Natural Language Processing
(for smart people who know nothing about natural language processing)

Hal Daume III | UMD | me@hal3.name | @haldaumeme3
Machine Learning Summer School 2012 @ UC Santa Cruz
What is NLP?

- Fundamental goal: deep understanding of text
  - Not just string processing or keyword matching
- End systems that we want to build
  - Simple: Spelling correction, text categorization, etc.
  - Complex: Speech recognition, machine translation, information extraction, dialog interfaces, question answering
  - Unknown: human-level comprehension (might be more than just NLP)
Topology of the Field

ICASSP

Automatic Speech Recognition

Information Retrieval

Human Language Technologies

Computational Linguistics

Natural Language Processing

SIGIR

ACL

???
What can I teach you in 6 hours...

- a bag is just language of not words
  (no math, no machine learning)

- NLP through the eyes of machine translation
  (a little math, a little machine learning)

- Machine learning for complex structured prediction problems (and then some)
  (lotsa math, lotsa machine learning)

- How to learn language by interacting with the world
  (some math, some machine learning)
Speech

- Automatic speech recognition (ASR)
  - Audio in, text out
  - SOTA: 0.3% error for digits, 10% dictation, 50%+ TV

"Speech Lab"

- Text to speech (TTS)
  - Text in, audio out
  - SOTA: completely intelligible (sometimes unnatural)
Question answering

- More than search
- Can be really easy: “What is the capital of Wyoming?”
- Can also be hard →
- Can be open ended: “What are the main issues in the global warming debate?”

- SOTA: Can do factoids, can't do much else
Information Extraction

Unstructured text to database entries

New York Times Co. named Russell T. Lewis, 45, president and general manager of its flagship New York Times newspaper, responsible for all business-side activities. He was executive vice president and deputy general manager. He succeeds Lance R. Primis, who in September was named president and chief operating officer of the parent.

<table>
<thead>
<tr>
<th>Person</th>
<th>Company</th>
<th>Post</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russell T. Lewis</td>
<td>New York Times newspaper</td>
<td>president and general manager</td>
<td>start</td>
</tr>
<tr>
<td>Russell T. Lewis</td>
<td>New York Times newspaper</td>
<td>executive vice president</td>
<td>end</td>
</tr>
<tr>
<td>Lance R. Primis</td>
<td>New York Times Co.</td>
<td>president and CEO</td>
<td>start</td>
</tr>
</tbody>
</table>

SOTA: 40% for hard things, 80% accuracy for multi-sentence templates, 90%+ for single easy fields
Summarization

- Condensing documents
  - Single or multiple
  - Extractive or abstractive
  - Indicative or informative
  - Etc...
- Very context (and user) dependent
- Involves analysis and generation
- SOTA: who knows...
Machine translation
Language comprehension

But man is not destined to vanish. He can be killed, but he cannot be destroyed, because his soul is deathless and his spirit is irrepressible. Therefore, though the situation seems dark in the context of the confrontation between the superpowers, the silver lining is provided by amazing phenomenon that the very nations which have spent incalculable resources are desperately trying to threaten each other, before the total holocaust.

The main point from the author's view is:

1. Man's soul and spirit can not be destroyed by superpowers.
2. Man's destiny is not fully clear or visible.
3. Man's soul and spirit are immortal.
4. Man's safety is assured by the delicate balance of power in terms of nuclear weapons.
5. Human society will survive despite the serious threat of total annihilation.
Why is language hard?

- Ambiguity abounds (some headlines)
  - Iraqi Head Seeks Arms
  - Teacher Strikes Idle Kids
  - Kids Make Nutritious Snacks
  - Stolen Painting Found by Tree
  - Local HS Dropouts Cut in Half
  - Enraged Cow Injures Farmer with Ax
  - Hospitals are Sued by 7 Foot Doctors
  - Ban on Nude Dancing on Governor's Desk
- Why are these funny?
Why is language hard?

- Ambiguity abounds (some headlines)
  - Iraqi Head Seeks Arms
  - Teacher Strikes Idle Kids
  - Kids Make Nutritious Snacks
  - Stolen Painting Found by Tree
  - Local HS Dropouts Cut in Half
  - Enraged Cow Injures Farmer with Ax
  - Hospitals are Sued by 7 Foot Doctors
  - Ban on Nude Dancing on Governor's Desk
- Why are these funny?
Why (else) is language hard?

• Data sparsity
  • NYT: 500M+ words of newswire text
  • Brown Corpus: 1M words of tagged “balanced” text
  • Penn Treebank: 1M words of parsed WSJ text
  • Canadian Hansard: 10M+ words of parallel French/English text
  • The Web: *B+ words of who knows what...

Brain teaser:
You've already seen N tokens of text.
This contained a bunch of word types.
What is the probability that the N+1st word will be new?
Why (else) is language hard?

![Graph showing the frequency of unigrams and bigrams across different numbers of words. The graph illustrates the probability distribution of word occurrences, with unigrams having a higher frequency than bigrams.]
Structural ambiguity

Time flies like an arrow.

I saw the Grand Canyon flying to New York.

I saw the man on the hill with the telescope.

I ate spaghetti with....

.... meatballs

.... a fork
Hurricane Emily howled toward Mexico's Caribbean coast on Sunday packing 135 mph winds and torrential rain and causing panic in Cancun, where frightened tourists squeezed into musty shelters.

SOTA: 90% accuracy for mid-1980s WSJ text
Syntax doesn't nail down semantics

• Every morning someone's alarm clock wakes me up

• Every person on this island speaks two languages

• Please come see me next Friday
Discourse: coreference

President John F. Kennedy was assassinated.
The President was shot yesterday.
Relatives said that John was a good father.
JFK was the youngest president in history.
His family will bury him tomorrow.
Friends of the Massachusetts native will hold
a candlelight service in Mr. Kennedy's
home town.

SOTA: 70-80% accuracy for newswire text
Discourse structure

• I only like traveling to Europe.
  So I submitted a paper to ACL.

• I only like traveling to Europe.
  Nevertheless I submitted a paper to ACL.
Pragmatics

Rules of conversation:
- Can you tell me what time it is?
- Could you pass the salt?

Speech acts:
- I bet you $50 that the Jazz will win.
- Will you marry me?
Pragmatics

Scene 1: Penn Station, NY
(Hal) Long Beach?
(Passerby) Downstairs, LIRR Station.

S2: Ticket Counter, LIRR
(Hal) Long Beach?
(Clerk) $4.50.

Scene 3: Info. Booth, LIRR Station
(Hal) Long Beach?
(Clerk) 4:19, Track 17.

Scene 4: On train, vicinity of Forest Hills
(Hal) Long Beach?
(Conductor) Change at Jamaica.

Scene 5: Next train, near Lynbrook
(Hal) Long Beach?
(Conductor) Right after Island Park.
Translate Centauri → Arcturan

Your assignment, translate this Centauri sentence to Arcturan:

*farok crrrok hihok yorok clok kantok ok-yurp*

<table>
<thead>
<tr>
<th>Centauri Sentences</th>
<th>Arcturan Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. ok-voon ororok sprok .</td>
<td>7a. lalok farok ororok lalok sprok izok enemok</td>
</tr>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hillat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok clok .</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hillat .</td>
</tr>
<tr>
<td>5a. wiwok farok izok stok .</td>
<td>11a. lalok nok crrrok hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjatquat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat krat quat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
A Bit of History

1940s   Computations begins, AI hot, Turing test
         Machine translation = Code-breaking?
1950s   Cold war continues
1960s   Chomsky and statistics, ALPAC report
1970s   Dry spell
1980s   Statistics makes significant advances in speech
1990s   Web arrives
         Statistical revolution in machine translation, parsing, IE, etc
         Serious “corpus” work, increasing focus on evaluation
2000s   Focus on optimizing loss functions, reranking
         Even more focus on evaluation, automatic metrics
         Huge process in machine translation
         Gigantic corpora become available, scaling
         New challenges
Collection: RoadVehicle

GAF Arg: 1

Mt: BaseKB
isa: "PublicConstant-DefinitionalGAFsOK" "PublicConstant-CommentOK" "PublicConstant"

Mt: TransportationVocabularyMt
isa: "ExistingObjectType" "ProductType"
genl: "WheeledVehicle" "TransportationDevice-Vehicle" "LandTransportationDevice"
   "TransportationContainerProduct"

Mt: ProductGVocabularyMt
disjointWith: "TrainEngine"

Mt: TransportationVocabularyMt
comment: "A specialization of both LandTransportationDevice and TransportationDevice-Vehicle. Each instance of RoadVehicle is a vehicle designed primarily for travel on roads (although some instances may also have limited off-road capabilities). Notable specializations of RoadVehicle include Automobile, Truck, and Bus-RoadVehicle. Since RoadVehicle is a specialization of TransportationDevice-Vehicle, each instance of RoadVehicle is self-powered. Consequently, road transportation devices which are not self-powered (for example, all the instances of Bicycle) are not included in this collection."

Mt: EnglishParaphraseMt
What is Nearby NLP?

• Computational Linguistics
  • Using computational methods to learn more about how language works
  • We end up doing this and using it

• Cognitive Science
  • Figuring out how the human brain works
  • Includes the bits that do language
  • Humans: the only working NLP prototype!

• Speech?
  • Mapping audio signals to text
  • Traditionally separate from NLP, converging?
  • Two components: acoustic models and language models
  • Language models in the domain of stat NLP
English is an outlier...

