Information Extraction: foundations and rule-based approaches

Brendan O'Connor

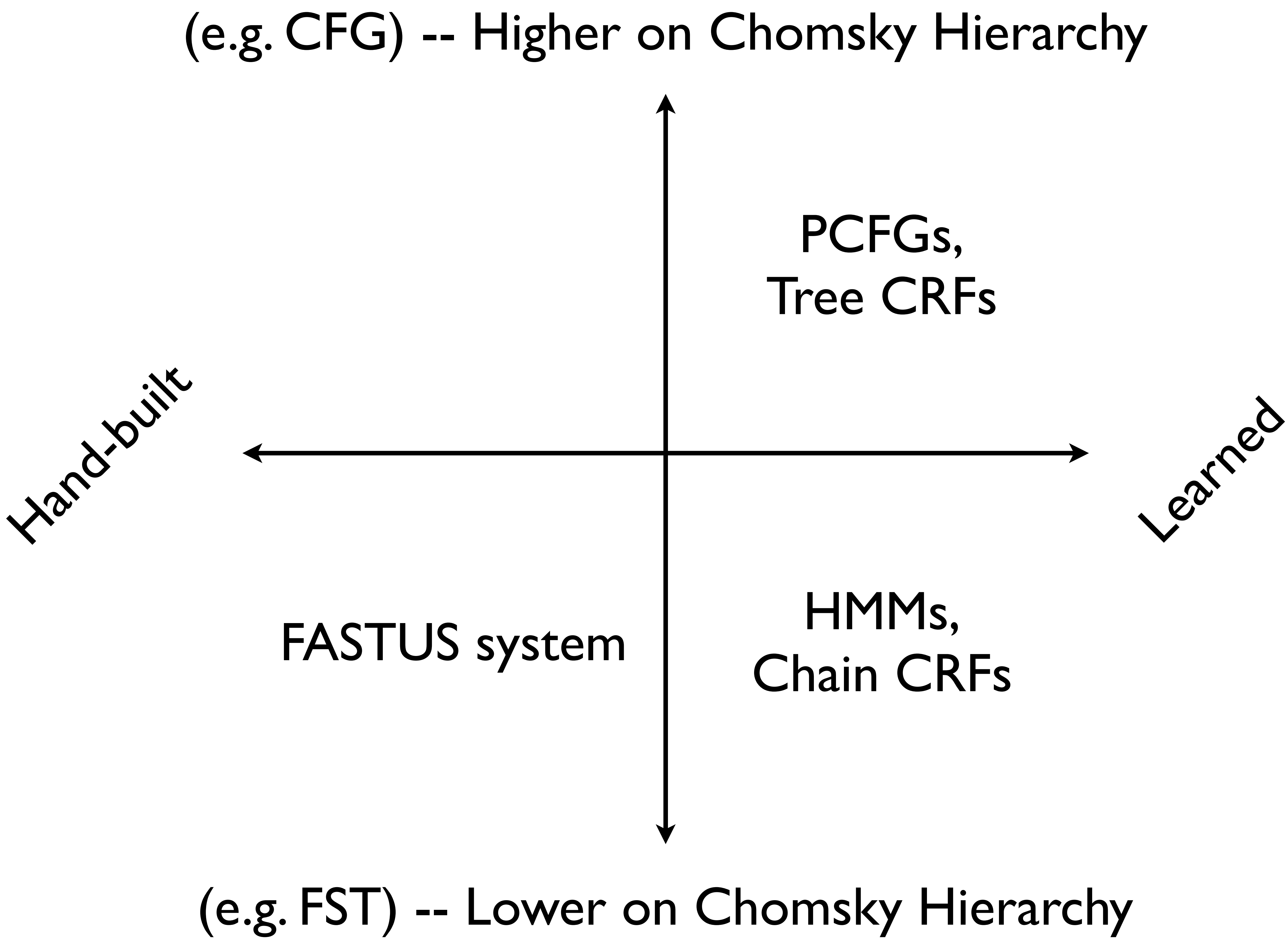
Structured Prediction, 11/17/2011

Outline

- Problem
  - Theoretical foundations: frames and scripts
  - The template-filling paradigm
- Early methods
  - Rule-based
  - Rule-based *and* empirically driven:

    SRI FASTUS case study

(It’s all Structured Prediction)



we can break down structured prediction methods into two dimensions.
first is how high up the chomsky hierarchy you go -- the level of complexity and recurisiveness of your structures. the second is whether you design the models by hand, or learn them from data.

so far in this course, everything we’ve done is on the learned side. at the finite-state level there are things like HMMs or chain CRFs with bounded memory (markovian windows). at the CFG level there’s PCFGs and tree CRFs. and there can be more stuff too, like skip-chain CRFs and various increasingly intractable MRFs and stuff.

what we haven’t talked about, at all, are models built by hand. these are not as popular any more. the case study for today, the FASTUS system, is in the lower-left quadrant. but there are interesting comparisons both up and to the right quadrants.

Natural Language Understanding

- For question-answering, dialogue systems, story understanding, etc… one subproblem: want a relational <u>meaning representation</u>
  - (Why relational?)
- Predicate-Argument structures
  - e.g. V(S, O): verb has noun arguments
  - (~Verb) Actions/Events/Frames, *having*
  - (~Noun) Roles/Slots/Arguments

why relational -- you could communicate about the world with single symbols of individual propositions, but that's wasteful, you cross-product out the space too much. Language is compositional and combinatorial, suggesting we use some sort of relational structures to communicate, and this might be a requirement for a meaning representation.

the way to do this is with Predicate-Argument structures.
in syntax, the most basic of all is a subject-verb-object. the verb is a predicate, and it has two noun arguments, a subject and an object. now this isn't the whole story of course, there's many other arguments and such in language -- adjectives can modify nouns, nouns can modify nouns, etc. but SVO is most basic.

when you start talking about semantics, you generalize the Predicate-Argument pairs beyond verbs and nouns. for example, for the predicate, you might have actions, events, or frames, and one of those has a number of roles, slots, or arguments. (following our example, verbs often denote actions and events, though other linguistic things can too.) there are many different types of Predicate-Argument structures, potentially.

Example

Text	I saw a person
SVO syntactic structures	see(I, person) [*verb*=see, *subj*=I, *directobj*=person]
Semantic roles	[*event*=see, *agent*=I, *patient*=person]

(Caveat, IANALinguist!)

here are some simple examples.

semantic roles -- for this simple example all we've done is rename the arguments, but these are supposed to generalize beyond syntax and encode certain types of recurring roles across verbs. some people argue there is a core set of several or maybe a dozen semantic roles. the agent has volition and is causing actions to happen, the patient is a target of the action, an instrument is the means of accomplishing the action, etc.

i always get confused, i'm not a linguist. some people argue that semantic roles don't hold across verbs, that all you do with them is to normalize across different syntactic manifestations. but whatever, in any case there is potential value in representing semantics with predicate-argument structures.

Example

Text	I saw a person
Feature-structure (frame-style?) representation	[*type*= SeeingEvent *time*= Past *subj*= [*word*= I, *grampers*=1st, *num*= sg] ...]

(High-level syntax like LFG / HPSG?)
(Or is it low-level semantics?)

once you go into these pred-arg structures, you can start stuffing in all sorts of features for different grammatical and semantic attributes. ok, this diagram is conflating semantick-y things with high-level syntactic analysis you'd see in a unification grammar like lfg or hpsg. but you might have stuff like, the verb is in the past tense so we know the time the event happened was in the past ... the subject word is 1st-person-singular, etc. lots more predicates and arguments.

