6.891: Lecture 21 (November 24th, 2003)

Relation Extraction

Overview

- A supervised method for relation extraction
  (an extension of statistical parsing methods)

- Two partially supervised methods:
  ([Brin, 1998], and [Agichtein and Gravano, 2000])

- A sketch of a cotraining approach for the partially-supervised task

Information Extraction

Named Entity Recognition

INPUT: Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT: Profits soared at [Company Boeing Co.], easily topping forecasts on [Location Wall Street], as their CEO [Person Alan Mulally] announced first quarter results.

Relationships between Entities

INPUT: Boeing is located in Seattle. Alan Mulally is the CEO.

OUTPUT:

{Relationship = Company-Location
Company = Boeing
Location = Seattle}

{Relationship = Employer-Employee
Employer = Boeing Co.
Employee = Alan Mulally}

Extraction From Entire Documents

Hi [PERSON Ted] and [PERSON Hill],
Just a reminder that the game move will need to be entered [TIME tonight]. We will need data on operations, rawmaterials ordering, and details of the bond to be sold.
[PERSON Hill]: I will be in the [LOCATION lobby] after the class at [TIME 9 pm]. how about we meet in the [LOCATION lobby] around that time (i.e when both our classes are over).
[PERSON Ted]: Let me know how you are going to provide the bond related input information. We can either meet in the [LOCATION lobby] around [TIME 5.30 pm] or you can e-mail me the info.
Thanks, [PERSON Ajay]

TIME	9 pm, 18th September	TIME	5.30 pm, 18th September
LOCATION	Lobby, Building NE43	LOCATION	Lobby, Building NE43
PERSON	David Hill, Ajay Sinclair	PERSON	Ted Jones, Ajay Sinclair
TOPIC	data on operations. . .	TOPIC	bond related input information

10TH DEGREE is a full service advertising agency specializing in direct and interactive marketing. Located in Irvine CA, 10TH DEGREE is looking for an Assistant Interactive Account Manager to help manage and coordinate interactive marketing initiatives for a marquee automotive account. Experience in online marketing, automotive and/or the advertising agency field is a plus.

Assistant Account Manager Responsibilities

Ensures smooth implementation of programs and initiatives Helps manage the delivery of projects and key client deliverables ...

Compensation: $50,000 – $80,000 Hiring Organization: 10TH DEGREE

Principals only. Recruiters, please don't contact this job poster. Please, no phone calls about this job! Please do not contact job poster about other services, products or commercial interests. Reposting this message elsewhere is NOT OK. this is in or around Orange County - Irvine

⇓

INDUSTRY	Advertising
POSITION	Assistant Account Manager
LOCATION	Irvine, CA
COMPANY	10th Degree
SALARY	$50,000 – $80,000

Relationship Extraction

[Miller et. al, 2000]

An example:

Donald M. Goldstein, a historian at the University of Pittsburgh . . .

- Entity information to be extracted:
  - Named entity boundaries:
    Organizations, people, and locations
  - Person descriptors: “a historian at the University of Pittsburgh” refers to “Donald M. Goldstein”
- Entity relationships to be extracted:
  - Employer/Employee relations
    (e.g., *Goldstein* is employed at *University of Pittsburgh*)
  - Company/product relations
  - Organization/headquarters-location relation

Relationship Extraction: Annotation

Another example:

Nance, who is a paid consultant to ABC News, said . . .

- The following information was annotated:
  - *Nance* as a person; *ABC News* as an organization; *a paid consultant to ABC News* as a descriptor
  - A *coreference* link between *Nance* and *a paid consultant to ABC News*
  - An *employer-relation* link from *a paid consultant to ABC News* to *ABC News*

Next question: how can we build a model which recovers this information?

The Basic Approach

- Build a statistical parsing model which simultaneously recovers syntactic relation and the information extraction information