<table>
<thead>
<tr>
<th>Turkish</th>
<th>English</th>
<th>Turkish</th>
<th>English</th>
<th>Turkish</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>uyuyorum</td>
<td>I am sleeping</td>
<td>uyuorsun</td>
<td>you are sleeping</td>
<td>uyumadanan</td>
<td>you are sleeping</td>
</tr>
<tr>
<td>uyuyor</td>
<td>he/she/it is sleeping</td>
<td>uyuoloruz</td>
<td>we are sleeping</td>
<td>uyumadanan</td>
<td>we must sleep</td>
</tr>
<tr>
<td>uyuyorsunuz</td>
<td>you are sleeping</td>
<td>uyumaliyiz</td>
<td>they are sleeping</td>
<td>uyumadanan</td>
<td>without sleeping</td>
</tr>
<tr>
<td>uyuduk</td>
<td>we slept</td>
<td>uyumadanan</td>
<td>we must sleep</td>
<td>uyumadanan</td>
<td></td>
</tr>
<tr>
<td>uyuman</td>
<td>your sleeping</td>
<td>uyumadanan</td>
<td></td>
<td>uyumadanan</td>
<td></td>
</tr>
</tbody>
</table>
Classical MT (1970s and 1980s)
How Much Analysis?

Interlingua

Source Semantics

Target Semantics

Source Syntax

Target Syntax

Source Morphology

Target Morphology

Source Words

Target Words

Analysis

Generation

Direct

Classical

Becoming Popular

Not practical for open domain
Syntactic Transfer

- Now, just get a bunch of linguists to sit down and write rules and grammars

```
S
  / \  
A   V
 / \  /
NP  NP
  / \
D   N
```

```
S
  / \  
A   V
 /   /
NP  NP
  / \
D   N
```

The student will see the man

Der Student wird den Mann sehen
While We're Busy Writing Grammars

Information Source $\Rightarrow$ Noisy Channel $\Rightarrow$ Corrupted Output

$p(s)$ $\Rightarrow$ $p(o | s)$ $\Rightarrow$ $p(s | o) \propto p(s) * p(o | s)$

“Imagined” Words $\Rightarrow$ Speech Process $\Rightarrow$ Acoustic Signal

Need: $p(\text{word sequence})$ and $p(\text{signal} | \text{word sequence})$
Acoustic Modeling: $p(a \mid w)$

**Signal:**

**Transcription:** the man ate

**Key notion:** acoustic-word

---

Chicken-and-egg problem that we can solve using Expectation Maximization
Language Modeling: $p(w)$

- **Sentence = sequence of symbols from alphabet**

\[
p(w_1, w_2, \ldots, w_I) = \prod_{i=1}^{I} p(w_i \mid w_1, \ldots, w_{i-1})
\]

In practice, probabilities are estimated from a large corpus, but are “smoothed” intelligently to avoid zero probability $n$-grams.

Language modeling is often the art of good smoothing.

See [Goodman 1998]

The beloved $n$-gram language model
Speech Rec = Machine Translation?

- Peter F. Brown
- Stephen A. Della Pietra
- Vincent J. Della Pietra
- Robert Mercer
- *The Mathematics of Statistical Machine Translation: Parameter Estimation*
- Computational Linguistics 19 (2), June 1993
  
  “Brown 93”

- Maybe most important paper in NLP in last 20 years
### Centauri/Arcturan [Knight 97]

**Your assignment, translate this to Arcturan:**

<table>
<thead>
<tr>
<th>Centauri</th>
<th>Arcturan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. ok-voon ororok sprok .</td>
<td>farok crrrok hihok yorok clok kantok ok-yurp .</td>
</tr>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7a. lalok farok ororok lalok sprok izok enemok .</td>
</tr>
<tr>
<td></td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok clok .</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok farok izok stok .</td>
<td>11a. lalok nok crrrok hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjatquat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat kratquat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
Centauri/Arcturan [Knight 97]

Your assignment, translate this to Arcturan:

<table>
<thead>
<tr>
<th>1a. ok-voon ororok sprok .</th>
<th>7a. lalok <strong>farok</strong> ororok lalok sprok izok enemok .</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok clok .</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok <strong>farok</strong> izok stok .</td>
<td>11a. lalok nok crrrok hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjat quat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat krat quat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
## Centauri/Arcturan [Knight 97]

**Your assignment, translate this to Arcturan:**

<table>
<thead>
<tr>
<th>Centauri</th>
<th>Arcturan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. ok-voon ororok sprok .</td>
<td>7a. lalok <strong>farok</strong> ororok lalok sprok izok enemok .</td>
</tr>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7b. wat <strong>jjet</strong> bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok clok .</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok <strong>farok</strong> izok stok .</td>
<td>11a. lalok nok crroko hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat <strong>jjet</strong>quat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat krat quat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
### Centauri/Arcturan [Knight 97]

**Your assignment, translate this to Arcturan:**

<table>
<thead>
<tr>
<th>1a. ok-foon ororok sprok .</th>
<th>7a. lalok <strong>farok</strong> ororok lalok sprok izok enemok .</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b. at-foon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-foon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-foon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-foon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok clok .</td>
</tr>
<tr>
<td>4b. at-foon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok <strong>farok</strong> izok stok .</td>
<td>11a. lalok nok <strong>errrokk</strong> hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjat quat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat kratquat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
Your assignment, translate this to Arcturan:  

<table>
<thead>
<tr>
<th>1a. ok-voon ororok sprok</th>
<th>7a. lalok farok ororok lalok sprok izok enemok</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b. at-voon bichat dat</td>
<td>7b. wat jjat bichat wat dat vat eneat</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok ploks prok</td>
<td>8a. lalok brok anok ploks nok</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat</td>
<td>8b. iat lat pippat rrat nnat</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3a. erok sprok izok <strong>hihok</strong> ghirok</td>
<td>9a. wiwok nok izok kantok ok-yurp</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat</td>
<td>9b. totat nnat quat oloat at-yurp</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok</td>
<td>10a. lalok mok nok yorok ghirok clok</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat</td>
<td>10b. wat nnat gat mat bat hilat</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5a. wiwok farok izok stok</td>
<td>11a. lalok nok crrrok <strong>hihok</strong> yorok zanzanok</td>
</tr>
<tr>
<td>5b. totat jjat quat cat</td>
<td>11b. wat nnat arrat mat zanzanat</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok</td>
<td>12a. lalok ralok nok izok <strong>hihok</strong> mok</td>
</tr>
<tr>
<td>6b. wat dat krat quat cat</td>
<td>12b. wat nnat farat arrat vat gat</td>
</tr>
</tbody>
</table>
### Centauri/Arcturan [Knight 97]

Your assignment, translate this to Arcturan:  

<table>
<thead>
<tr>
<th>English</th>
<th>Arcturan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. ok-oon ororok sprok .</td>
<td>7a. lalok farok ororok lalok sprok izok enemok .</td>
</tr>
<tr>
<td>1b. at-oon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-oon anok ploks ploks sprok .</td>
<td>8a. lalok brok anok ploks nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-oon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok <strong>hihok</strong> ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-oon anok drok brok jok .</td>
<td>10a. lalok mok nok <strong>yorok</strong> ghirok clok .</td>
</tr>
<tr>
<td>4b. at-oon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok farok izok stok .</td>
<td>11a. lalok nok crrrok <strong>hihok yorok</strong> zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjat quat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok <strong>hihok</strong> mok .</td>
</tr>
<tr>
<td>6b. wat dat krat quat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
# Centauri/Arcturan [Knight 97]

**Your assignment, translate this to Arcturan:**

<table>
<thead>
<tr>
<th>English</th>
<th>Arcturan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. ok-voon ororok sprok .</td>
<td>7a. lalok farok ororok lalok sprok izok enemok .</td>
</tr>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnatquat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok [clok] .</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok farok izok stok .</td>
<td>11a. lalok nok crrrok hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjat quat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat krat quat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
Your assignment, translate this to Arcturan:  

<table>
<thead>
<tr>
<th>Centauri/Arcturan [Knight 97]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Your assignment, translate this to Arcturan:</strong></td>
</tr>
</tbody>
</table>
| 1a. ok-voon ororok sprok . | 7a. lalok farok ororok lalok sprok izok enemok .  
1b. at-voon bichat dat . | 7b. wat jjat bichat wat dat vat eneat .  
| 2a. ok-drubel ok-voon anok plok sprok . | 8a. lalok brok anok plok nok .  
2b. at-drubel at-voon pippat rrat dat . | 8b. iat lat pippat rrat nnat .  
| 3a. erok sprok izok hihok ghirok . | 9a. wiwok nok izok kantok ok-yurp .  
3b. totat dat arrat vat hilat . | 9b. totat nnat quat oloat at-yurp .  
| 4a. ok-voon anok drok brok jok . | 10a. lalok mok nok yorok ghirok clok .  
4b. at-voon krat pippat sat lat . | 10b. wat nnat gat mat bat hilat .  
| 5a. wiwok farok izok stok . | 11a. lalok nok crrrok hihok yorok zanzanok .  
5b. totat jjat quat cat . | 11b. wat nnat arrat mat zanzanat .  
| 6a. lalok sprok izok jok stok . | 12a. lalok rarok nok izok hihok mok .  
6b. wat dat krat quat cat . | 12b. wat nnat forat arrat vat gat .  