Example

Text	I believe I saw a person
Frame-style representation	TopCtx => *event*= believe *agent*= I *theme*= BeliefCtx BeliefCtx => *event*= see *agent*= I *patient*= person
	ctx(TopCtx) ctx(BeliefCtx) inctx(TopCtx, event(believe)) inctx(TopCtx, agent(believe, I)) inctx(TopCtx, theme(believe, BeliefCtx)) inctx(BeliefCtx, event(see)) inctx(BeliefCtx, agent(see, I)) inctx(BeliefCtx, patient(see, person))

(Factivity via Davidsonian semantics,
description/modal logic formalism: Bobrow et al 2005)

or here's more information, contexted events. now the sentence is more complex. the believing event and the seeing of a person event, you could call them "facts" or "propositions", but they aren't quite true in the same way. one approach is to use a "contexted logic", so you say there's a top context or possible world of the speaker's statement, in which the believing event happened, then within the world of the belief, this seeing event happened, and you're allowed to make the imaginary-world-context the object ("theme" I think??) of the believing.

note you can represent this in a flatter pred-arg structure that looks like a list of logical assertions. assert there are two differnet contexts, then facts (the little pred-arg tuples comprising the event tuples) are asserted within a given context.

anyways, this has more structure than the previous examples, but the point is there's all sorts of different semantic phenomena you want various sorts of predicate-argument structures for. now let's turn back to the simplest flat pred-arg structres we had with semantic role events.

Rough sketch: different theoretical traditions?

(leaving out logical semantics, discourse... just flat pred-arg structures)

	Computational Linguistics		Artificial Intelligence
Theory (incomplete)	**Case Grammar** Fillmore 1964, "The Case for Case"		**Frames** Schank and Abelson 1977, "Scripts, Plans, Goals, Understanding"
Datasets	FrameNet VerbNet PropBank (OntoNotes)	Both are predicate-argument recognition problems; structurally similar.	MUC ACE (GENIA)
Struct. Pred. Task	Semantic Role Labeling		Template-Filling IE
Venues	ACL, EMNLP...	More recent work merges annotation levels: i.e. OntoNotes, GENIA	(MUC), AAAI...

this is a horribly reductionist diagram, but there is a genuine bit of separation in these literatures. linguistics and AI are different areas. what we've been talking about with the semantic roles and such basically derives from Fillmore's classic theory of Case Grammar, with lots of other work by others through the years (Jackendoff, Levin, others i'm forgetting). the theories are nice, but to make it concrete you need to make datasets that computers can read. in this vein, ones you may have heard of include framenet, verbnet, propbank, and current work is on ontonotes. Then for any of these, you can analyze text and label it with its lexicon and labels. this is a structured prediction task, and it's called semantic role labeling.

but there's another theoretical tradition too -- frames, or sometimes called scripts. again lots of people working on this but one of the big names is roger schank; schank and abelson 1977 is the main book on it. i'll argue that it eventually evolved into what we now call "template-filling information extraction.", typified by the MUC competition and datasets. also ACE, and also the biomed IE corpus GENIA, though i think that one became more broad over the years.

anyways, the SRL and template-filling IE tasks are, as structured prediction problems, extremely similar. when you read the literature there are funny holes and stuff because people in different research communities tend to publish about different ones. however recent work has merged these strands more and more; both ontonotes and genia have multilevel annotations from syntactic to more semantic labels.

Scripts/Frames
Schank and Abelson (1977)

John went to Lundy's. He ordered lobster. He paid the check and left.

KS	CD
John **$RESTAURANT**	S1 John **PTRANS** John **to** restuarant
	↑E
Customer=John	S1 John **MTRANS** waitress **ATRANS** lobster **to** John **to** waitress
Food=lobster	↑rE
Name=Lundy's	
Location=Brooklyn	
	S1 John **INGEST** lobster **to** stomach
	↑rE
	S1 John **ATRANS** money **to** management
	↑rE
	S1 John **PTRANS** John from restaurant

Thus an entire story spanning many script and non-script-like events would be represented as a linked causal chain of Conceptual Dependency conceptualizations, some subset of which would be linked via the Script link to the scriptname that governs it at the Knowledge Structure level.

the schank and abelson book is kind of crazy and maddeningly vague, but still a bit interesting. i tried to find one picture that might tell something useful about the theory, so here we go.

PTRANS -- physical transfer, like john moved john to the restaurant
MTRANS
ATRANS
rE -- effect resulting from
E

schank is still around. his website is crazy, take a look. he had lots of students, many of them are still around at various universities (like ed hovy's talk a few weeks ago, he was a schank student), two of his former students are here at cmu. if you talk to anyone over 50 or 60 who was in AI back in those days, try asking about roger schank, you will get extremely strong opinions. it is interesting.

Newswire IE: “Sketchy Scripts”

- Gerald DeJong 1982, “FRUMP System”
- The first template-filling IE system?

The $ARREST sketchy script
contains requests for the following events:

1. Police go to where the suspect is.
2. There is optional fighting between the suspect and police.
3. The suspect is apprehended.
4. The suspect is taken to a police station.
5. The suspect is charged.
6. The suspect is incarcerated or released on bond.

gerald dejong is now at uiuc

everyone cites this paper as the first template-filling IE system. the william cohen and andrew mccallum kdd03 tutorial cites it, and jurafsky and martin book does so too. so it must be true.

it’s in this edited volume here filled mostly with schankian stuff. this one is interesting for a reason we’ll get to.

here is an example sketchy script -- it loosely suggests a collection of events that should go together, i guess with a temporal order maybe.

it’s almost like a parody of a hollywood producer or something.

if you go make a probabilistic narrative model, run this on Law and Order episodes. make sure you get really low cross entropy

What it does

Input text

UGANDA TODAY TOOK FORMAL CONTROL OF AN AMERICAN OIL REFINERY.

Script: "country taking economic control of an industry from another"

```
((<=>        (*ATRANS*)
  MANNER     (*FORCED*)
  ACTOR      (*POLITY*)
  OBJECT     (*CONT*)
      TYPE       (*ECONOMIC*)
      PART       (*SPEC-INDUSTRY*)
  TO         (*POLITY*)
  FROM       (*POLITY*) ))
```

System output

```
((<=>      (*ATRANS*)
MANNER     (*FORCED*)
ACTOR      (*UGANDA*)
TO         (*UGANDA*)
OBJECT     (*CONT*)
    TYPE       (*ECONOMIC* CERTAINTY (7))
    PART       (*REFINERY*)
        TYPE       (*OIL*)
        OWNER      (*USA*)
FROM       (*USA*   CERTAINTY (9)) ))
```


he wrote the template, that specifies types of arguments -- a "sketchy script".
the system fills out a template based on input text from a newswire article.

UGANDA TODAY TOOK FORMAL CONTROL OF AN AMERICAN OIL REFINERY.

SUBSTANTIATOR:
PREDICTING (ACTOR) IS SUBJECT OF (TAKE1 2 NIL PAST)

Using its syntactic knowledge, SUBSTANTIATOR determines that the ACTOR will probably be found as the subject of the verb "took."

FOUND POSSIBLE (*POLITY*) FROM WORD# (0) UGANDA
(ACTOR) HAS BEEN FILLED WITH (*UGANDA*)
(TO) HAS BEEN FILLED WITH (*UGANDA*)

Indeed, a *POLITY* was found where the syntactic subject was expected. Therefore it must be the conceptual ACTOR. The TO role is also filled with the same *POLITY* because the verb sense TAKE1 contains the information that its ACTOR and TO role fillers are the same.

PREDICTOR:
PREDICTING ROLE (OBJECT) WILL BE FILLED WITH AN ELEMENT FROM LIST (*POSS* *CONT*)

There are several predicted conceptualizations that the partial under construction can match. Some of them are abstract transfers of POSSession, others of CONTrol. Thus, to differentiate which prediction the text might satisfy, PREDICTOR asks that the OBJECT be filled with either *POSS* or *CONT*.