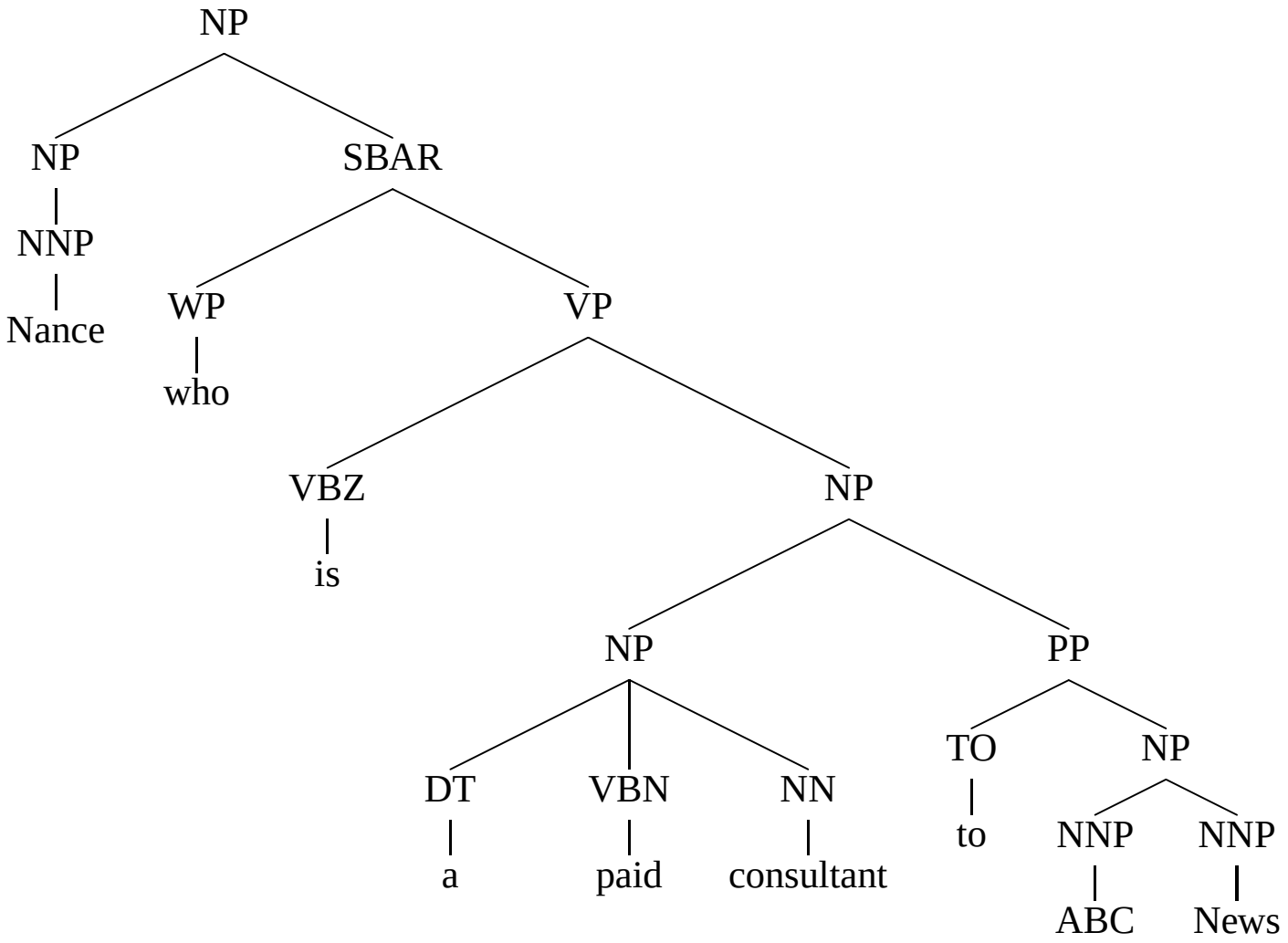
  To do this:

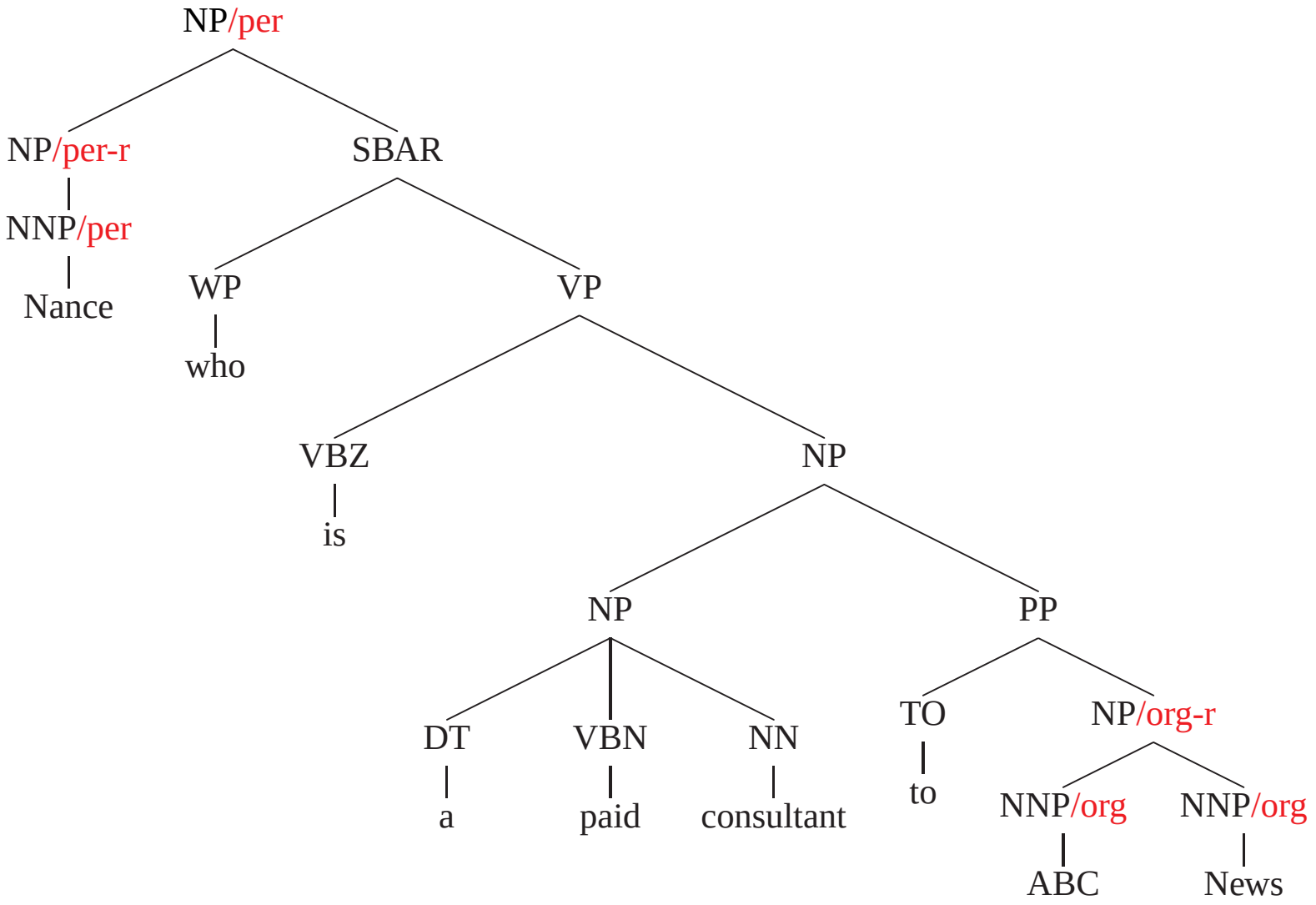
- Step 1: annotate training sentences for entities, descriptors, coreference links, and relation links