---

42/225  

Hal Daumé III, me@hal3.name  

NLP
### Centauri/Arcturan [Knight 97]

**Your assignment, translate this to Arcturan:**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. ok-voon ororok sprok .</td>
<td>7a. lalok farok ororok lalok sprok izok enemok .</td>
<td></td>
</tr>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
<td></td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
<td></td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
<td></td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
<td></td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
<td></td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok clok .</td>
<td>process of elimination</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
<td></td>
</tr>
<tr>
<td>5a. wiwok farok izok stok .</td>
<td>11a. lalok nok crrrok hihok yorok zanzanok .</td>
<td></td>
</tr>
<tr>
<td>5b. totat jjat quat cat .</td>
<td>11b. wat nnat arrat mat zanzanat .</td>
<td></td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
<td></td>
</tr>
<tr>
<td>6b. wat dat krat quat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
<td></td>
</tr>
</tbody>
</table>
### Centauri/Arcturan [Knight 97]

Your assignment, translate this to Arcturan:

<table>
<thead>
<tr>
<th>1a. ok-voon ororok sprok .</th>
<th>7a. lalok farok ororok lalok sprok izok enemok .</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnatquat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok clok .</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok farok izok stok .</td>
<td>11a. lalok nok [cerrrok hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjat quat cat .</td>
<td>11b. wat nnat arrat mat zanzan .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat krat quat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
**Centauri/Arcturan [Knight 97]**

Your assignment, translate this to Arcturan: `farok errrok hihok yorok clok kantok ok-yurp`

<table>
<thead>
<tr>
<th>1a. ok-voon ororok sprok .</th>
<th>7a. lalok farok ororok lalok sprok izok enemok .</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b. at-voon bichat dat .</td>
<td>7b. wat jjat bichat wat dat vat eneat .</td>
</tr>
<tr>
<td>2a. ok-drubel ok-voon anok plok sprok .</td>
<td>8a. lalok brok anok plok nok .</td>
</tr>
<tr>
<td>2b. at-drubel at-voon pippat rrat dat .</td>
<td>8b. iat lat pippat rrat nnat .</td>
</tr>
<tr>
<td>3a. erok sprok izok hihok ghirok .</td>
<td>9a. wiwok nok izok kantok ok-yurp .</td>
</tr>
<tr>
<td>3b. totat dat arrat vat hilat .</td>
<td>9b. totat nnat quat oloat at-yurp .</td>
</tr>
<tr>
<td>4a. ok-voon anok drok brok jok .</td>
<td>10a. lalok mok nok yorok ghirok .</td>
</tr>
<tr>
<td>4b. at-voon krat pippat sat lat .</td>
<td>10b. wat nnat gat mat bat hilat .</td>
</tr>
<tr>
<td>5a. wiwok farok izok stok .</td>
<td>11a. lalok nok [errrok] hihok yorok zanzanok .</td>
</tr>
<tr>
<td>5b. totat jjat quat cat .</td>
<td>11b. wat nnat arrat mat zanzana .</td>
</tr>
<tr>
<td>6a. lalok sprok izok jok stok .</td>
<td>12a. lalok rarok nok izok hihok mok .</td>
</tr>
<tr>
<td>6b. wat dat kratquat cat .</td>
<td>12b. wat nnat forat arrat vat gat .</td>
</tr>
</tbody>
</table>
Unsupervised EM Training

... la maison ..... la maison bleue ...... la fleur ...

... the house ..... the blue house ...... the flower ...

All P(french-word | english-word) equally likely
Unsupervised EM Training

... la maison ..... la maison bleue ...... la fleur ...

... the house ..... the blue house ...... the flower ...

“la” and “the” observed to co-occur frequently, so $P(\text{la} \mid \text{the})$ is increased.
Unsupervised EM Training

… la maison ..... la maison bleue ..... la fleur ...

... the house ..... the blue house ..... the flower ...

“maison” co-occurs with both “the” and “house”, but $P($maison $|$ house) can be raised without limit, to 1.0, while $P($maison $|$ the) is limited because of “la”

(pigeonhole principle)
Unsupervised EM Training

... la maison ..... la maison bleue ...... la fleur ...

\|
\|
\|
\|
\|

... the house ..... the blue house ...... the flower ...

settling down after another iteration
Unsupervised EM Training

… la maison ….. la maison bleue …… la fleur …

… the house ….. the blue house …… the flower …

Inherent hidden structure revealed by EM training!

“A Statistical MT Tutorial Workbook” (Knight, 1999). Promises free beer.


Software: GIZA++
The IBM model: “Brown93”

Mary did not slap the green witch

Mary not slap slap slap the green witch

Mary not slap slap slap NULL the green witch

Maria no daba una botefada a la verde bruja

Maria no daba una botefada a la bruja verde

Use the EM algorithm for training the parameters
Ready-to-use Data
Decoding for Machine Translation

Problem in NP-hard; use search:

Greedy Search

Beam Search

A* Search

Integer Programming
Progress in machine translation

insistent Wednesday may recurred her trips to Libya tomorrow for flying

Cairo 6-4 (AFP) - an official announced today in the Egyptian lines company for flying Tuesday is a company "insistent for flying" may resumed a consideration of a day Wednesday tomorrow her trips to Libya of Security Council decision trace international the imposed ban comment .

And said the official " the institution sent a speech to Ministry of Foreign Affairs of lifting on Libya air , a situation her receiving replying are so a trip will pull to Libya a morning Wednesday ".

Egyptair Has Tomorrow to Resume Its Flights to Libya

Cairo 4-6 (AFP) - said an official at the Egyptian Aviation Company today that the company egypair may resume as of tomorrow, Wednesday its flights to Libya after the International Security Council resolution to the suspension of the embargo imposed on Libya.

"The official said that the company had sent a letter to the Ministry of Foreign Affairs, information on the lifting of the air embargo on Libya, where it had received a response, the first take off a trip to Libya on Wednesday morning ".

Phrase-Based Translation

[ Koehn, Och and Marcu, NAACL03 ]

Australia is with North Korea

Australia is diplomatic relations with North Korea

is one of the few countries

is one of the few countries

Australia

North Korea

diplomatic relations

few countries
Training Phrase-Based MT Systems

[Koehn, Och and Marcu, NAACL03]
Decoding Phrase-Based MT

Maria no daba una botefada a la bruja verde

Mary did not slap the green witch

• Each step induces a cost attributed to:
  • Language model probability: \( p(\text{slap} \mid \text{did not}) \)
  • T-table probability: \( p(\text{the} \mid \text{a la}) \) and \( p(\text{a la} \mid \text{the}) \)
  • Distortion probability: \( p(\text{skip } l) \) [for a la --> verde]
  • Length penalty
  • ...
Hierarchical Phrase-Based MT

[Chiang, ACL05]

Australia is with North Korea, one of the few countries that have diplomatic relations with North Korea.
Hierarchical Phrase-Based MT

[Chiang, ACL05]
Syntax for MT

Kevin Knight, Daniel Marcu, Ignacio Thayer, Jonathan Graehl, Jon May, Steve DeNeefe
Syntax for MT

- **Decoding:**
  - Tree-to-tree/string automata
  - CKY parsing algorithm
  - Currently only unigram LM and trigram reranking
  - No syntax-based LM

- **Rule learning:**
  - Parsed English corpus
  - Aligned data (GIZA++)
  - Extract rules and assign probabilities

Phrase-based MT

BLEU

Syntax-based MT

Time
Traditional NLP

- The process of building computational models for understanding natural language

- **INPUT**: natural language text (or speech)

- **OUTPUT**: representation of the meaning of the text

- **Big Question**: What does “understand” mean? (NLU)
Examples of structured problems

Google Translate

This text has been automatically translated from Arabic:

Moscov stressed tone against Iran on its nuclear program. He called Russian Foreign Minister Tehran to take concrete steps to restore confidence with the international community, to cooperate fully with the IAEA. Conversely Tehran expressed its willingness.

from Arabic to English BETA

The man ate a tasty sandwich
Examples of demonstrations
Examples of demonstrations
## NLP as transduction

<table>
<thead>
<tr>
<th>Task</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine Translation</td>
<td>Ces deux principes se tiennent à la croisée de la philosophie, de la politique, de l’économie, de la sociologie et du droit.</td>
<td>Both principles lie at the crossroads of philosophy, politics, economics, sociology, and law.</td>
</tr>
<tr>
<td>Document Summarization</td>
<td>Argentina was still obsessed with the Falkland Islands even in 1994, 12 years after its defeat in the 74-day war with Britain. The country’s overriding foreign policy aim continued to be winning sovereignty over the islands.</td>
<td>The Falkland islands war, in 1982, was fought between Britain and Argentina.</td>
</tr>
<tr>
<td>Syntactic Analysis</td>
<td>The man ate a big sandwich.</td>
<td>The man ate a big sandwich</td>
</tr>
</tbody>
</table>

...many more...

07/19/12
Structured prediction 101

Learn a function mapping inputs to complex outputs:

\[ f : X \rightarrow Y \]
Structured prediction 101

Learn a function mapping inputs to complex outputs:

\[ f : X \rightarrow Y \]

Input Space  Decoding  Output Space

\[
\begin{array}{c}
\text{l} \\
\text{can} \\
\text{can} \\
\text{a} \\
\text{can}
\end{array}
\]
Why is structure important?

- Correlations among outputs
  - Determiners often precede nouns
  - Sentences usually have verbs

- Global coherence
  - It just *doesn't make sense* to have three determiners next to each other

- My objective (aka “loss function”) forces it
  - Translations should have good sequences of words
  - Summaries should be coherent
Outline: Part II

- Learning to Search
  - Incremental parsing
  - Learning to queue
- Refresher on Markov Decision Processes
- Inverse Reinforcement Learning
  - Determining rewards given policies
  - Maximum margin planning
- Learning by Demonstration
  - Searn
  - Dagger
- Discussion
Refresher on Binary Classification
What does it mean to learn?

- Informally:
  - to predict the future based on the past

- Slightly-less-informally:
  - to take *labeled examples* and construct a function that will label them as a human would

- Formally:
  - Given:
    - A fixed unknown distribution D over X*Y
    - A loss function over Y*Y
    - A finite sample of (x,y) pairs drawn i.i.d. from D
  - Construct a function f that has low expected loss with respect to D
Feature extractors

- A feature extractor $\Phi$ maps examples to vectors

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...