SUBSTANTIATOR:
PREDICTING (OBJECT) IS VOBJECT OF (TAKE1 2 NIL PAST)

SUBSTANTIATOR has used its syntactic knowledge to decide that if the conceptual OBJECT is specified in the text, it will be the object of the verb "took."

FOUND POSSIBLE (*ABSTRACT*) FROM WORD# 4
(OBJECT) HAS BEEN FILLED WITH (*CONT*)

At word number 4, SUBSTANTIATOR found what it was looking for: a word that means *CONT*.

PREDICTOR:
PREDICTING ROLE (OBJECT PART) WILL BE FILLED WITH AN ELEMENT FROM LIST (*HUMAN* *SPEC-INDUSTRY*)

Again PREDICTOR is trying to differentiate between several viable predictions. The (OBJECT PART) must be filled with either a human or a specific industry.

SUBSTANTIATOR:
WORD# (5) OF CAN POSSIBLY ADD (OBJECT PART)

SUBSTANTIATOR found a preposition that it thinks can provide the desired information.

TENTATIVELY RESOLVING OF TO OF1

PREDICTING (OBJECT PART) IS POBJECT OF (OF1 5)

Here it is looking for the object of the preposition "of" at word 5.

FOUND POSSIBLE (*INDUSTRY*) FROM WORD 9
(OBJECT PART) HAS BEEN FILLED WITH (*REFINERY*)

As the object of the preposition SUBSTANTIATE found "refinery," which it knows is a kind of industry.

PREDICTING (OBJECT PART CLASS) IS MODIFIER OF (REFINERY1 9)
FOUND POSSIBLE (*PRODUCT*) FROM WORD 8
(OBJECT PART CLASS) HAS BEEN FILLED WITH (*OIL*)

To be a specific industry, the kind of refinery must be determined. It decides that the kind of refinery, if present, will probably be an adjective modifier of "refinery" at word 9. It finds "OIL" at word 8.

PREDICTOR:
PREDICTING ROLE (OBJECT TYPE) WILL BE FILLED WITH AN ELEMENT FROM LIST (*ECONOMIC*)

The PREDICTOR has by now narrowed down the number of viable predicted conceptualizations to one. That one requires that the type of control taken over the industry be economic.

SUBSTANTIATOR:
PREDICTING (OBJECT TYPE) IS MODIFIER
LOOKING FOR MODIFIER OF (CONT1 4 10)
TEXT ANALYZER UNABLE TO FIND MODIFIER

SUBSTANTIATOR decides that if the OBJECT TYPE role is present in the text, it will be as an adjective modifier of word 4 "control" as in "took economic control." However, the input phrase does not say "economic control" so the text analyzer cannot add the OBJECT TYPE role.

TRYING INFERENCE RULE
INFERENCER ASKS SLOT (OBJECT) BE FILLED WITH *CONT*
(OBJECT) ALREADY FILLED WITH *CONT*
INFERENCER ASKS SLOT (OBJECT PART) BE FILLED WITH *INDUSTRY*
(OBJECT PART) ALREADY FILLED WITH (*REFINERY*)

SUBSTANTIATE decides to try to infer the desired role filler. It finds an inference rule that can add *ECONOMIC* in the OBJECT TYPE role of *ATRANS* acts provided certain conditions are met. Inference rules are indexed by the conceptual act, and the role they add. Thus, they can be found efficiently. The conditions required by this rule include that control of an industry be changing hands. If that is

i guess you dont have page limits in edited volumes.
he’s annotating the debug output of the system.
the algorithm it’s running is some crazy heuristic search thing.
there’s a semantic module that knows about the script,
and a text analysis module that looks at the text and tries to match it into the slots and they go back and forth.

	true, then the control is probably of type *ECONOMIC*.
ALL TESTS FOR INFERENCE ARE TRUE— INFERRING (OBJECT TYPE) IS (*ECONOMIC*) (OBJECT TYPE) HAS BEEN FILLED WITH (*ECONOMIC* CERTAINTY (7))	
PREDICTOR: PREDICTING ROLE (FROM) WILL BE FILLED WITH AN ELEMENT FROM LIST (*POLITY*)	Finally PREDICTOR requests that the FROM role be filled with a *POLITY*.
SUBSTANTIATOR: TEXT ANALYZER UNABLE TO ADD (FROM) - CALLING INFERENCE PROCEDURES	However, SUBSTANTIATOR cannot add the FROM role using the text.
TRYING INFERENCE RULE INFERENCER ASKS SLOT (OBJECT PART OWNER) BE FILLED WITH *POLITY*	An inference rule is found that says that for abstract transfers the entity giving up the object is probably the same as the current owner of the object. Thus, the problem has been reduced to finding the OBJECT PART OWNER.
FILLER MISSING - SUBSTANTIATOR CALLED PREDICTING (OBJECT PART OWNER) IS MODIFIER OF WORD (9) LOOKING FOR MODIFIER OF (REFINERY1 9) FOUND POSSIBLE (*ANIMATE*) FROM WORD 7 (OBJECT PART OWNER) HAS BEEN FILLED WITH (*USA*)	SUBSTANTIATOR has decided that if the owner is specified in the text, it is probably an adjective modifier of refinery'' at word 9. And indeed the owner is found to be the U.S.
ALL TESTS FOR INFERENCE ARE TRUE— INFERRING (FROM) IS (*USA* CERTAINTY (9))	The inference is made.
PREDICTOR: PREDICTED CONCEPTUALIZATION SATISFIED:	And finally, the predicted conceptualization has been fleshed out.

```
((<=>       (*ATRANS*)
 MANNER     (*FORCED*)
 ACTOR      (*UGANDA*)
 TO         (*UGANDA*)
 OBJECT     (*CONT*)
      TYPE       (*ECONOMIC* CERTAINTY (7))
      PART       (*REFINERY*)
           TYPE       (*OIL*)
           OWNER      (*USA*)
 FROM       (*USA*   CERTAINTY (9)) ))
```

The conceptualization produced contains the information that the industry changing hands is an oil refinery of the United States, that the country taking it is Uganda, and that the country giving it up is the United States. All of this was built in a very purposeful manner. The text was never examined without knowing what conceptual structure was to be built and approximately where in the text it would be found.

It seems as though a lot of work has been done to arrive at the correct parse of the sentence. Indeed, PREDICTOR and SUBSTANTIATOR each had to produce a large number of subresults. However, each of these subresults was achieved very efficiently. Very little work had to be done for any of them. The overall process is made much easier and more efficient because of the exchange of information between PREDICTOR and SUBSTANTIATOR.

A SIX-DAY FRUMP TEST

FRUMP was run in real time on UPI stories for 6 days from April 10 through April 15, 1980. This was a test of seven of FRUMP's sketchy scripts. The seven scripts are:

$MEET	Meetings between organizations
$ACCUSE	One polity condemning the actions of another
$WAR	Incidents of fighting
$AGREE	Agreements between organizations
$MAKE-RELATIONS	Countries establishing diplomatic ties
$BREAK-RELATIONS	Countries breaking diplomatic ties
$AID	A polity giving aid to an organization

several pages later you get an answer. boom.

now, the biggest criticism of the schankians they wrote these crazy things with only a few dozen words of lexical coverage and ran them on like one or two stories or something. very bad generalization.

but there's something cool here -- a real-world test! they took the new news that came out every day and ran it through their system. they're trying to detect several script templates here.

you know everyone wants real-time streaming twitter analysis now? this is the same thing. but with newswire, and lisp.