- Step 2: train a parser on the Penn treebank, and apply it to the new training sentences. **Force the parser to produce parses that are consistent with the entity/descriptor etc. boundaries**

- Step 3: enhance the parse trees to include the information extraction information (we'll come to this soon)

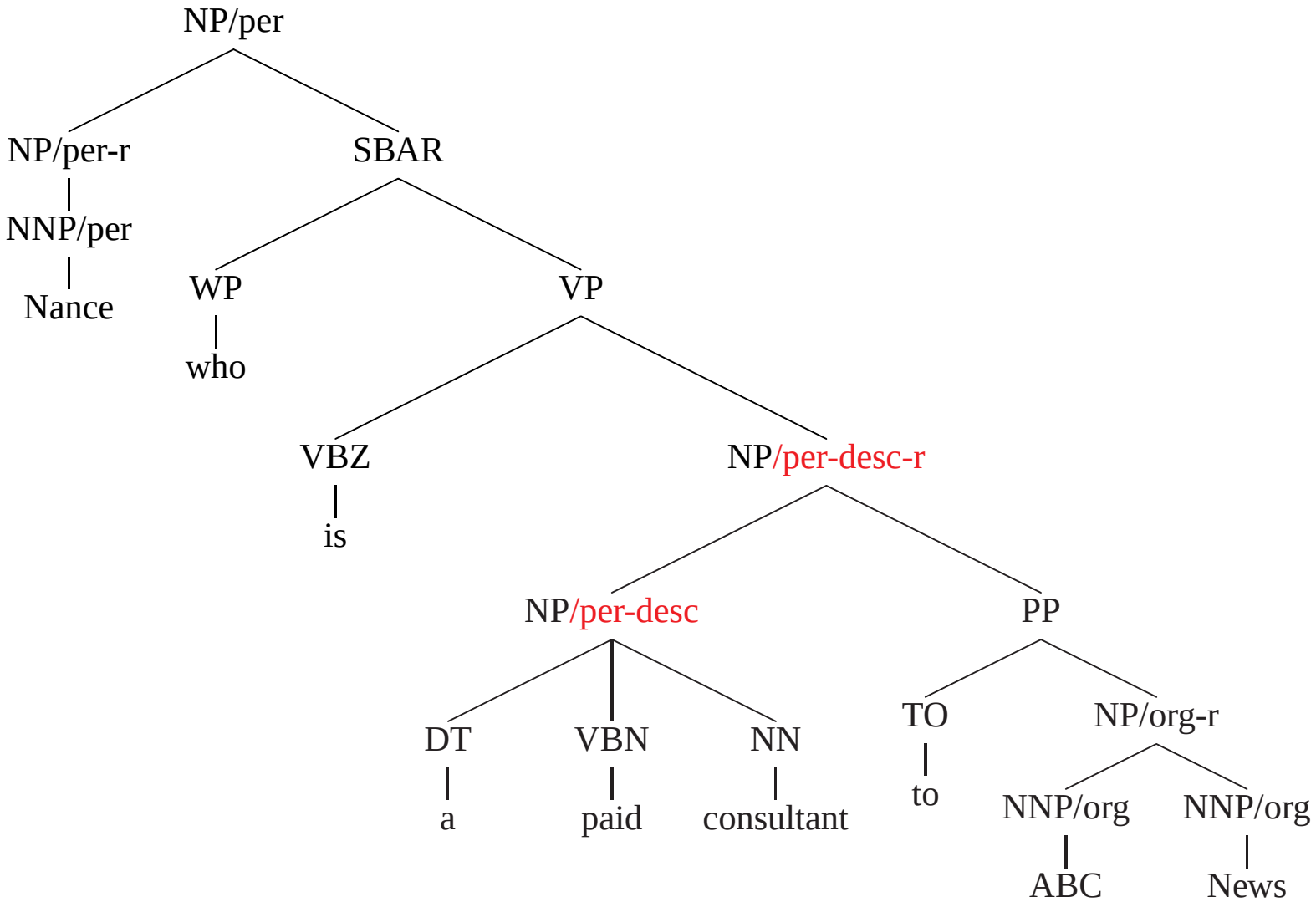
- Step 4: **re-train** the parser on the new training data, and with the new annotations


NP
NP
SBAR
NNP
WP
VP
Nance
who
VBZ
NP
is
NP
PP
DT
VBN
NN
TO
NP
a
paid
consultant
to
NNP
NNP
ABC
News