$\Phi$

- Feature vectors in NLP are frequently sparse
Linear models for binary classification

- Decision boundary is the set of “uncertain” points
- Linear decision boundaries are characterized by weight vectors

<table>
<thead>
<tr>
<th>$x$</th>
<th>$\Phi(x)$</th>
<th>$w$</th>
<th>$\sum_i w_i \Phi_i(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIAS</td>
<td>1</td>
<td>BIAS : -3</td>
<td>(1)(-3) +</td>
</tr>
<tr>
<td>free</td>
<td>1</td>
<td>free : 4</td>
<td>(1)(4) +</td>
</tr>
<tr>
<td>money</td>
<td>1</td>
<td>money : 2</td>
<td>(1)(2) +</td>
</tr>
<tr>
<td>the</td>
<td>0</td>
<td>the : 0</td>
<td>(0)(0) +</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

“free money”
The perceptron

- Inputs = feature values
- Params = weights
- Sum is the response

- If the response is:
  - Positive, output +1
  - Negative, output -1

- When training, update on errors:
  \[ \mathbf{w} = \mathbf{w} + y \phi(x) \]

"Error" when:
\[ y \mathbf{w} \cdot \phi(x) \leq 0 \]
Why does that update work?

- When $y w^{old} \cdot \phi(x) \leq 0$, update $w^{new} = w^{old} + y \phi(x)$

\[
y w^{new} \phi(x) = y \left( w^{old} + y \phi(x) \right) \phi(x) \\
= y w^{old} \phi(x) + yy \phi(x) \phi(x)
\]

\[<0 \quad >\]
Support vector machines

- Explicitly optimize the \textit{margin}.

- Enforce that all training points are correctly classified.

\[\begin{align*}
\max_w \text{ margin} & \quad s.t. \quad \text{all points are correctly classified} \\
\max_w \text{ margin} & \quad s.t. \quad y_n w \cdot \phi(x_n) \geq 1 \quad \forall n \\
\min_w \|w\|^2 & \quad s.t. \quad y_n w \cdot \phi(x_n) \geq 1 \quad \forall n
\end{align*}\]
Support vector machines with \textit{slack}

- Explicitly optimize the \textit{margin}

- Allow some “noisy” points to be misclassified

\[
\min_{w, \xi} \quad \frac{1}{2} \|w\|^2 + C \sum_n \xi_n \\
\text{subject to} \quad y_n w \cdot \phi(x_n) + \xi_n \geq 1, \quad \forall n \\
\xi_n \geq 0, \quad \forall n
\]
Batch versus stochastic optimization

- Batch = read in all the data, then process it
- Stochastic = (roughly) process a bit at a time

\[
\begin{align*}
\min_{w,\xi} & \quad \frac{1}{2} \|w\|^2 + C \sum_n \xi_n \\
\text{s.t.} & \quad y_n w \cdot \phi(x_n) + \xi_n \geq 1 \\
& \quad \xi_n \geq 0, \quad \forall n
\end{align*}
\]

- For n=1..N:
  - If \( y_n w \cdot \phi(x_n) \leq 0 \)
  - \( w = w + y_n \phi(x_n) \)
Stochastically optimized SVMs

SVM Objective

For n=1..N:
- If \( y_n w \cdot \phi(x_n) \leq 1 \)
  - \( w = w + y_n \phi(x_n) \)
- \( w = \left(1 - \frac{1}{CN}\right)w \)

Implementation Note:
Weight shrinkage is SLOW. Implement it lazily, at the cost of double storage.

For n=1..N:
- If \( y_n w \cdot \phi(x_n) \leq 0 \)
  - \( w = w + y_n \phi(x_n) \)
From Perceptron to Structured Perceptron
Perceptron with multiple classes

- Store separate weight vector for each class $w_1, w_2, \ldots, w_K$

- For $n=1..N$:
  - Predict:
    \[
    \hat{y} = \arg \max_k w_k \cdot \phi(x_n)
    \]
  - If $\hat{y} \neq y_n$
    \[
    \begin{align*}
    w_{\hat{y}} &= w_{\hat{y}} - \phi(x_n) \\
    w_{y_n} &= w_{y_n} + \phi(x_n)
    \end{align*}
    \]

?: Why does this do the right thing?
Perceptron with multiple classes v2

- Originally:

\[
\begin{array}{ccc}
\mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 \\
\hline
\mathbf{W}
\end{array}
\]

- For $n=1..N$:
  - Predict:
    \[
    \hat{y} = \arg \max_k \mathbf{w}_k \cdot \phi(\mathbf{x}_n)
    \]
  - If $\hat{y} \neq y_n$
    \[
    \mathbf{w}_{\hat{y}} = \mathbf{w}_{\hat{y}} - \phi(\mathbf{x}_n)
    \]
    \[
    \mathbf{w}_{y_n} = \mathbf{w}_{y_n} + \phi(\mathbf{x}_n)
    \]

- For $n=1..N$:
  - Predict:
    \[
    \hat{y} = \arg \max_k \mathbf{w} \cdot \phi(\mathbf{x}_n, k)
    \]
  - If $\hat{y} \neq y_n$
    \[
    \mathbf{w} = \mathbf{w} - \phi(\mathbf{x}_n, \hat{y}) + \phi(\mathbf{x}_n, y_n)
    \]
Perceptron

- Originally:
  - \( \text{"free money"} \)

- For \( n=1..N \):
  - Predict:
    - \( \hat{y} = \arg \max_k w_k \cdot \phi(x_n) \)
    - If \( \hat{y} \neq y_n \):
      - \( w_{\hat{y}} = w_{\hat{y}} - \phi(x_n) \)
      - \( w_{y_n} = w_{y_n} + \phi(x_n) \)

- For \( n=1..N \):
  - Predict:
    - \( \hat{y} = \arg \max_k w \cdot \phi(x_n, k) \)
    - If \( \hat{y} \neq y_n \):
      - \( w = w - \phi(x_n, \hat{y}) + \phi(x_n, y_n) \)
Features for structured prediction

- Allowed to encode *anything* you want

\[ \phi(x, y) = \]

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I_Pro</td>
<td>1</td>
<td>&lt;s&gt;-Pro</td>
</tr>
<tr>
<td>can_Md</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>can_Vb</td>
<td>1</td>
<td>Pro-Md</td>
</tr>
<tr>
<td>a_Dt</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>can_Nn</td>
<td>1</td>
<td>Md-Vb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vb-Dt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dt-Nn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nn-(&lt;/s&gt;)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Output features, Markov features, other features
Structured perceptron

For n=1..N:
  • Predict:
    \[ \hat{y} = \text{arg max}_k w \cdot \phi(x_n, k) \]
  • If \( \hat{y} \neq y_n \):
    \[
    w = w - \phi(x_n, \hat{y}) + \phi(x_n, y_n)
    \]

Collins, EMNLP02
Argmax for sequences

- If we only have output and Markov features, we can use Viterbi algorithm:

(plus some work to account for boundary conditions)
Structured perceptron as ranking

- For n=1..N:
  - Run Viterbi: \( \hat{y} = \arg \max_k w \cdot \phi(x_n, k) \)
  - If \( \hat{y} \neq y_n \):
    \[ w = w - \phi(x_n, \hat{y}) + \phi(x_n, y_n) \]

- When does this make an update?

<table>
<thead>
<tr>
<th>Pro</th>
<th>Md</th>
<th>Vb</th>
<th>Dt</th>
<th>Nn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Md</td>
<td>Dt</td>
<td>Vb</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Nn</td>
<td>Dt</td>
<td>Md</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Nn</td>
<td>Dt</td>
<td>Nn</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Vb</td>
<td>Dt</td>
<td>Md</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Vb</td>
<td>Dt</td>
<td>Vb</td>
</tr>
<tr>
<td>I</td>
<td>can</td>
<td>can</td>
<td>a</td>
<td>can</td>
</tr>
</tbody>
</table>
From perceptron to margins

Maximize Margin
\[ \min_{\mathbf{w}, \xi} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_n \xi_n \]

Minimize Errors
\[ s.t. \quad y_n \mathbf{w} \cdot \phi(x_n) + \xi_n \geq 1, \quad \forall n \]

Each point is correctly classified, modulo \( \xi \)

\[ \min_{\mathbf{w}, \xi} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_n \xi_n, \hat{y} \]

Response for truth
\[ s.t. \quad \mathbf{w} \cdot \phi(x_n, y_n) \]

Response for other
\[ -\mathbf{w} \cdot \phi(x_n, \hat{y}) + \xi_n \geq 1, \quad \forall n, \hat{y} \neq y_n \]

Each true output is more highly ranked, modulo \( \xi \)
From perceptron to margins

\[ \min_{w, \xi} \frac{1}{2} \|w\|^2 + C \sum_n \xi_n, \hat{y} \]

Response for truth

\[ s.t. \quad w \cdot \phi(x_n, y_n) \]
\[ - w \cdot \phi(x_n, \hat{y}) \]
\[ + \xi_n \geq 1, \forall n, \hat{y} \neq y_n \]

Response for other

Each true output is more highly ranked, modulo \( \xi \)
### Ranking margins

- Some errors are worse than others...

<table>
<thead>
<tr>
<th>Pro</th>
<th>Md</th>
<th>Vb</th>
<th>Dt</th>
<th>Nn</th>
</tr>
</thead>
</table>

**Margin of one**

<table>
<thead>
<tr>
<th>Pro</th>
<th>Md</th>
<th>Md</th>
<th>Dt</th>
<th>Vb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Md</td>
<td>Dt</td>
<td>Nn</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Nn</td>
<td>Dt</td>
<td>Md</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Nn</td>
<td>Dt</td>
<td>Nn</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Vb</td>
<td>Dt</td>
<td>Md</td>
</tr>
<tr>
<td>Pro</td>
<td>Md</td>
<td>Vb</td>
<td>Dt</td>
<td>Vb</td>
</tr>
<tr>
<td>I</td>
<td>can</td>
<td>can</td>
<td>a</td>
<td>can</td>
</tr>
</tbody>
</table>

[Taskar+ al, JMLR05; Tshochandaritis, JMLR05]
Accounting for a loss function

- Some errors are worse than others...