TABLE 5.1
Six-Day Frump Evaluation

SCRIPT	STORIES CORRECT	STORIES NEARLY CORRECT	STORIES MISSED	STORIES WRONG
$MEET	14	3	13	1
$ACCUSE	5	4	3	1
$WAR	13	8	16	7
$AGREE	11	7	7	6
$MAKE-RELATIONS	0	0	0	0
$BREAK-RELATIONS	2	0	0	1
$AID	0	0	0	0
TOTAL	45	22	39	16

(unusually statistical for a Schankian, and in 1982!)

it’s a confusion matrix! (sum and divide among the columns to get precision and recall.)

ok we can complain, what does “Nearly Correct” mean. but at least they’re doing something here. and it really was a hidden test set.

Application: event analysis in international relations

- Analyze time-series of friendly vs. hostile country-country interactions, coded from newswire text
  - Manual coding (~1960's): hire undergrad annotators to read thousands of articles
  - Machine coding (KEDS) -- based on SVO extraction

    Phil Schrodt (1993, 1994... 2011)
    http://eventdata.psu.edu/

no one in computer science knows about this work but it is cool.

the system phil schrodt built is, as far as i could tell from some of the papers about it, mostly about SVO extraction, often from just the first sentence of a newswire article. but they're been running it and working on variants of it, for years now.

Kansas Event Data System -- now he's at Penn State, so i think the name has changed.

Application: event analysis in international relations

EXAMPLES OF WEIS EVENT CODES

11. REJECT

111 Turn down proposal; reject protest demand; threat
112 Refuse; oppose; refuse to allow

12. ACCUSE

121 Charge, criticize, blame, disapprove
122 Denounce, denigrate, abuse

13. PROTEST

131 Make complaint (not formal)
132 Make formal complaint or protest

17. THREATEN

171 Threat without specific negative sanctions
172 Threat with specific nonmilitary negative sanctions
173 Threat with force specified
174 Ultimatum: threat with negative sanctions and time

18. DEMONSTRATE

181 Non-military demonstration; walk out on
182 Armed force mobilization, exercise and/or display

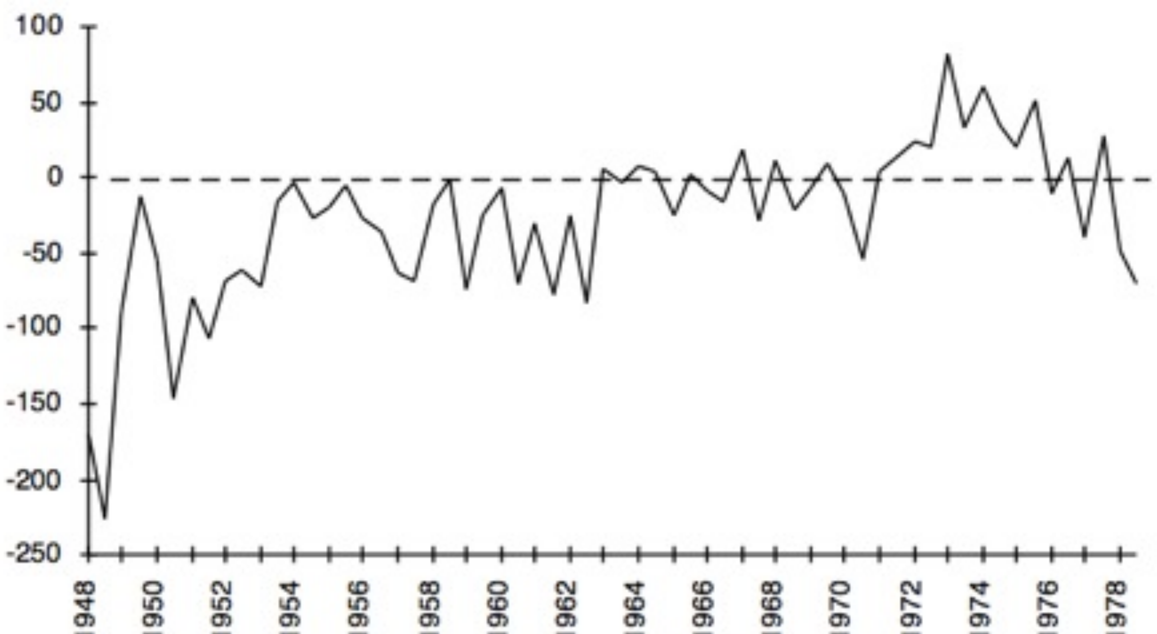
Table 2
WEIS Coding of 1990 Iraq-Kuwait Crisis

Date	Source	Target	WEIS Code	Type of Action
900717	IRQ	KUW	121	CHARGE
900717	IRQ	UAE	121	CHARGE
900723	IRQ	KUW	122	DENOUNCE
900724	IRQ	ARB	150	DEMAND
900724	IRQ	OPC	150	DEMAND
900725	IRQ	EGY	054	ASSURE
900727	IRQ	KUW	160	WARN
900731	IRQ	KUW	182	MOBILIZATION
900801	KUW	IRQ	112	REFUSE
900802	IRQ	KUW	223	MILITARY FORCE

here are coding standards political scientists made, decades before anyone tried to use IE to do it. undergrads annotated lots of articles with this. they worried a lot about interannotator agreement and stuff like that. here's an example of a time series of events.

Application: event analysis in international relations

Figure 1
USA Actions Towards USSR, 1948-1978



Crosscorrelation of Arms Transfers and International Cooperation from Recepient to Supplier