Add semantic tags showing named entities

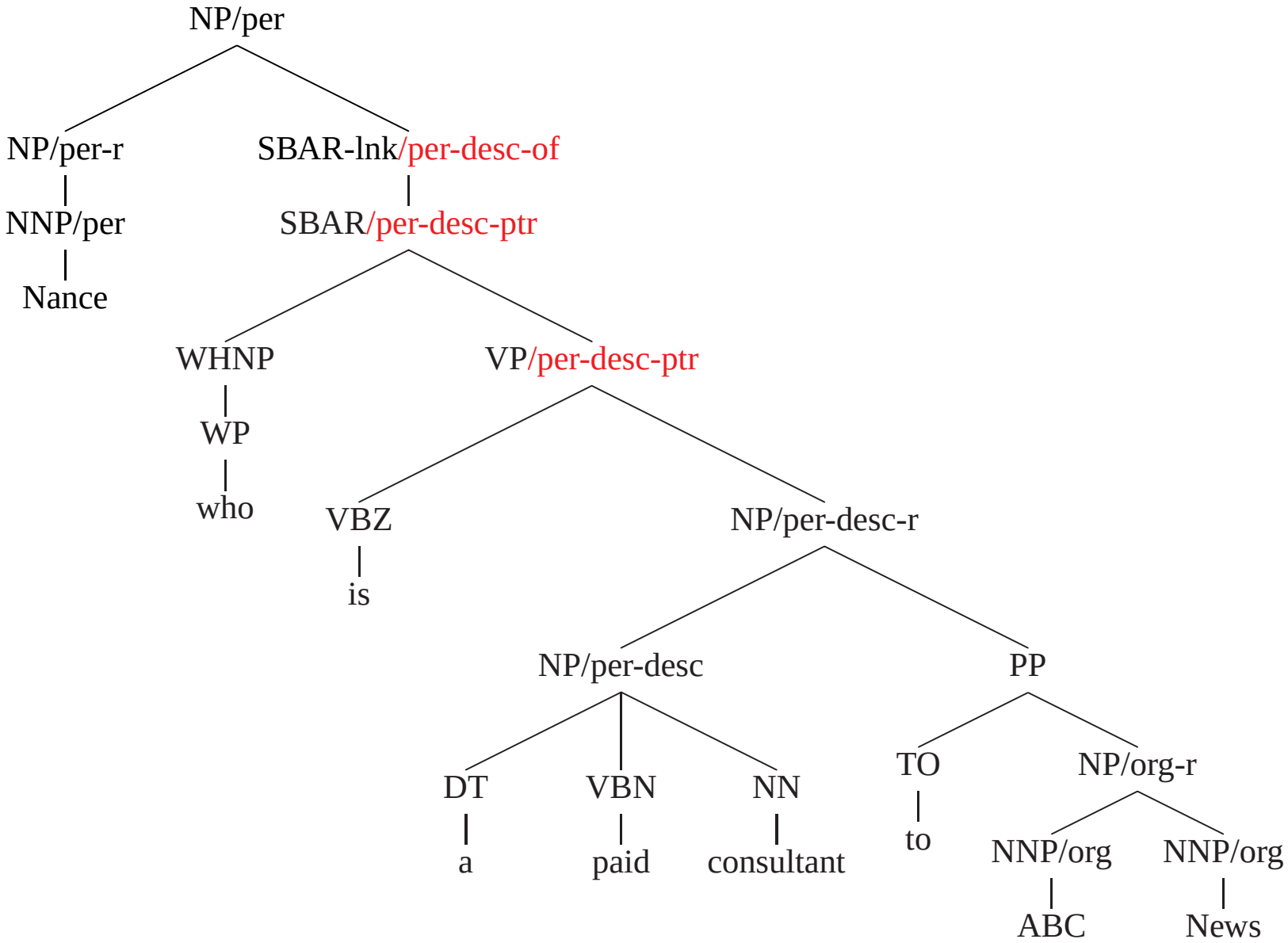
org = organization, per = person, org-r = organization “reportable” (complete), per-r = person “reportable” (complete)