Margin of $l(y,y')$
Accounting for a loss function

\[ \forall \hat{y}, \quad w \cdot \phi(x_n, y_n) - w \cdot \phi(x_n, \hat{y}) + \xi_n \geq l(y_n, \hat{y}) \]

is equivalent to

\[ \max_{\hat{y}} \quad w \cdot \phi(x_n, y_n) - w \cdot \phi(x_n, \hat{y}) + \xi_n \geq l(y_n, \hat{y}) \]

\[ w \cdot \phi(x_n, y_n) - w \cdot \phi(x_n, \hat{y}) + \xi_n \geq 1 \]

\[ w \cdot \phi(x_n, y_n) - w \cdot \phi(x_n, \hat{y}) + \xi_n \geq l(y_n, \hat{y}) \]
Augmented argmax for sequences

- Add “loss” to each wrong node!

What are we assuming here?
**M³N Objective**

For $n=1..N$:
- Augmented Viterbi:
  \[
  \hat{y} = \text{arg max}_k w \cdot \phi(x_n, k) + l(y_n, k)
  \]
  
  \[
  \text{If } \hat{y} \neq y_n:
  w = w - \phi(x_n, \hat{y}) + \phi(x_n, y_n)
  \]

\[
  w = \left(1 - \frac{1}{CN}\right)w
  \]

For $n=1..N$:
- Viterbi:
  \[
  \hat{y} = \text{arg max}_k w \cdot \phi(x_n, k)
  \]
  
  \[
  \text{If } \hat{y} \neq y_n:
  w = w - \phi(x_n, \hat{y}) + \phi(x_n, y_n)
  \]

[Ratliff+al, AIStats07]
Stacking

- **Structured models:** accurate but slow
  ![Diagram of structured models]

- **Independent models:** less accurate but fast
  ![Diagram of independent models]

- **Stacking:** multiple independent models
  ![Diagram of stacking]

07/19/12
Training a stacked model

- Train independent classifier $f_1$ on input features

- Train independent classifier $f_2$ on input features + $f_1$'s output

- **Danger:** overfitting!

- **Solution:** cross-validation

07/19/12
Do we really need structure?

- **Structured models**: accurate but slow
  ![Structured Model Diagram](https://example.com/structured_model.png)

- **Independent models**: less accurate but fast
  ![Independent Model Diagram](https://example.com/independent_model.png)

- **Goal**: transfer power to get fast+accurate
  ![Goal Model Diagram](https://example.com/goal_model.png)

- **Questions**: are independent models...
  - ... expressive enough? (approximation error)
  - ... easy to learn? (estimation error)
“Compiling” structure out

\[ \text{Labeled Data} \]

\[ \text{CRF}(f_1) \quad \text{POS: } 95.0\% \quad \text{NER: } 75.3\% \]

\[ \text{IRL}(f_1) \quad \text{POS: } 91.7\% \quad \text{NER: } 69.1\% \]

\[ f_1 = \text{words/prefixes/suffixes/forms} \]

07/19/12
“Compiling” structure out

CRF($f_1$)  POS:  95.0%  
NER:  75.3%

$\begin{align*}
f_1 &= \text{words/prefixes/suffixes/forms} \\
f_2 &= f_1 \text{ applied to a larger window}
\end{align*}$

IRL($f_1$)  POS:  91.7%  
NER:  69.1%

IRL($f_2$)  POS:  94.4%  
NER:  66.2%

CompIRL($f_2$)  POS:  95.0%  
NER:  72.7%

07/19/12
Decomposition of errors

\[ \text{CRF}(f_1): \ p_c \]

\[ \text{coherence} \]

\[ \text{marginalized CR} \]

\[ \text{nonlinearities} \]

\[ \text{global information} \]

\[ \text{Theorem:} \quad \KL(p_C \parallel p_{1*}) = \KL(p_C \parallel p_{MC}) + \KL(p_{MC} \parallel p_{A*}) + \KL(p_{A*} \parallel p_{1*}) \]

Sum of MI on edges

POS = 0.003 (95.0% → 95.0%)

NER = 0.009 (76.3% → 76.0%)

Train a truncated CRF

NER: 76.0% → 72.7%

Train a marginalized CRF

NER: 76.0% → 76.0%

Liang+D+Klein, ICML08
Structure compilation results

Part of speech

Named Entity

Parsing

- Structured
- Independent

[Liang+D+Klein, ICML08]

07/19/12
Learning to Search
Argmax is *hard!*

- Classic formulation of structured prediction:

  \[ \text{score}(x, y) \]

  *something we learn to make “good” x,y pairs score highly*

- At test time:

  \[ f(x) = \arg\max_{y \in Y} \text{score}(x, y) \]

- Combinatorial optimization problem
  - Efficient only in very limiting cases
  - Solved by heuristic search: beam + A* + local search

07/19/12
Argmax is *hard*!

- Classical \[ \text{argmax}_x \text{score}(x, y) \]

  
  \[ f(x) \]

  
  \[ \text{Order these words: bart better I madonna say than} \]

- At test time \[ \text{argmax}_x \text{score}(x, y) \]

  
  \[ f(x) \]

- Combinatorial optimization problem
  - Efficient only in very limiting cases
  - Solved by heuristic search: beam + A* + local search

07/19/12

Argmax is hard!

- Classical optimization
  \[ \text{score}(x, y) \]

- At test time
  \[ f(x) \]

- Combinatorial optimization problem
  \[ \text{argmax}_y \text{score}(x, y) \]

- Efficient only in very limiting cases
- Solved by heuristic search: beam + A* + local search

Order these words: bart better I madonna say than ,
Best search (32.3): I say better than bart madonna ,
Original (41.6): better bart than madonna , I say


07/19/12
Argmax is **hard**!

- Classical \( f \)
- At test time \( \text{score}(x, y) \)
- Combinatorial optimization problem
  - Efficient only in very limiting cases
  - Solved by heuristic search: beam + A* + local search

*Order these words:* bart better I madonna say than ,

*Best search (32.3):* I say better than bart madonna ,

*Original (41.6):* better bart than madonna , I say

*Best search (51.6):* and so could really be a neural apparently thought things as dissimilar firing two identical


07/19/12
Argmax is hard!

- **Classical:**

  \[
  f(x) = \text{argmax}_y \text{score}(x, y)
  \]

  \[
  \text{score}(x, y) = \text{Best search (32.3)}: \text{I say better than bart madonna}, \text{I say}
  \]

  \[
  \text{Original (41.6): better bart than madonna, I say}
  \]

  \[
  \text{Best search (51.6): and so could really be a neural apparently thought things as dissimilar firing two identical}
  \]

  \[
  \text{Original (64.3): could two things so apparently dissimilar as a thought and neural firing really be identical}
  \]

  \[
  \text{[Soricut, PhD Thesis, USC 2007]}
  \]

- **At test:**

  \[
  f(x) = \text{argmax}_y \text{score}(x, y)
  \]

  \[
  \text{Efficient only in very limiting cases}
  \]

  \[
  \text{Solved by heuristic search: beam + A* + local search}
  \]

  \[
  07/19/12
  \]
Train a classifier to make decisions

Incremental parsing, early 90s style

[Magerman, ACL95]
Incremental parsing, mid 2000s style

Train a classifier to make decisions

[Collins+Roark, ACL04]
Learning to beam-search

For $n = 1..N$:
- Viterbi:
  \[ \hat{y} = \arg\max_k w \cdot \phi(x_n, k) \]
- If $\hat{y} \neq y_n$:
  \[ w = w - \phi(x_n, \hat{y}) + \phi(x_n, y_n) \]

[Collins+Roark, ACL04]
Learning to beam-search

- For n=1..N:
  - Run beam search until truth falls out of beam
  - Update weights immediately!
Learning to beam-search

For n=1..N:
- Run beam search until truth falls out of beam
- Update weights immediately!
- Restart at truth
Incremental parsing results

- □ - No early update, no repeated use of examples
- ○ - Early update, no repeated use of examples
- ■ - Early update, repeated use of examples

F-measure parsing accuracy

Number of passes over training data

[Collins+Roark, ACL04]
Generic Search Formulation

- Search Problem:
  - Search space
  - Operators
  - Goal-test function
  - Path-cost function

- Search Variable:
  - Enqueue function

Varying the **enqueue** function can give us DFS, BFS, beam search, A* search, etc...

- nodes := MakeQueue(S0)

- while nodes is not empty
  - node := RemoveFront(nodes)
  - if node is a goal state
    - return node
  - next := Operators(node)
  - nodes := Enqueue(nodes, next)

- fail
Online Learning Framework (LaSO)

- \( \text{nodes} := \text{MakeQueue}(S0) \)
- \( \textbf{while} \text{ nodes is not empty} \)
  - \( \text{node} := \text{RemoveFront}(\text{nodes}) \)
  - \( \text{if none of } \{\text{node}\} \cup \text{nodes is y-good or node is a goal & not y-good} \)
    - \( \text{sibs} := \text{siblings(node, y)} \)
    - \( w := \text{update}(w, x, \text{sibs}, \{\text{node}\} \cup \text{nodes}) \)
    - \( \text{nodes} := \text{MakeQueue}(\text{sibs}) \)
  - \( \text{else} \)
    - \( \text{if node is a goal} \)
      - \( \text{next} := \text{Operators}(\text{node}) \)
      - \( \text{nodes} := \text{Enqueue}(\text{nodes, next}) \)

\( \text{Monotonicity: for any node, we can tell if it can lead to the correct solution or not} \)

If we erred...

Where should we have gone?