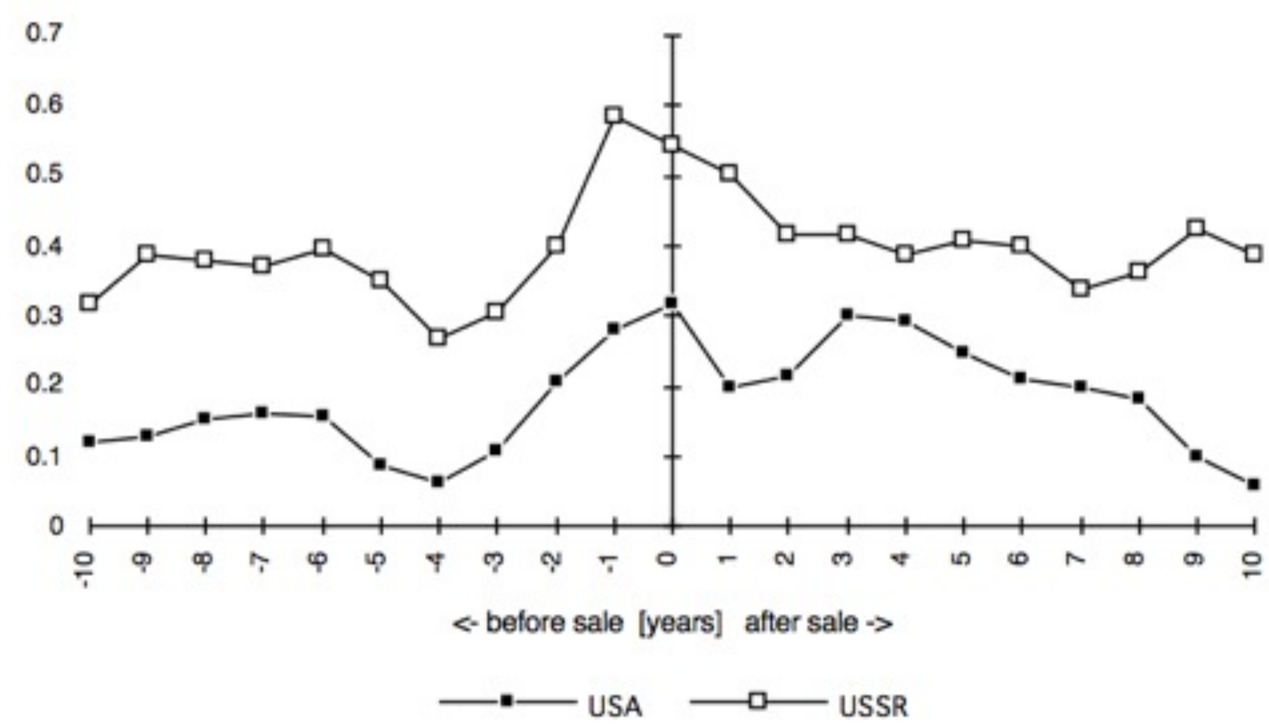
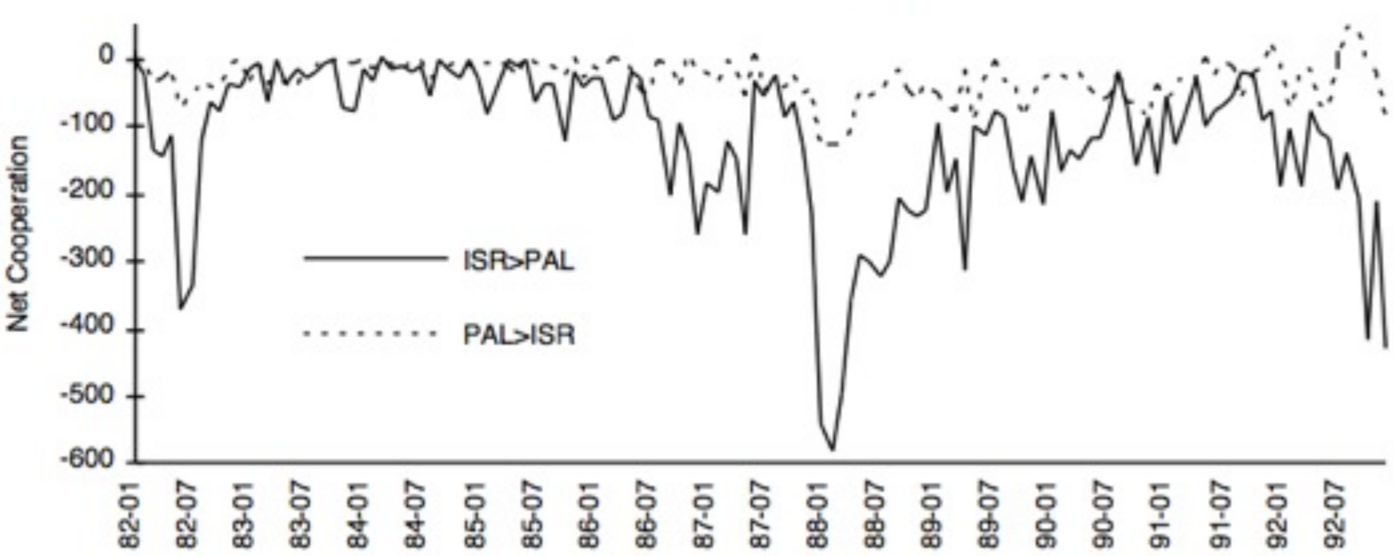


Figure 2
Israel-Palestinian interactions, 1982-1992



(These graphs are from manual coding; IE evaluations in Schrodt and Gerner 1994, King and Lowe 2001)

you can see various international events, like cold war to detente, or the first intifada. also they can use this data to answer substantive questions, like the temporal relationship of arms sales to friendly vs. hostile interactions between countries. (cross-correlation: ?ccf in R)

Message Understanding Conferences (MUC)

- Bakeoff format: shared task, dataset, hidden test set for competitive evaluation
- Different domains – involving specific events
  - (1987) MUC-1: Fleet operations
  - (1991-2) MUC-3, 4: Terrorist activities in Latin America
  - (1993-7) Corporate Joint Ventures, Microelectronic production, Negotiation of Labor Disputes, Airplane crashes, and Rocket/Missile Launches
- ACE (1999-2008) – Automated Content Extraction

this may have been the first bakeoff format shared task in NLP -- at least if you don't count speech and information retrieval, which had these things for a while beforehand.

ACE is kind of a follow-up to MUC. it has more data and annotations

MUC Template-Filling IE

Input: text

San Salvador, 19 Apr 89 (ACAN-EFE) – [TEXT] Salvadoran President-elect Alfredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime.

...

Garcia Alvarado, 56, was killed when a bomb placed by urban guerrillas on his vehicle exploded as it came to a halt at an intersection in downtown San Salvador.

...

Vice President-elect Francisco Merino said that when the attorney general's car stopped at a light on a street in downtown San Salvador, an individual placed a bomb on the roof of the armored vehicle.

...

According to the police and Garcia Alvarado's driver, who escaped unscathed, the attorney general was traveling with two bodyguards. One of them was injured.

Output: extract an event record ("terrorist attack") with the following attributes:

Incident: Date	- 19 Apr 89
Incident: Location	El Salvador: San Salvador (city)
Incident: Type	Bombing
Perpetrator: Individual ID	"urban guerrillas"
Perpetrator: Organization ID	"FMLN"
Perpetrator: Organization Confidence	Suspected or Accused by Authorities: "FMLN"
Physical Target: Description	"vehicle"
Physical Target: Effect	Some Damage: "vehicle"
Human Target: Name	"Roberto Garcia Alvarado"
Human Target: Description	"attorney general": "Roberto Garcia Alvarado" "driver" "bodyguards"

here's the task. note the very domain-specific template. there are several high-level roles or argument types -- incident, perpetrator, targets. the system has to fill in the template with fragments of text from the document.

FASTUS System





- Hobbs, Appelt, Bear, Israel, Kameyana, Stickel, Tyson 1997, “A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text.”
  - From SRI, for early-90‘s MUC
- Hand-built patterns -- but statistically guided development
- Great case study: realistic end-to-end system, with clear architecture, formalisms, and *engagement with the data*
  - Example of how to build a rule-based NLP system -- useful skill in a pinch

Recognizer/Chunker Pipeline

Text

1. Complex Words
2. Basic Phrases
3. Complex Phrases
4. Domain Events
5. Merging Structures

Linguistically general
(~syntax)

Domain specific
(~semantics)

Structure

[Every stage is a Finite State Transducer]

FST's for recognition

(Xerox FST syntax: think of it as a super-regex)

```
# DateParser.script
# Copyright (C) 2004  Lauri Karttunen

define Day     [{Monday} | {Tuesday} | {Wednesday} | {Thursday} |
                {Friday} | {Saturday} | {Sunday}] ;
define Month29 {February};
define Month30 [{April} | {June} | {September} | {December}];
define Month31 [{January} | {March} | {May} | {July} | {August} |
                {October} | {December}] ;
define Month   [Month29 | Month30 | Month31];

# Numbers from 1 to 31
define Date    [OneToNine | [1 | 2] ZeroToNine | 3 [%0 | 1]] ;
# Numbers from 1 to 9999
define Year [OneToNine ZeroToNine^<4];
# Day or [Month and Date] with optional Day and Year
define AllDates [Day | (Day {, }) Month { } Date ({, } Year)];

[...]
define ValidDates [AllDates & MaxDays & LeapDates];
define DateParser [ValidDates @-> "<DATE>" ... "</DATE>"];
```

Add tags for later processing

open-source implementation: http://code.google.com/p/foma/wiki/ExampleScripts

Lauri Karttunen is famous for lots of finite-state morphology stuff. i think this is a demo script he wrote for identifying dates in a text with an FST.

actually nearly all of it is just FSA-like. the key bit for how you use it is the bottom. it spits out these XML-ish tags around the strings matching ValidDates pattern. this is what FST's can do.