Add semantic tags showing descriptors

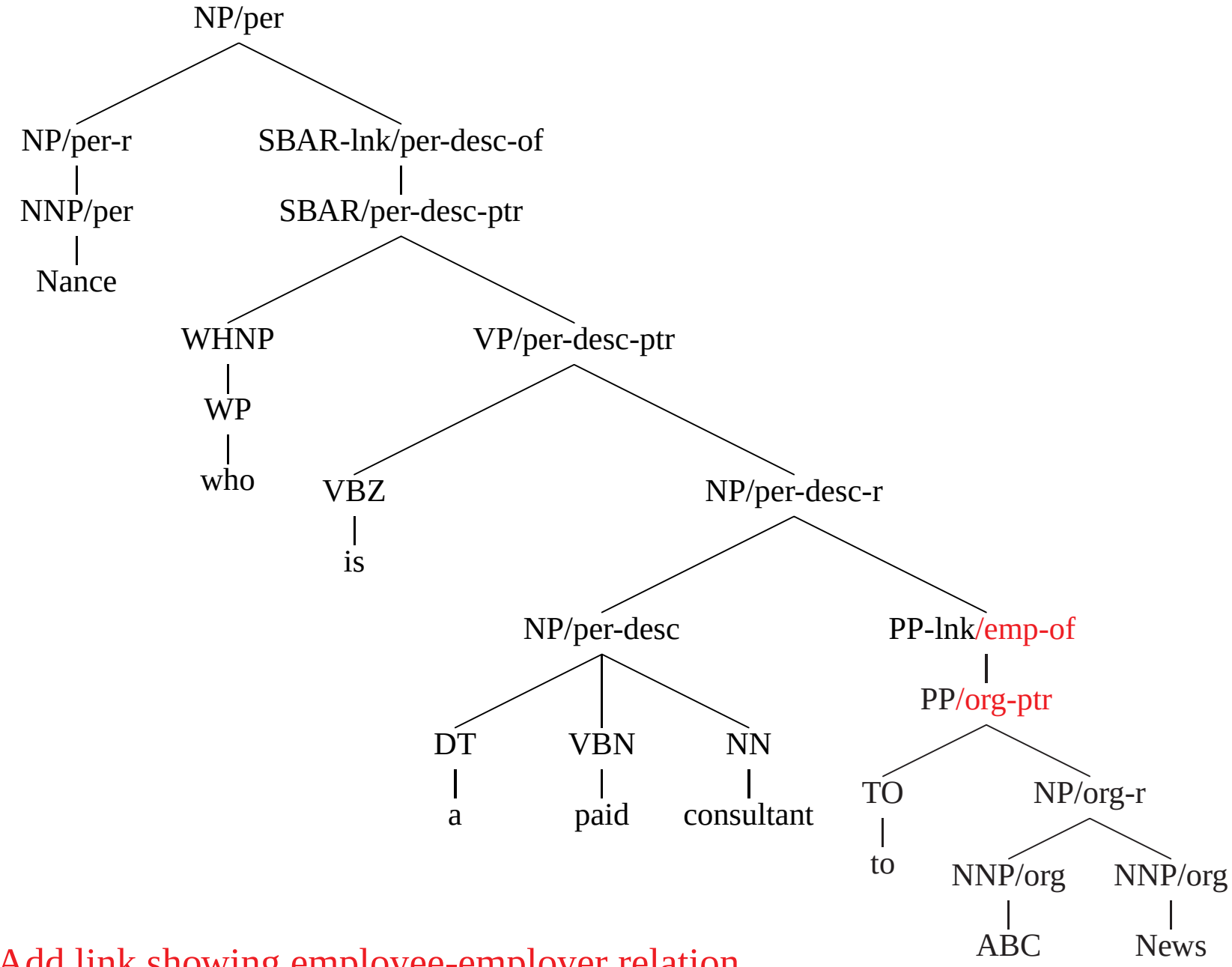
per-desc = person descriptor,

per-desc-r = person descriptor “reportable” (complete)



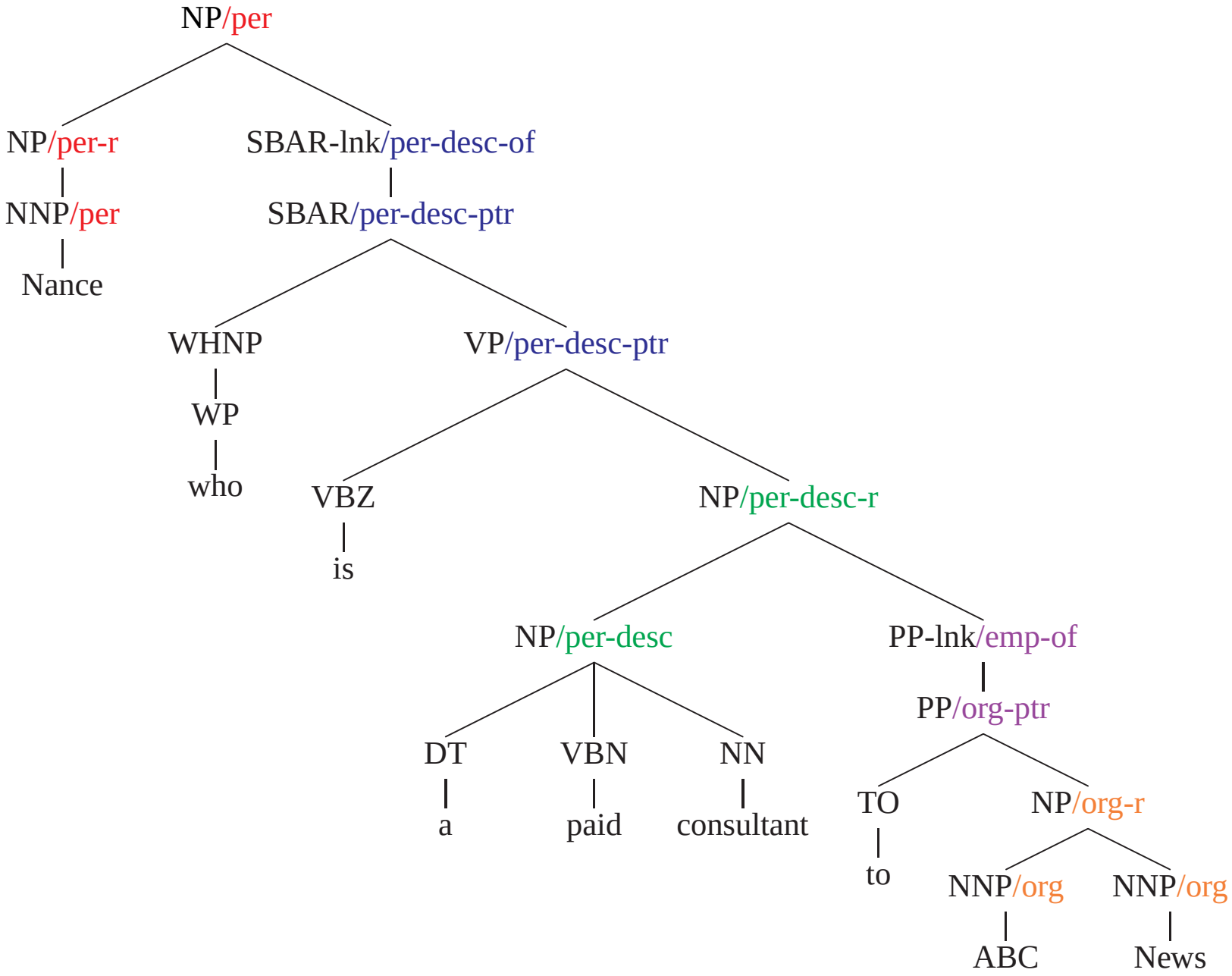
Add semantic tags showing link between “Nancy” and the descriptor