Update our weights based on the good and the bad choices

Continue search...

[D+Marcu, ICML05; Xu+al, JMLR09]
Search-based Margin

- The margin is the amount by which we are correct:

- Note that the margin and hence linear separability is also a function of the search algorithm!
Syntactic chunking Results

- Large Margin (beam 25)
- Large Margin (Exact)
- Perceptron Search (Exact)
- Best prior results
- Standard Perceptron Updates
- Semi-CRF model

- [Zhang+Damerau+Johnson 2002]; timing unknown
- [Collins 2002]

- 4 min
- 24 min
- 22 min
- 33 min
Tagging+chunking results

Joint tagging/chunking accuracy vs. Training Time (hours) [log scale]

- **Large Margin (beam 25/50)**: 23 min
- **Large Margin (beam 10)**: 7 min
- **Sutton model**: 3 min

[D+Marcu, ICML05; Xu+al, JMLR09]

*Sutton*
Variations on a beam

- Observation:
  - We needn't use the same beam size for training and decoding
  - Varying these values independently yields:

<table>
<thead>
<tr>
<th>Training Beam</th>
<th>Decoding Beam</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>25</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>93.9</td>
<td>92.8</td>
<td>91.9</td>
<td>91.3</td>
<td>90.9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>90.5</td>
<td>94.3</td>
<td>94.4</td>
<td>94.1</td>
<td>94.1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>89.5</td>
<td>94.3</td>
<td>94.4</td>
<td>94.2</td>
<td>94.2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>88.7</td>
<td>94.2</td>
<td>94.5</td>
<td>94.3</td>
<td>94.3</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>88.4</td>
<td>94.2</td>
<td>94.4</td>
<td>94.2</td>
<td>94.4</td>
<td></td>
</tr>
</tbody>
</table>

[D+Marcu, ICML05; Xu+al, JMLR09]
What if our model sucks?

- Sometimes our model *cannot* produce the “correct” output
  - canonical example: machine translation

![Diagram of model outputs and updates]

- N-best list or “optimal decoding” or ...
- Best achievable output
- “Local” update
- “Bold” update
- Reference
- Good Outputs
- Model Outputs
- Current Hypothesis
Local versus bold updating...

Machine Translation Performance (Bleu)
Refresher on Markov Decision Processes
Reinforcement learning

- **Basic idea:**
  - Receive feedback in the form of rewards
  - Agent’s utility is defined by the reward function
  - Must learn to act to maximize expected rewards
  - Change the rewards, change the learned behavior

- **Examples:**
  - Playing a game, reward at the end for outcome
  - Vacuuming, reward for each piece of dirt picked up
  - Driving a taxi, reward for each passenger delivered
Markov decision processes

What are the values (expected future rewards) of states and actions?

\[ V(s)^* = 30 \]

\[ Q(s,a_1)^* = 30 \]

\[ Q(s,a_2)^* = 23 \]

\[ Q(s,a_3)^* = 17 \]
Markov Decision Processes

- An MDP is defined by:
  - A set of states \( s \in S \)
  - A set of actions \( a \in A \)
  - A transition function \( T(s, a, s') \)
  - Prob that \( a \) from \( s \) leads to \( s' \)
  - i.e., \( P(s' \mid s, a) \)
  - Also called the model
  - A reward function \( R(s, a, s') \)
  - Sometimes just \( R(s) \) or \( R(s') \)
  - A start state (or distribution)
  - Maybe a terminal state

- MDPs are a family of non-deterministic search problems
- Total utility is one of:
  \[
  \sum_{t} r_t \quad \text{or} \quad \sum_{t} \gamma^t r_t
  \]
Solving MDPs

- In deterministic single-agent search problem, want an optimal plan, or sequence of actions, from start to a goal
- In an MDP, we want an optimal policy $\pi(s)$
  - A policy gives an action for each state
  - Optimal policy maximizes expected if followed
  - Defines a reflex agent

Optimal policy when $R(s, a, s') = -0.04$ for all non-terminals $s$
Example Optimal Policies

\[ R(s) = -0.01 \]

\[ R(s) = -0.03 \]

\[ R(s) = -0.4 \]

\[ R(s) = -2.0 \]
Optimal Utilities

- Fundamental operation: compute the optimal utilities of states $s$ (all at once)

- Why? Optimal values define optimal policies!

- Define the utility of a state $s$: $V^*(s) = \text{expected return starting in } s \text{ and acting optimally}$

- Define the utility of a q-state $(s,a)$: $Q^*(s,a) = \text{expected return starting in } s, \text{ taking action } a \text{ and thereafter acting optimally}$

- Define the optimal policy: $\pi^*(s) = \text{optimal action from state } s$
The Bellman Equations

- Definition of utility leads to a simple one-step lookahead relationship amongst optimal utility values:
  
  Optimal rewards = maximize over first action and then follow optimal policy

- Formally:

  $$V^*(s) = \max_a Q^*(s, a)$$

  $$Q^*(s, a) = \sum_{s'} T(s, a, s') \left[ R(s, a, s') + \gamma V^*(s') \right]$$

  $$V^*(s) = \max_a \sum_{s'} T(s, a, s') \left[ R(s, a, s') + \gamma V^*(s') \right]$$
Solving MDPs / memoized recursion

- Recurrences:

\[ V_0^*(s) = 0 \]

\[ V_i^*(s) = \max_a Q_i^*(s, a) \]

\[ Q_i^*(s, a) = \sum_{s'} T(s, a, s') \left[ R(s, a, s') + \gamma V_{i-1}^*(s') \right] \]

\[ \pi_i(s) = \arg\max_a Q_i^*(s, a) \]

- Cache all function call results so you never repeat work
- What happened to the evaluation function?
Q-Value Iteration

- Value iteration: iterate approx optimal values
  - Start with $V_0^*(s) = 0$, which we know is right (why?)
  - Given $V_i^*$, calculate the values for all states for depth $i+1$:

\[
V_{i+1}(s) \leftarrow \max_a \sum_{s'} T(s, a, s') \left[ R(s, a, s') + \gamma V_i(s') \right]
\]

- But Q-values are more useful!
  - Start with $Q_0^*(s,a) = 0$, which we know is right (why?)
  - Given $Q_i^*$, calculate the q-values for all q-states for depth $i+1$:

\[
Q_{i+1}(s, a) \leftarrow \sum_{s'} T(s, a, s') \left[ R(s, a, s') + \gamma \max_{a'} Q_i(s', a') \right]
\]
RL = Unknown MDPs

- If we *knew* the MDP (i.e., the reward function and transition function):
  - Value iteration leads to optimal values
  - Will always converge to the truth

- Reinforcement learning is what we do when we *do not know* the MDP
  - All we observe is a *trajectory*
  - \((s_1,a_1,r_1, s_2,a_2,r_2, s_3,a_3,r_3, \ldots)\)

- Many algorithms exist for this problem; see Sutton+Barto's excellent book!
Q-Learning

- Learn $Q^*(s,a)$ values
  - Receive a sample $(s,a,s',r)$
  - Consider your old estimate: $Q(s, a)$
  - Consider your new sample estimate:
    \[
    Q^*(s, a) = \sum_{s'} T(s, a, s') \left[ R(s, a, s') + \gamma \max_{a'} Q^*(s', a') \right]
    \]

- Incorporate the new estimate into a running average:
  \[
  \text{sample} = R(s, a, s') + \gamma \max_{a'} Q(s', a')
  \]
  \[
  Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + (\alpha) \text{[sample]}
  \]
Exploration / Exploitation

- Several schemes for forcing exploration
  - Simplest: random actions (ε greedy)
    - Every time step, flip a coin
    - With probability ε, act randomly
    - With probability 1-ε, act according to current policy

- Problems with random actions?
  - You do explore the space, but keep thrashing around once learning is done
  - One solution: lower ε over time
  - Another solution: exploration functions
Q-Learning

- In realistic situations, we cannot possibly learn about every single state!
  - Too many states to visit them all in training
  - Too many states to hold the q-tables in memory

- Instead, we want to generalize:
  - Learn about some small number of training states from experience
  - Generalize that experience to new, similar states:

  \[ Q(s, a) = w_1 f_1(s, a) + w_2 f_2(s, a) + \ldots + w_n f_n(s, a) \]

- Very simple stochastic updates:

  \[ Q(s, a) \leftarrow Q(s, a) + \alpha [error] \]
  \[ w_i \leftarrow w_i + \alpha [error] f_i(s, a) \]
Inverse Reinforcement Learning

(aka Inverse Optimal Control)
Inverse RL: Task

- Given:
  - measurements of an agent's behavior over time, in a variety of circumstances
  - if needed, measurements of the sensory inputs to that agent
  - if available, a model of the environment.

- Determine: the reward function being optimized

- Proposed by [Kalman68]
- First solution, by [Boyd94]
Why inverse RL?

- Computational models for animal learning
  - “In examining animal and human behavior we must consider the reward function as an unknown to be ascertained through empirical investigation.”

- Agent construction
  - “An agent designer [...] may only have a very rough idea of the reward function whose optimization would generate 'desirable' behavior.”
  - eg., “Driving well”

- Multi-agent systems and mechanism design
  - learning opponents’ reward functions that guide their actions to devise strategies against them
IRL from Sample Trajectories

- Optimal policy available through sample behavior (e.g., driving a car)

- Want to find Reward function that makes this policy look as good as possible

- Write $R_w(s) = w \phi(s)$ so the reward is linear

and $V^\pi_w(s_0)$ be the value of the starting state

$$\max_w \sum_{k=1}^{K} f \left( V^\pi_w(s_0) - V^\pi_{w_k}(s_0) \right)$$

How good does the “optimal policy” look?

How good does the some other policy look?