(note they do more complicated stuff for morphology)

this is actually an open-source implementation of Xerox's pattern language for FST's. it is fairly new. i believe it compiles to target OpenFST, a lower level algorithmic library for weighted FST's; it does all the unions and minimization and other finite state stuff, so compiles this pattern script into an FST that does date recognition. (OpenFST, in turn is a clone of the old AT&T finite state libraries.)

FSA's for recognition

(Perl-style regex for emoticons)

```
NormalEyes = r'[:=]'
Wink = r'[;]'

NoseArea = r'(|o|O|-)'   ## rather tight precision, \S might be
reasonable...

HappyMouths = r'[D\)\]]'
SadMouths = r'[\(\[]'
Tongue = r'[pP]'
OtherMouths = r'[doO/\\]'  # remove forward slash if http://'s
aren't cleaned

Emoticon = (
    "("+NormalEyes+"|"+Wink+")" +
    NoseArea +
    "("+Tongue+"|"+OtherMouths+"|"+SadMouths+"|"+HappyMouths+")"
)
```

https://github.com/brendano/tweetmotif/blob/master/emoticons.py

heck, you can even use standard perl/unix regexes for recognition. half the battle in maintainability is just decomposing the rules with nice names. no one does this when you have the hacky perl mentality, but you totally can. here's one i wrote for emoticons.

note there are precision/recall tradeoffs with every decision you make when writing rules like this. for example, forward-slash for emoticon mouth gives horrible false positives if there are URLs in the text :/

(skipping ahead, FASTUS stage 4)

Event Patterns

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capital- ized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and “metal wood” clubs a month.

<Company/ies> <Set-up> <Joint-Venture> with <Company/ies>

↓

Relationship:	TIE-UP
Entities:	“Bridgestone Sports Co.” “a local concern” “a Japanese trading house”
Joint Venture Company:	–
Activity:	–
Amount:	–

<Produce> <Product>

↓

Activity:	PRODUCTION
Company:	–
Product:	“golf clubs”
Start Date:	–

ok back to FASTUS. skipping ahead, here’s the core of the algorithm.
you have to write lots of these templated patterns for a particular template you want to be filling. these patterns were made to identify instances of these two different events.
[[BTW -- see “AIML”, AI Markup Language, people use it to make chatbots. it’s basically lots of patterns kind of like this. ELIZA kind of worked like this.]]
Already you can see, if you were running this directly on sequence of words in the text, you have problems. all these multiwords and names, and then relative clauses and stuff separating the words you actually care about. need to do some syntactic analysis first.

(1/5) Complex Words

Text

1. **Complex Words**
2. Basic Phrases
3. Complex Phrases
4. Domain Events
5. Merging Structures

Structure

- Multiword expressions
- Names

back to the pipeline. this first part is simple. you have to have lists of names, and heuristics for identifying types of names like "Co." meaning "company".

BTW, lots of issues here in modern NLP analysis too

(2/5) Basic Phrases

- Small noun chunks
- Verb chunks
- Function word classes
- Some entity classes
- ... this is dictionary lookup + contextual disambiguation. Compare to CRF/HMM?

Company Name:	Bridgestone Sports Co.
Verb Group:	said
Noun Group:	Friday
Noun Group:	it
Verb Group:	had set up
Noun Group:	a joint venture
Preposition:	in
Location:	Taiwan
Preposition:	with
Noun Group:	a local concern
Conjunction:	and
Noun Group:	a Japanese trading house
Verb Group:	to produce
Noun Group:	golf clubs
Verb Group:	to be shipped
Preposition:	to
Location:	Japan

this is now called “chunking” -- the sentence is divided into non-overlapping subsequences of tokens. imagine the rules for each one -- not too hard to get started.

lots of trickiness though. for example, there’s probably a preposition regex including “to”. but “to be” needs to be a verb, and needs to want to grab the next verb to the right “shipped”. you can imagine lots of priority orderings and overrides. i good pattern rule language should let you do these things.

note that, fundamentally, these are the same sources of information as in a HMM or CRF chunker/tagger. emissions weights are soft versions of lexicons (FST-unions). transition weights are local contextual information. etc.

(these would be called “noun chunks” now)

Noun groups are recognized by a finite-state grammar that encompasses most of the complexity that can occur in English noun groups, including numbers, numerical modifiers like “approximately”, other quantifiers and determiners, participles in adjectival position, comparative and superlative adjectives, conjoined adjectives, and arbitrary orderings and conjunctions of prenominal nouns and noun-like adjectives. Thus, among the noun groups recognized are

approximately 5 kg
more than 30 people
the newly elected president
the largest leftist political force
a government and commercial project

Finite-state syntactic parsing!

(3/5) Complex Phrases

- Complex noun groups (noun phrases): PP attachments, appositives, noun conjunction
- Complex verb groups: Conjunctions, auxiliaries, modalities

Collapse across some verb auxiliaries ...

GM *formed* a joint venture with Toyota.
GM *announced it was forming* a joint venture with Toyota.
GM *signed an agreement forming* a joint venture with Toyota.
GM *announced it was signing an agreement to form* a joint venture with Toyota.

“announced it was forming” as a synonym to “form” -- at a deep natural language understanding level, these are different. but perhaps in this domain, if you’re a business analyst or something, they’re as good as synonyms.

(3/5) Complex Phrases

- Complex noun groups (noun phrases): PP attachments, appositives, noun conjunction
- Complex verb groups: Conjunctions, auxiliaries, modalities

Collapse across some verb auxiliaries ...

but not others

GM *formed* a joint venture with Toyota.
GM *announced it was forming* a joint venture with Toyota.
GM *signed an agreement forming* a joint venture with Toyota.
GM *announced it was signing an agreement to form* a joint venture with Toyota.

The status of the joint venture is "Planned" rather than "Existing":

GM *will form* a joint venture with Toyota.
GM *plans to form* a joint venture with Toyota.
GM *expects to form* a joint venture with Toyota.
GM *announced plans to form* a joint venture with Toyota.

but sometimes the modalities really do matter. "planning to" is weak and soft in this domain compared to "announced".