per-desc-of = person/descriptor link, per-desc-ptr = person/descriptor pointer



Add link showing employee-employer relation

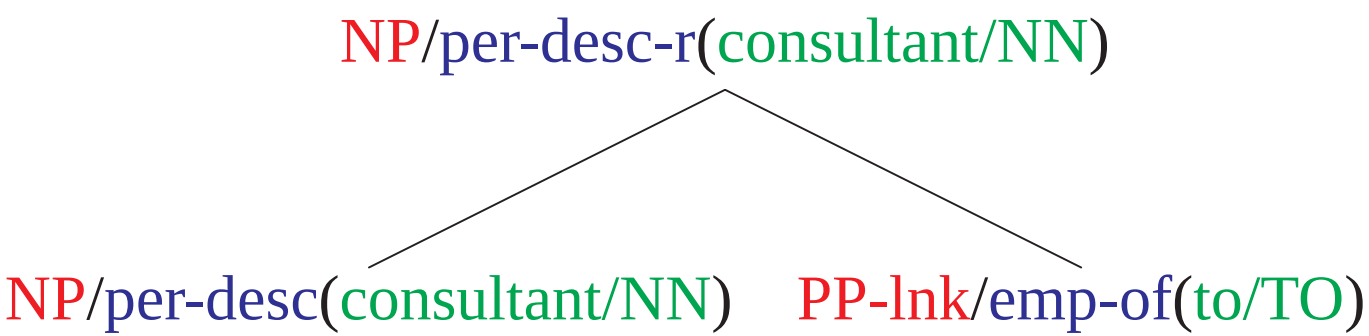
emp-of = employee-of link, emp-ptr = employee-of pointer



PERSON entity, PERSON descriptor link, DESCRIPTOR, EMPLOYER-OF relation, ORG entity

Building a Parser

- We now have context-free rules where each non-terminal in the grammar has
  - A syntactic category
  - A semantic label
  - A head-word/head-tag



- It's possible to modify syntactic parsers to estimate rule probabilities in this case

Modeling Rule Productions as Markov Processes

- Step 1: generate category of head child

NP/per-desc-r(consultant/NN)