[Ng+Russell, ICML00]
Apprenticeship Learning via IRL

- For \( t = 1, 2, \ldots \)
  - Inverse RL step:
    Estimate expert’s reward function \( R(s) = w^T \phi(s) \) such that under \( R(s) \) the expert performs better than all previously found policies \( \{\pi_i\} \).

- RL step:
  Compute optimal policy \( \pi_t \) for the estimated reward \( w \).
Car Driving Experiment

- No explicit reward function at all!
- Expert demonstrates proper policy via 2 min. of driving time on simulator (1200 data points).
- 5 different “driver types” tried.
- Features: which lane the car is in, distance to closest car in current lane.
- Algorithm run for 30 iterations, policy hand-picked.
- Movie Time! (Expert left, IRL right)
“Nice” driver
“Evil” driver
Maxent IRL

Distribution over trajectories:

\[ P(\zeta) \]

Match the reward of observed behavior:

\[ \sum_{\zeta} P(\zeta) f_\zeta = f_{\text{dem}} \]

Maximizing the **causal entropy** over trajectories given stochastic outcomes:

\[ \max H(P(\zeta) || O) \]

(Condition on random uncontrolled outcomes, but only after they happen)
Data collection

25 Taxi Drivers

Over 100,000 miles

Length
Speed
Road
Type
Lanes

Accidents
Construction
Congestion
Time of day

[Ziebart+al, AAAI08]
Predicting destinations....
Planning as structured prediction

[Ratliff+al, NIPS05]
Maximum margin planning

- Let $\mu(s,a)$ denote the probability of reaching q-state $(s,a)$ under current model $w$

$$\begin{align*}
\max_w & \text{ margin } \quad \text{s.t. } \quad \text{planner run with } w \\
\text{yields human output} & \\
\min_w & \frac{1}{2} \|w\|^2 \quad \text{s.t. } \quad \mu(s,a)w \cdot \phi(x_n,s,a) \\
\text{Q-state visitation frequency by human} & \\
\mu(s,a)w \cdot \phi(x_n,s,a) \geq 1 \quad , \quad \forall n,s,a \quad , \\
\text{Q-state visitation frequency by planner} \quad & \\
\text{All trajectories, and all q-states} & \\
\end{align*}$$
Optimizing MMP

MMP Objective

SOME MATH

For $n=1..N$:

- Augmented planning:
  
  Run $A^*$ on current (augmented) cost map to get $q$-state visitation frequencies $\mu(s, a)$

- Update: $w = w + \sum_s \sum_a \left[ \hat{\mu}(s, a) - \mu(s, a) \right] \phi(x_n, s, a)$

- Shrink: $w = \left( 1 - \frac{1}{CN} \right) w$
Maximum margin planning movies

[Ratliff+al, NIPS05]
Parsing via inverse optimal control

- State space = all partial parse trees over the full sentence labeled “S”
- Actions: take a partial parse and split it anywhere in the middle
- Transitions: obvious
- Terminal states: when there are no actions left
- Reward: parse score at completion

[Neu+Szepevari, MLJ09]
Parsing via inverse optimal control

[Neu+Szepevari, MLJ09]
Learning by Demonstration
Integrating search and learning

Input: Le homme mange l' croissant.
Output: The man ate a croissant.

Hyp: The man ate
Cov: Le homme mange l' croissant.

Classifier 'h'

Hyp: The man ate a fox
Cov: Le homme mange l' croissant.

Hyp: The man ate happy
Cov: Le homme mange l' croissant.

Hyp: The man ate a
Cov: Le homme mange l' croissant.

[D+Marcu, ICML05; D+Langford+Marcu, MLJ09]
Reducing search to classification

- Natural chicken and egg problem:
  - Want $h$ to get low expected future loss
  - ... on future decisions made by $h$
  - ... and starting from states visited by $h$

- Iterative solution

Input: Le homme mange l' croissant.
Output: The man ate a croissant.

$\text{Hyp: The man ate a croissant} \\
\text{Cov: Le homme mange l' croissant.}$

Loss = 0

$\text{Hyp: The man ate a fox} \\
\text{Cov: Le homme mange l' croissant.}$

Loss = 1.8

$\text{Hyp: The man ate happy} \\
\text{Cov: Le homme mange l' croissant.}$

Loss = 1.2

$\text{Hyp: The man ate a} \\
\text{Cov: Le homme mange l' croissant.}$

Loss = 0.5

$\text{Hyp: The man ate} \\
\text{Cov: Le homme mange l' croissant.}$

Loss = 0

07/19/12

[D+Langford+Marcu, MLJ09]
Reduction for Structured Prediction

- Idea: view structured prediction in light of search

Each step here looks like it could be represented as a weighted multi-class problem.

Can we formalize this idea?

Loss function:
\[
\begin{align*}
L([N V R], [N V R]) &= 0 \\
L([N V R], [N V V]) &= 1/3 \\
&\vdots
\end{align*}
\]
Reducing Structured Prediction

Desired: good policy on test data (i.e., given only input string)

Key Assumption: Optimal Policy for training data

Given: input, true output and state;
Return: best successor state

Weak!
**How to Learn in Search**

**Idea:** Train based only on optimal path (ala MEMM)

**Better Idea:** Train based only on optimal policy, then train based on optimal policy + a little learned policy, then train based on optimal policy + a little more learned policy, then ... eventually only use learned policy
How to Learn in Search

- Translating $D^S_P$ into $Searn(D^S_P, \text{loss}, \pi)$:
  - Draw $x \sim D^S_P$
  - Run $\pi$ on $x$, to get a path
  - Pick position uniformly on path
  - Generate example with costs given by expected (wrt $\pi$) completion costs for “loss”
Algorithm: Searn-Learn($A$, $D^{SP}$, loss, $\pi^*$, $\beta$)
1: Initialize: $\pi = \pi^*$
2: while not converged do
3: Sample: $D \sim \text{Searn}(D^{SP}, \text{loss}, \pi)$
4: Learn: $h \leftarrow A(D)$
5: Update: $\pi \leftarrow (1-\beta) \pi + \beta h$
6: end while
7: return $\pi$ without $\pi^*$

Ingredients for Searn:
Input space ($X$) and output space ($Y$), data from $X$
Loss function ($\text{loss}(y, y')$) and features
“Optimal” policy $\pi^*(x, y_0)$

Theorem: $L(\pi) \leq L(\pi^*) + \text{avg} \text{(loss)} \cdot T \ln T + c(1+\ln T) / T$
But what about demonstrations?

- What did we assume before?

**Key Assumption:** *Optimal Policy for training data*

Given: input, true output and state; Return: best successor state

- We can have a *human* (or system) demonstrate, thus giving us an *optimal policy*
3d racing game (TuxKart)

Input:

Resized to 25x19 pixels (1425 features)

Output:

Steering in [-1,1]
DAgger: Dataset Aggregation

- Collect trajectories from expert $\pi^*$
- Dataset $D_0 = \{ (s, \pi^*(s)) \mid s \sim \pi^* \}$
- Train $\pi_1$ on $D_0$
- Collect new trajectories from $\pi_1$
  - But let the expert steer!
- Dataset $D_1 = \{ (s, \pi^*(s)) \mid s \sim \pi_1 \}$
- Train $\pi_2$ on $D_0 \cup D_1$

- In general:
  - $D_n = \{ (s, \pi^*(s)) \mid s \sim \pi_n \}$
  - Train $\pi_n$ on $\cup_{i<n} D_i$

If $N = T \log T$, $L(\pi_n) < T \epsilon_N + O(1)$ for some $n$
Experiments: Racing Game

Input:

![Image of a racing game scene](Features)

Resized to 25x19 pixels (1425 features)

Output:

Steering in [-1,1]
Average falls per lap

![Graph showing the average falls per lap against the number of training data points for different algorithms: DAgger, SMILe(0.1), and Supervised. The graph indicates that DAgger has the lowest average falls, followed by SMILe(0.1), and then Supervised.](image-url)

[Ross+Gordon+Bagnell, AIStats11]
Super Mario Bros.

From Mario AI competition 2009

Input:

Output:

Jump in \( \{0,1\} \)
Right in \( \{0,1\} \)
Left in \( \{0,1\} \)
Speed in \( \{0,1\} \)

Extracted 27K+ binary features from last 4 observations
(14 binary features for every cell)
Training (expert)
Test-time execution (classifier)
Test-time execution (Dagger)
Average distance per stage

![Graph showing the average distance per stage as a function of the number of training data points. The graph compares DAgger, Searn(1), Searn(0.4), SMILe(0.1), and Supervised methods. The x-axis represents the number of training data points, while the y-axis represents the average distance per stage. The legend is included on the right side of the graph.]

[Ross+Gordon+Bagnell, AIStats11]
Perceptron vs. LaSO vs. Searn

- Incremental perceptron
- LaSO
- Searn / DAGger

Un-learnable decision
Discussion
Relationship between SP and IRL

- Formally, they're (nearly) the same problem
  - See humans performing some task
  - Define some loss function
  - Try to mimic the humans

- Difference is in philosophy:
  - (I)RL has little notion of beam search or dynamic programming
  - SP doesn't think about separating reward estimation from solving the prediction problem
  - (I)RL has to deal with stochasticity in MDPs
Important Concepts

- Search and loss-augmented search for margin-based methods
- Bold versus local updates for approximate search
- Training on-path versus off-path
- Stochastic versus deterministic worlds
- Q-states / values
- Learning reward functions vs. matching behavior
Language comprehension

But man is not destined to vanish. He can be killed, but he cannot be destroyed, because his soul is deathless and his spirit is irrepressible. Therefore, though the situation seems dark in the context of the confrontation between the superpowers, the silver lining is provided by amazing phenomenon that the very nations which have spent incalculable resources are desperately trying to destroy each other before the total holocaust.