Other finite-state technology in NLP

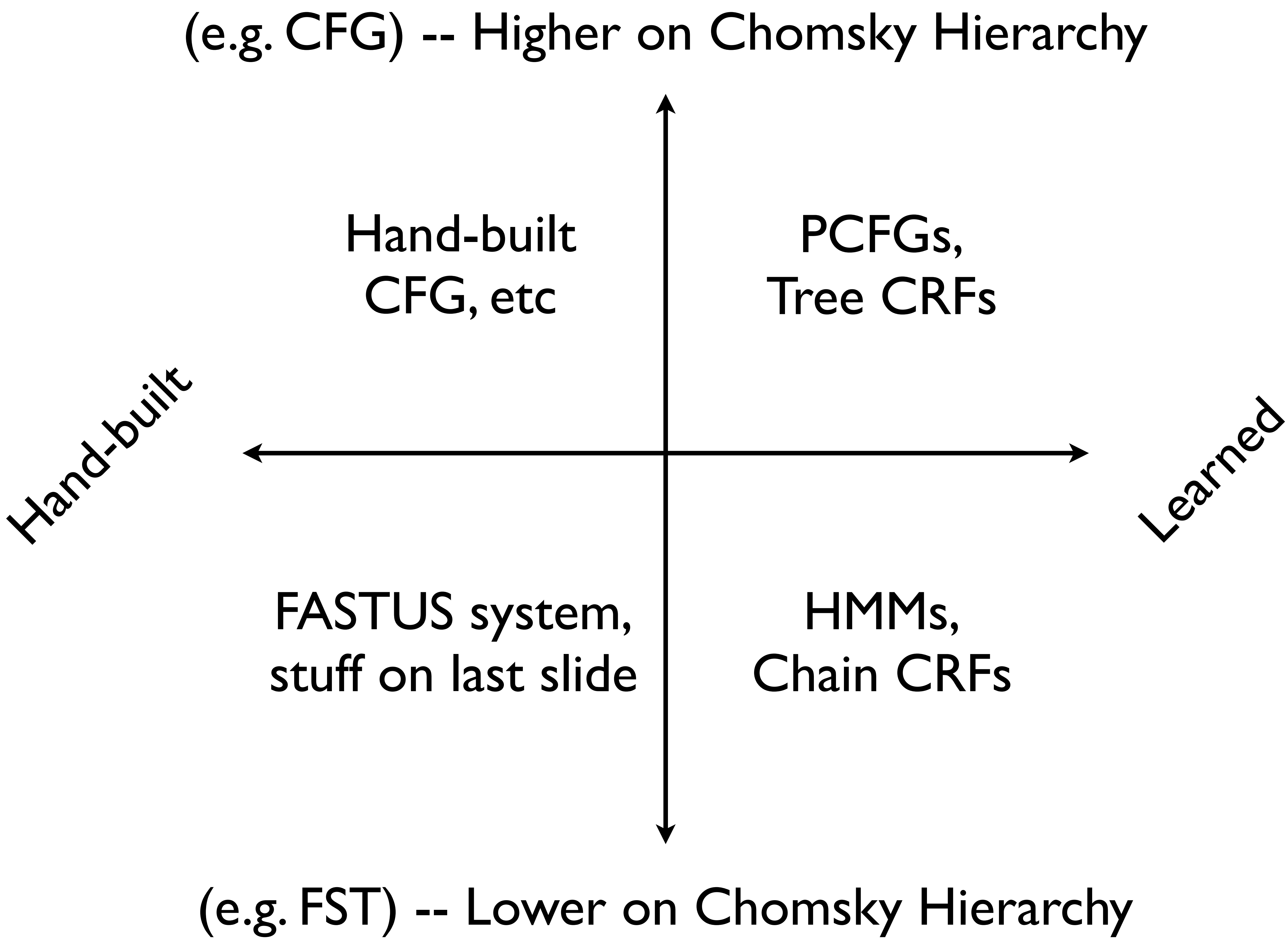
- Pereira 1990 -- finite-state approximations of grammars
- Abney 1996 -- finite-state partial parsing via cascades (still can download his CASS system)
- Morphology -- e.g. Inxight analyzer
- Book: *Finite State Devices for Natural Language Processing*, ed. Roche and Schabes, 1997 (containing the Hobbs article)

ok, that's the core of their syntax system.

this is a lot of fairly sophisticated syntactic analysis. if someone told you you need a recursive CFG-style parser to do this, maybe you don't always. there's been lots of work along these lines.

also, finite-state methods are especially popular in morphology, where they're a pretty plausible explanation of lots of the phenomena.

(It's all Structured Prediction)



hmms and chain crfs are pretty popular these days. maybe the finite-state level of the chomsky hierarchy is good enough, especially if you hack it up for a little bit of depth-bounded structure...

(4/5): Domain Events

(5/5): Merge Structures

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capital- ized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and "metal wood" clubs a month.

<Company/ies> <Set-up> <Joint-Venture> with <Company/ies>

↓

Relationship:	TIE-UP
Entities:	"Bridgestone Sports Co." "a local concern" "a Japanese trading house"
Joint Venture Company:	–
Activity:	–
Amount:	–

<Produce> <Product>

↓

Activity:	PRODUCTION
Company:	–
Product:	"golf clubs"
Start Date:	–

“Pseudo-Syntax”

A certain amount of “pseudo-syntax” is done in Stage 4. The material between the end of the subject noun group and the beginning of the main verb group must be read over. There are patterns to accomplish this. Two of them are as follows:

Subject {Preposition NounGroup}* VerbGroup

Subject Relpro {NounGroup | Other}* VerbGroup {NounGroup | Other}* VerbGroup

The first of these patterns reads over prepositional phrases. The second over relative clauses. The verb group at the end of these patterns takes the subject noun group as its subject. There is another set of patterns for capturing the content encoded in relative clauses, of the form

Subject Relpro {NounGroup | Other}* VerbGroup

Generalizing an SVO template

S **V** **O**
GM **manufactures** **cars**.

illustrates a general pattern for recognizing a company's activities. But the same semantic content can appear in a variety of ways, including

Cars are **manufactured** by **GM** ...
GM, which **manufactures** **cars** ...
... **cars**, which are **manufactured** by **GM** ...
... **cars** **manufactured** by **GM** ...
GM is to **manufacture** **cars**.
Cars are to be **manufactured** by **GM**.
GM is a **car** **manufacturer**.

These are all systematically related to the active form of the sentence. Therefore, there is no reason a user should have to specify all the variations. The FASTUS system is able to generate all of the variants of the pattern from the simple active (**S**-**V**-**O**) form.
These transformations are **executed at compile time**, producing the more detailed set of patterns, so that at run time there is no loss of efficiency.

by cross-product exploding the FST (is ok!)

this is starting to look more like semantic roles --
they're generalizing over different types of syntactic relations
to get the semantic arguments.

there's a space/time tradeoff here -- they're going for high space, since you cross-product all these syntactic variations against every S-V-O active voice triple given by the user. then you have a fast FST for runtime.

(4/5) Domain Events

(5/5) Merge Structures

Activity:	PRODUCTION
Company:	–
Product:	“golf clubs”
Start Date:	–

Activity:	PRODUCTION
Company:	“Bridgestone Sports Taiwan Co.”
Product:	–
Start Date:	DURING: January 1990

Relationship:	TIE-UP
Entities:	“Bridgestone Sports Co.” “a local concern” “a Japanese trading house”
Joint Venture Company:	–
Activity:	–
Amount:	–

Relationship:	TIE-UP
Entities:	–
Joint Venture Company:	“Bridgestone Sports Taiwan Co.”
Activity:	–
Amount:	NT$20000000

Bridgestone Sports Co. said Friday it has set up a **joint venture** in Taiwan with **a local concern** and **a Japanese trading house** to **produce golf clubs** to be shipped to Japan.

The joint venture, **Bridgestone Sports Taiwan Co.**, **capitalized** at **20 million new Taiwan dollars**, will start **production** in **January 1990** with production of 20,000 iron and “metal wood” clubs a month.

Run all the templated patterns, they extract all these events. but they’re fragmentary and talk about the same things. we need to merge them.