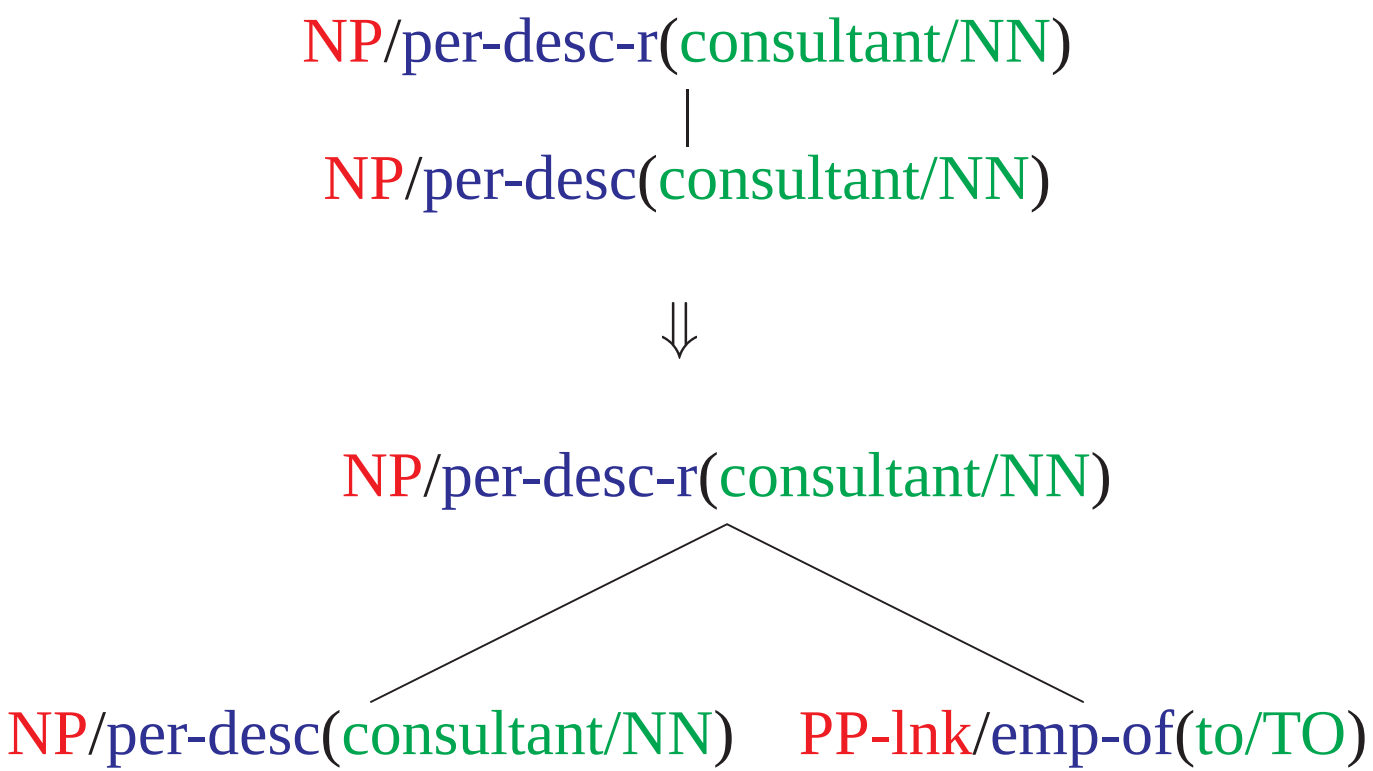
$\Downarrow$

NP/per-desc-r(consultant/NN)
|
NP/per-desc(consultant/NN)

$P_h(\text{NP/per-desc, NN} \mid \text{NP/per-desc-r})$

Modeling Rule Productions as Markov Processes

- Step 2: generate right modifiers in a Markov chain



$P_h(\text{NP/per-desc, NN} \mid \text{NP/per-desc-r}) \times$
$P_d(\text{PP-lnk/emp-of(to/TO)} \mid \text{NP/per-desc-r, consultant/NN, NP/per-desc})$

Summary

- Goal: build a parser that recovers syntactic structure, named entities, descriptors, and relations

- Annotation: mark entity boundaries, descriptor boundaries, links between entities and descriptors

- Enriched parse trees: given annotation, and a parse tree, form a new **enriched** parse tree

- The statistical model: non-terminals now include syntactic category, semantic label, head word, head tag. Rule probabilities are estimated using similar methods to syntactic parsers

- Results: precision = 81%, recall = 64% in recovering relations (employer/employee, company/product, company/headquarters-location)

Overview

- A supervised method for relation extraction
  (an extension of statistical parsing methods)
- Two partially supervised methods:
  ([Brin, 1998], and [Agichtein and Gravano, 2000])
- A sketch of a cotraining approach for the partially-supervised task

Partially Supervised Approaches to Relation Extraction

- Last lecture: introduced a partially supervised method for named entity classification
- Basic observation: "redundancy" in that either spelling or context of an entity is often sufficient to determine its type
- Lead to *cotraining* approaches, where two classifiers bootstrap each other from a small number of seed rules
- **Can we apply these kind of methods to relation extraction?**

From [Brin, 1998]

The World Wide Web provides a vast source of information of almost all types, ranging from DNA databases to resumes to lists of favorite restaurants. However, this information is often scattered among many web servers and hosts using many different formats. If these chunks of information could be extracted from the World Wide Web and integrated into a structure form, they would form an unprecedented source of information. It would include the largest international directory of people, the largest and most diverse databases of products, the greatest bibliography of academic works, and many other useful resources.

From [Brin, 1998]

> For data we used a repository of 24 million web pages totalling 147 gigabytes. This data is part of the Stanford WebBase and is used for the Google search engine [BP], and other research projects. As a part of the search engine, we have built an inverted index of the entire repository.
>
> The repository spans many disks and several machines. It takes a considerable amount of time to make just one pass over the data even without doing any substantial processing. Therefore, in these [sic] we only made passes over subsets of the repository on any given iteration.

[BP] Sergey Brin and Larry Page. Google search engine.
`http://google.stanford.edu`

Two Examples

- From [Brin, 1998]:
  authors/book-titles, data = web data, seeds are

Isaac Asimov	The Robots of Dawn
David Brin	Startide Rising
James Gleik	Chaos: Making a New Science
Charles Dickens	Great Expectations
William Shakespeare	The Comedy of Errors

- From [Agichtein and Gravano, 2000]:
  companies/head-quarter locations, data = text, seeds are

Microsoft	Redmond
Exxon	Irving
IBM	Armonk
Boeing	Seattle
Intel	Santa Clara

DIPRE [Brin, 1998]

A pattern is a 5 tuple:

- *Order*: author preceding title, or visa versa
- *URL-prefix*: a prefix of the URL of the page of the pattern
- *prefix*: up to 10 characters preceding the author/title pair
- *middle*: the characters between the author and title
- *suffix*: up to 10 characters following the author/title pair

DIPRE: Inducing Patterns from Data

- Find all instances of seeds on web pages.
  Basic question: how do we induce patterns from these examples?

- Answer = Following procedure:

1. Group all occurrences together which have the same values for *order*, *middle*
2. For any group: Set *url-prefix* to be longest common prefix of the group's URLs, *prefix* to be the longest common prefix of the group, *suffix* to be the longest common suffix
3. For each group's pattern, calculate its specificity as

$$spec(p) = n|middle||url\text{-}prefix||prefix||suffix|$$

   where n is the number of examples in the group, $|x|$ is the length of x in characters
4. **If** specificity exceeds some threshold, include the pattern
5. **Else If** all patterns occur on same webpage, reject the pattern
6. **Else** create new sub-groups grouped by characters in the urls which is one past *url-prefix*, and repeat the procedure in step 2 for these new sub-groups.