The main point from the author's view is:

1. Man's soul and spirit can not be destroyed by superpowers.
2. Man's destiny is not fully clear or visible.
3. Man's soul and spirit are immortal.
4. Man's safety is assured by the delicate balance of power in terms of nuclear weapons.
5. Human society will survive despite the serious threat of total annihilation.
“Understanding” is being able to interact with the world based on language inputs
...but where do we get data?
Walkthroughs

You will appear on screen, and you will see that you have no weapon in hand whatsoever! Walk north to enter the open cave above you, and an Old Man will be inside. He addresses you with the following, insightful sentence: "It is dangerous to go alone! Take this." At this point you should step forward to grasp the Wooden Sword (used with the A Button), and then head back into the overworld.
Walkthroughs

You will appear on screen, and you will see that you have no weapon in hand whatsoever! Walk north to enter the open cave above you, and an Old Man will be inside. He addresses you with the following, insightful sentence: "It is dangerous to go alone! Take this." At this point you should step forward to grasp the Wooden Sword (used with the A Button), and then head back into the overworld.

Go to the small opening in the rock face on your right to arrive at H9, where you should kill the four Red Octoroks before going through the opening on the right. H10 has 5 Blue Tektites waiting for you, so be sure to kill them off before leaving the screen through the right as these fellows are an excellent source of Rupees, Fairies, and Hearts. H11 has four Blue Tektites, and after going right to ...
Why is this so hard?

You will appear on screen, and you will see that you have no weapon in hand whatsoever!

Walk north to enter the open cave above you, and an Old Man will be inside.

He addresses you with the following, insightful sentence: "It is dangerous to go alone! Take this."

At this point you should step forward to grasp the Wooden Sword (used with the A Button), and then head back into the overworld.
Walk north to enter the cave...

1111111001111111
11111213001111111
1113000001111111
11130000001111111
1300000004111111
0000000000000000
5600000000000011
1100000000000011
1100000000000011
111111111111111
111111111111111

U U U U L L U U U
L L U L L L U U U
L L L U L U U U
L L L L L U U U
L U L L L U U U
L U L L L U U U
U L L L L U U U

a=8  b=8  c=0  d=0
e=0  f=3  g=3  h=8
i=6  j=0  k=0

Hal Daumé III, me@hal3.name
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

*Instructing computers how to perform tasks:*

**UAV Learning task:**

*Identify suspicious vehicles*

**Programming**  **Function induction from data**
Learning: /lɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

*Instructing computers how to perform tasks:*

UAV Learning task:
*Identify suspicious vehicles*

Programming          Function induction from data
Learning: \(\text{/lɜrnɪŋ/}\)

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task: Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming                Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

**Instructing computers how to perform tasks:**

**UAV Learning task:**

*Identify suspicious vehicles*

Programming  Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming  Function induction from data
Learning: /lɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
   Identify suspicious vehicles

Programming            Function induction from data
Learning: /lɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming           Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task: Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

*Identify suspicious vehicles*

Programming Function induction from data
Learning: /ˈlɜːrnɪŋ/

“to gain knowledge by study, instruction, or experience”

**Instructing computers how to perform tasks:**

**UAV Learning task:**

*Identify suspicious vehicles*

---

**Programming**

**Function induction from data**
Learning: /ˈlɜrniŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming  Function induction from data
Learning: /ˈlɜrniŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming  Function induction from data
Learning: /lɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming  Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming  Function induction from data
Learning: /lərning/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming          Function induction from data
Learning: /lɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming               Function induction from data
Learning: /ˈlaːrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming 

Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

**Instructing computers how to perform tasks:**

**UAV Learning task:**

*Identify suspicious vehicles*

**Programming**

**Function induction from data**
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

**Instructing computers how to perform tasks:**

**UAV Learning task:**

*Identify suspicious vehicles*

Programming  Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming  Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

*Identify suspicious vehicles*

Programming Function induction from data
Learning: /ˈlɜrniŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

*Instructing computers how to perform tasks:*

**UAV Learning task:**

*Identify suspicious vehicles*

Programming  Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:
Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrnɪŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming Function induction from data
Learning: /ˈlɜrniŋ/

“to gain knowledge by study, instruction, or experience”

Instructing computers how to perform tasks:

UAV Learning task:

Identify suspicious vehicles

Programming  Function induction from data
Hal's Wager

- Give me a structured prediction problem where:
  - Annotations are at the lexical level
  - Humans can do the annotation with reasonable agreement
  - You give me a few thousand labeled sentences

- Then I can learn reasonably well...
  - ...using one of the algorithms we talked about

- Why do I say this?
  - Lots of positive experience
  - I'm an optimist
  - I want your counter-examples!
Open problems

- How to do SP when argmax is intractable....
  - Bad: simple algorithms diverge [Kulesza+Pereira, NIPS07]
  - Good: some work well [Finley+Joachims, ICML08]
  - And you can make it fast! [Meshi+al, ICML10]

- How to do SP with delayed feedback (credit assignment)
  - Kinda just works sometimes [D, ICML09; Chang+al, ICML10]
  - Generic RL also works [Branavan+al, ACL09; Liang+al, ACL09]

- What role does structure actually play?
  - Little: only constraints outputs [Punyakanok+al, IJCAI05]
  - Little: only introduces non-linearities [Liang+al, ICML08]

- Role of experts?
  - what if your expert isn't actually optimal?
  - what if you have more than one expert?
  - what if you only have trajectories, not the expert?
Things I have no idea how to solve...

```
all : (a → Bool) → [a] → Bool

Applied to a predicate and a list, returns `True' if all elements of the list satisfy the predicate, and `False' otherwise.
```

```
%module main:MyPrelude
%data main:MyPrelude.MyList aadj =
  {main:MyPrelude.Nil;
   main:MyPrelude.Cons aadj ((main:MyPrelude.MyList aadj))};
%rec
{main:MyPrelude.myzuall :: %forall tadA . (tadA ->
  ghczmprim:GHCziBool.Bool)
  ->
  (main:MyPrelude.MyList tadA) ->
  ghczmprim:GHCziBool.Bool =

  \ @ tadA
  (padk::tadA -> ghczmprim:GHCziBool.Bool)
  (dsddE::(main:MyPrelude.MyList tadA)) ->
  %case ghczmprim:GHCziBool.Bool dsddE
  %of (wildBl::(main:MyPrelude.MyList tadA))
  {main:MyPrelude.Nil ->
    ghczmprim:GHCziBool.True;
    main:MyPrelude.Cons
    (xadm::tadA) (xsadn::(main:MyPrelude.MyList tadA)) ->
    %case ghczmprim:GHCziBool.Bool (padk xadm)
    %of (wild1Xc::ghczmprim:GHCziBool.Bool)
    {ghczmprim:GHCziBool.False ->
      ghczmprim:GHCziBool.False;
      ghczmprim:GHCziBool.True ->
      main:MyPrelude.myzuall @ tadA padk xsadn})
)
```

(s1) A father had a family of sons who were perpetually quarrelling among themselves. (s2) When he failed to heal their disputes by his exhortations, he determined to give them a practical illustration of the evils of disunion; and for this purpose he one day told them to bring him a bundle of sticks. (s3) When they had done so, he placed the faggot into the hands of them to break it. (s4) They met this command with strength, and we the faggot, took it again put them in. (s5) again put them in. (s6) "My sons, if you try to break sticks together, you will be broken by each other, you will be broken by each other, you will be broken by each other.

\[
\begin{align*}
\text{Father} & \quad (\text{annoyed})_{a2} \quad \ldots \quad \text{shared} \quad \ldots \quad \text{Sons} \\
\text{m} & \quad \downarrow_{s2} \\
\text{m} & \quad \downarrow_{s2} \\
\text{a} & \quad \downarrow_{s2} \\
\text{m} & \quad \downarrow_{s2} \\
\text{m} & \quad \downarrow_{s2} \\
\end{align*}
\]

\[
\begin{align*}
\text{m} & \quad \downarrow_{s2} \\
\text{m} & \quad \downarrow_{s2} \\
\text{a} & \quad \downarrow_{s2} \\
\text{m} & \quad \downarrow_{s2} \\
\text{m} & \quad \downarrow_{s2} \\
\end{align*}
\]
Software

- Sequence labeling
  - Mallet  http://mallet.cs.umass.edu
  - CRF++  http://crfpp.sourceforge.net

- Search-based structured prediction
  - LaSO  http://hal3.name/TagChunk
  - Searn  http://hal3.name/searn

- Higher-level “feature template” approaches
  - Alchemy  http://alchemy.cs.washington.edu
  - Factorie  http://code.google.com/p/factorie
Summary

- Structured prediction is easy if you can do argmax search (esp. loss-augmented!)
- Label-bias can kill you, so iterate (Searn/Dagger)
- Stochastic worlds modeled by MDPs
- IRL is all about learning reward functions
- IRL has fewer assumptions
  - More general
  - Less likely to work on easy problems
- We're a long way from a complete solution
- Hal's wager: we can learn pretty much anything

Thanks! Questions?
Conclusion

Thanks! Questions?
References

See also:

http://www.cs.utah.edu/~suresh mediawiki/index.php/MLRG
http://braque.cc/ShowChannel?handle=P5BVAC34
Stuff we talked about explicitly

Other good stuff

- **Training structural SVMs when exact inference is intractable.** T. Finley and T. Joachims. ICML, 2008.
- **Structure compilation: trading structure for features.** P. Liang, H. Daume, D. Klein. ICML 2008.
- **Learning semantic correspondences with less supervision.** P. Liang, M. Jordan and D. Klein. ACL, 2009.
- **Maximum entropy Markov models for information extraction and segmentation.** A. McCallum, D. Freitag, F. Pereira. ICML 2000.
- **Learning and inference over constrained output.** V. Punyakanok, D. Roth, W. Yih, D. Zimak. IJCAI, 2005.