
  - perhaps another NLP paradigm shift will be needed

- Meanwhile there are many valuable applications of IE which do not require 100% performance

For More Information

grishman@cs.nyu.edu