The Overall Algorithm

1. Use the seed examples to label some data

2. Induce patterns from the labeled examples, using method described on the previous slide

3. Apply the patterns to data, to get a new set of author/title pairs

4. Return to step 2, and iterate

DIPRE: Inducing Patterns from Data

The patterns found in the first iteration:

`www.sff.net/locus/c.*`	<LI><B>*title*</B> by *author* (
`dns.city-net.com/lmann/awards/hugos/1984.html`	*title* by *author* (
`dolphin-upenn.edu/dcummins/texts/sf-award.htm`	*author* || *title* || (

- The 5 seeds produced 199 labeled instances, giving the 3 patterns above
- Applying the three patterns gave 4047 new book instances
- Searching 5 million web pages gave 3972 occurrences of these books
- This gave 105 patterns, 24 applied to more than one URL
- Applied to 2 million URLS produced 9369 unique (author,title) pairs
- Manual intervention: removed 242 “bogus” items where the author was “Conclusion”
- Final iteration: ran over 156,000 documents which contained the word “books”; induced 346 patterns, 15,257 (author,title) pairs

SNOWBALL [Agichtein and Gravano, 2000]

- Some minor differences: Task = organization/headquarters-location; Data = regular text (not web pages); Feature-vector representation of prefix/middle/suffix.

- An important difference:
  - They make the observation that they are learning a *function* (each organization has just one location), rather than a more general relation
  - They use this to define a measure of quality of a newly-proposed pattern P:

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

    were $P.positive$ is number of times the new pattern recovers an org/location pair seen at a previous iteration of training; and $P.negative$ is number of times it recovers a pair where the organization has been seen with a different location at a previous iteration

Overview

- A supervised method for relation extraction
  (an extension of statistical parsing methods)
- Two partially supervised methods:
  ([Brin, 1998], and [Agichtein and Gravano, 2000])
- A sketch of a cotraining approach for the partially-supervised task

Formalizing these Methods Using Cotraining?

- The methods are quite heuristic (little formal justification)
- Some important insights:
  - *Redundancy*: if you see an author/title pair on a web page which you know are related, most likely the web page contains a pattern expressing the same relation
  - *Learning functions may be easier:* restriction that each organization has just one location may allow us to obtain "negative" examples
- Can we formulate the task in a similar way to cotraining?

Recap: An Approach Using Minimal Supervision

- Assume a small set of "seed" rules

contains(Incorporated)	$\Rightarrow$	Organization
full-string=Microsoft	$\Rightarrow$	Organization
full-string=I.B.M.	$\Rightarrow$	Organization
contains(Mr.)	$\Rightarrow$	Person
full-string=New_York	$\Rightarrow$	Location
full-string=California	$\Rightarrow$	Location
full-string=U.S.	$\Rightarrow$	Location

- Assume a large amount of unlabeled data

.., says Mr. Cooper, a vice president of ...

- Methods gain leverage from redundancy:

  Either Spelling or Context alone is often sufficient to determine an entity's type

Recap: Cotraining

- We have domains $\mathcal{X}$, $\mathcal{Y}$
- We have **labeled** examples (x_i, y_i) for $i = 1 \ldots n$
- We have **unlabeled** examples (x_i) for $i = (n+1) \ldots (n+m)$
- We assume each example x_i splits into two views, x_{1i} and x_{2i}
- e.g., if x_i is a feature vector in $\mathbb{R}^{2d}$, then x_{1i} and x_{2i} are representations in $\mathbb{R}^d$.

Recap: Cotraining Summary

- $n + m$ training examples $x_i = (x_{1i}, x_{2i})$
- First n examples have labels y_i
- Learn functions F_1 and F_2 such that

$$F_1(x_{1i}) = F_2(x_{2i}) = y_i \qquad i = 1 \ldots n$$

$$F_1(x_{1i}) = F_2(x_{2i}) \qquad i = n + 1 \ldots n + m$$

Formalizing Partially-Supervised Relation Extraction

Task: for any organization in a document, identify the location which is its headquarters, or output *NULL* if no location in the document is its headquarters.

Much of the computer services industry remains healthy in spite of stock declines. With the move of the Boeing headquarters to Chicago, prospects for aerospace are less certain, but the company has continued to emphasize its investment in the Puget Sound area.
As the 15th largest city in Washington, Redmond continues to be dominated by high-tech industry and manufacturing, its steady and sound economic business base. Yet Redmond is also gaining recognition as a hot place to do business for many types of industries, from apparel to insurance, from start-ups and small businesses to decades-old institutions.
Microsoft continues to be the economic powerhouse anchoring Redmond. Its 52,163 employees outnumber the city population of 45,256. This illustrates that Microsoft has lots of prominent company.
Nintendo of America also makes its headquarters here. In 2001, Nintendo launched its highly anticipated Game Boy Advance, selling one million units in the US market in just six weeks. In November 2001, Nintendo launched the GameCube home video console, smashing all previous US sales records, and becoming the fastest-selling next generation hardware system.

- For Boeing, choose btwn Chicago, Puget Sound, Washington, Redmond, US, NULL
- For Microsoft, choose btwn Chicago, Puget Sound, Washington, Redmond, US, NULL
- For Nintendo of America, choose btwn Chicago, Puget Sound, Washington, Redmond, US, NULL

Two Views of the Data

- Each *instance* x is an organization/location pair, in context
  With the move of the Boeing headquarters to Chicago, . . .
- First view: *spelling* = (org,location) pair
- Second view: *context* = any features of the surrounding document
- Learning method for the first view:
  rote learning (look-up table mapping companies to locations)
- Learning method for the second view:
  feature-vector representation of the context, together with a global linear model