(4/5) Domain Events

(5/5) Merge Structures

Activity:	PRODUCTION
Company:	–
Product:	“golf clubs”
Start Date:	–

Activity:	PRODUCTION
Company:	“Bridgestone Sports Taiwan Co.”
Product:	–
Start Date:	DURING: January 1990

Relationship:	TIE-UP
Entities:	“Bridgestone Sports Co.”
	“a local concern”
	“a Japanese trading house”
Joint Venture Company:	–
Activity:	–
Amount:	–

Relationship:	TIE-UP
Entities:	–
Joint Venture Company:	“Bridgestone Sports Taiwan Co.”
Activity:	–
Amount:	NT$20000000

Decide identity coreference through name-matching and type compatibility; if arguments are coreferent, merge events

Activity:	PRODUCTION
Company:	“Bridgestone Sports Taiwan Co.”
Product:	“iron and ‘metal wood’ clubs”
Start Date:	DURING: January 1990

Relationship:	TIE-UP
Entities:	“Bridgestone Sports Co.”
	“a local concern”
	“a Japanese trading house”
Joint Venture Company:	“Bridgestone Sports Taiwan Co.”
Activity:	–
Amount:	NT$20000000

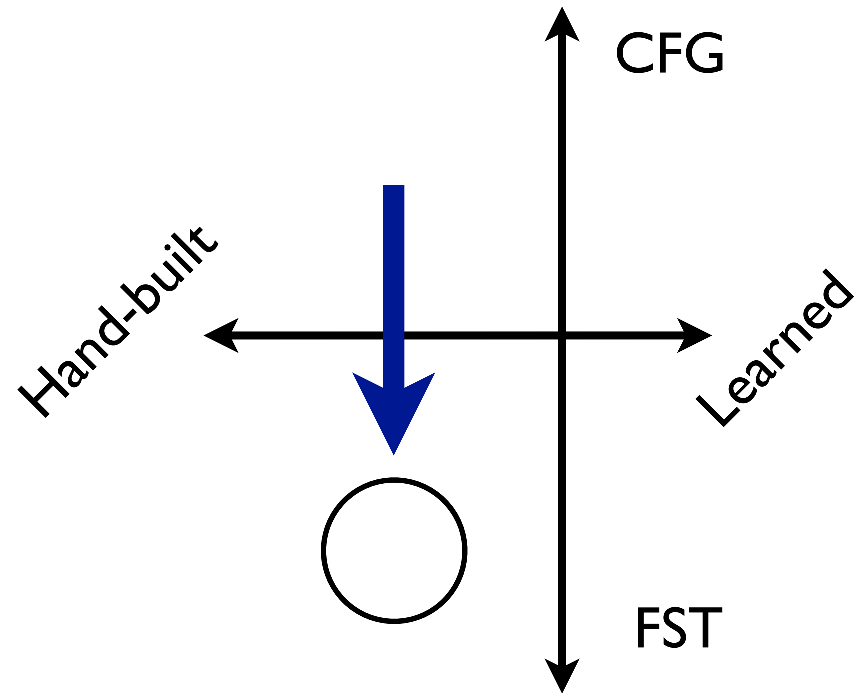
have to do coreference. sometimes making assumptions that these events are the same. this is kind of ok in these short newswire articles, because all the text is describing the same thing, or various aspects of it. simple discourse structures let you get away with sweeping assumptions.

The template as pragmatics

One of the lessons to be learned from our FASTUS experience is that many information extraction tasks are much easier than anyone ever thought. Although the full linguistic complexity of the texts is often very high, with long sentences and interesting discourse structure problems, the relative simplicity of the information-extraction task allows much of this linguistic complexity to be bypassed—indeed much more than we had originally believed was possible. The key to the whole problem, as we see it from our FASTUS experience, is to **do exactly the right amount of syntax, so that pragmatics can take over its share of the load**.

... like you're talking to a robot that only cares about terrorist activities in latin america, and tries really really hard to interpret everything like this.

Empirical Rule-based NLP



- Originally FASTUS was just a preprocessor for a more complex system. It was too slow, they threw it out -- deadline pressure
- Hours vs Minutes runtime on development set -- much faster development iterations

January: Designed FASTUS
Jan-May: Development
May 6: First test of the FASTUS system on a blind test set of 100 terrorist reports, which had been withheld as a fair test, and we obtained a score of **8% recall and 42% precision**.

At that point we began a fairly intensive effort to hill-climb on all 1300 development texts then available, doing periodic runs on the fair test to monitor our progress. This effort culminated in a score of **44% recall and 57% precision** in the wee hours of **June 1**, when we decided to run the official test. The rate of progress was rapid enough that even a few hours of work could be shown to have a noticeable impact on the score. Our scarcest resource was time, and our supply of it was eventually exhausted well before the point of diminishing returns.

We were thus able, **in three and a half weeks, to increase the system's F-score by 36.2 points, from 13.5 to 49.7**.

wish i could find it, there's this amazing graph of their F-Score over time. they throw out the parser, it starts increasing.

note they have sizable team working on this. need the strictly modular pipeline to stay sane.

The quick-and-dirty 75% solution

The FASTUS system was an order of magnitude faster than the other leading systems at MUC-4.

Out of the seventeen sites participating in MUC-4, only General Electric's system performed significantly better (a recall of 62% and a precision of 53% on the first test set), and their system had been under development for over five years (Sundheim, 1992).

Human intercoder reliability on information extraction tasks is in the 65-80% range. Thus, we believe this technology can perform at least 75% as well as humans.

(Claims are a little strong, but point stands)

Advantages of rule-based NLP

- Practically speaking, often not enough labeled data and unsupervised learning is a science project -- a little linguistic knowledge can go a long way
- Rule-based systems are state-of-the-art for some NLP tasks
  - Tokenization -- problem so simple
    (and many other small tasks... e.g. orthographic normalization)
  - Coreference -- problem so complex (CoNLL 2011, Stanford "DCoref")
  - Morphology (?)
- Finite state languages
  - Feature engineering
  - Time, date recognition...
  - William story about Minorthird
- Key lesson from FASTUS: use empirical methodology to keep on track
- Editorial: compared to machine learning, rule-based development forces you to *look at the data* -- the most important part in any approach

Since the mid-90's...

Text

1. Complex Words
2. Basic Phrases
3. Complex Phrases
4. Domain Events
5. Merging Structures

Linguistically general *(~syntax)*

Domain specific *(~semantics)*

Structure

the FASTUS pipeline roughly corresponds to the full text analysis pipeline needed by any NLP document-understanding application.

Since the mid-90's...

Text

1. Complex Words
2. Basic Phrases
3. Complex Phrases
4. Domain Events
5. Merging Structures

Structure

(remember, ignoring logical semantics, discourse ...)

Syntax: Lots of work.
POS, NER tagging,
phrase chunking, structure parsing, dependency parsing...

Pattern Learning: Lots of work.
Riloff bootstrapping...
Open IE (NELL, TextRunner)

Event Semantics: Far less work.
e.g. Chambers/Jurafsky 2011, learning the templates [with a crazy ad-hoc clustering cascade]
Haghighi/Klein 2010, template IE [with a crazy giant graphical model]

the most work has gone into syntactic analysis.

closer to IE, lots of work has sought to address the narrowness and brittleness of these per-domain handcrafted patterns.

Conclusions: frames and finite-state IE

- Concrete empirical tasks we see today may have interesting theoretical roots
- Interesting theories need concrete empirical definitions
- Finite-state patterns and hand-built rules: more powerful than you might think. Try the 80% solution first.
- Many open areas of research

When we first implemented the Complex Phrase level of processing, our intention was to use it only for complex noun groups, as in the attachment of "of" prepositional phrases to head nouns. Then in the final week before an evaluation, we wanted to make a change in what sorts of verbs were accepted by a set of patterns; this change, though, would have required our making extensive changes in the domain patterns. Rather than do this at such a late date, we realized it would be easier to define a complex verb group at the Complex Phrase level. We then immediately recognized that this was not an *ad hoc* device, but in fact the way we should have been doing things all along. We had stumbled onto an important property of

Input	Output

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and "metal wood" clubs a month.

TIE-UP-1:

Relationship:	TIE-UP
Entities:	"Bridgestone Sports Co."
	"a local concern"
	"a Japanese trading house"
Joint Venture Company:	"Bridgestone Sports Taiwan Co."
Activity:	ACTIVITY-1
Amount:	NT$20000000

ACTIVITY-1:

Activity:	PRODUCTION
Company:	"Bridgestone Sports Taiwan Co."
Product:	"iron and 'metal wood' clubs"
Start Date:	DURING: January 1990