Algorithms that learn to think on their feet
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 (more than just NLP?)
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
  - Scientists study whales from space

- Why are these funny?
- What does ambiguity imply about the role of learning?
Learning to be *fast!*

Quality = tradeoff(accuracy, time)
Feature-Frugal Dependency Parsing

NLP algorithms use a kitchen sink of features

Simultaneous Machine Interpretation

Quizbowl (Incremental Question Answering)

Outline
Dependency parsing

NLP algorithms use a kitchen sink of features
Case study: dependency parsing

The graph shows the time (in seconds) for different sentence lengths on a logarithmic scale. The y-axis represents the time in seconds, ranging from 0 to 250 with intervals of 50. The x-axis represents sentence length, ranging from 0 to 250 with intervals of 25. The graph includes three categories: Tagging, Features, and Parsing.

- **Average Sentence:**
  - Tagging: 3.73 seconds
  - Features: 32.80 seconds
  - Parsing: 0.12 seconds

The graph indicates that for shorter sentences, the time taken for parsing is significantly lower than for tagging and features. As sentence length increases, the time for all categories also increases, with parsing remaining the fastest across the range of sentence lengths.
Dynamic feature selection

Example

Classifier

h

Remaining Fts

FunnyIP
MailFrom
MailFromWeird
BlacklistedIP
SubjectLength
SubjectWords
BodySize
HasAttachment
...
BodyWords

(He+Eisner+D, NIPS 2012)
Dynamic feature selection

Example

| FunnyIP | 1 |

Classifier

- . 8

Remaining Fts

- MailFrom
- MailFromWeird
- BlacklistedIP
- SubjectLength
- SubjectWords
- BodySize
- HasAttachment
- ...
- BodyWords

Example

| h |
Dynamic feature selection

Example

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunnyIP</td>
<td>1</td>
</tr>
<tr>
<td>SubjectLength</td>
<td>26</td>
</tr>
</tbody>
</table>

Classifier

Remaining Fts

- MailFrom
- MailFromWeird
- BlacklistedIP
- SubjectWords
- BodySize
- HasAttachment
- ...
- BodyWords
Dynamic feature selection

Example

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunnyIP</td>
<td>1</td>
</tr>
<tr>
<td>SubjectLength</td>
<td>26</td>
</tr>
<tr>
<td>HasAttachment</td>
<td>-1</td>
</tr>
</tbody>
</table>

Remaining Fts

- MailFrom
- MailFromWeird
- BlacklistedIP
- SubjectWords
- BodySize
- BodyWords

Classifier

\[ h(\text{Example}) + 4 \]
Dynamic feature selection

Example
- FunnyIP: 1
- SubjectLength: 26
- HasAttachment: -1

Classifier

Remaining Fts
- MailFrom
- MailFromWeird
- BlacklistedIP
- SubjectWords
- BodySize
- BodyWords
- ...
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 $\bigcup_{i<n} D_i$

If $N = T \log T$, $L(\pi_n) < T \epsilon_N + O(1)$ for some $n$
The oracle too good!

- The oracle *knows the label*
- Picks feature with highest $y^*$ value
- Ends after selecting one feature
- Coach says how to improve, not the best thing to do

Pssst! You should choose $\text{argmin} \ E[l(a)]$

If $N = T \log T$, $L(\pi_n) < T \epsilon_N$ for some $n$

Provably smaller than oracle's epsilon!

90%
80%
70%

Overall accuracy

Oracle

Average cost/example

0.2
0.4

Coaching

Searn

Forward selection
Dependency parsing

NLP algorithms use a kitchen sink of features
Dependency parsing

Algorithms that think on their feet

Hal Daumé III (me@hal3.name)
Dependency parsing

Three steps:
1. Compute POS tags
2. Compute $k n^2$ features
3. Run directed MST

Edge Features:
- Lex(verb→noun)
- POS(verb→noun)
- skip(2)
- skip(Det)
- skip(Noun)
- skip(Det Noun)
- dist=3
- various regexps...
- spelling features
- etc...

NL−P
The system we learn to control

+ first feature group

5 features per gray edge
51 gray edge with unknown fate...

This time, the firms were ready.

Undetermined edge
Current 1-best tree
Winner edge
Loser edge
The system we learn to control

5 features per gray edge
51 gray edge with unknown fate...

$ This time, the firms were ready

Undetermined edge
Current 1-best tree
Winner edge
Loser edge

Non-projective decoding

(He+Eisner+D, EMNLP 2013)
The system we learn to control

5 features per gray edge
50 gray edge with unknown fate...

$ This time, the firms were ready

Undetermined edge
Current 1-best tree
Winner edge
Loser edge

Decide winners among the blue edges

(He+Eisner+D, EMNLP 2013)
The system we learn to control

5 features per gray edge
44 gray edge with unknown fate...

$ This time, the firms were ready

Remove losers in conflict with the winners

Undetermined edge
Current 1-best tree
Winner edge
Loser edge

(He+Eisner+D, EMNLP 2013)
The system we learn to control

5 features per gray edge
44 gray edge with unknown fate...

Remove losers in conflict with the winners

Undetermined edge
Current 1-best tree
Winner edge
Loser edge

$ This time, the firms were ready
The system we learn to control

$+$ next feature group

27 features per gray edge

44 gray edge with unknown fate...

$\text{This time, the firms were ready}$

Undetermined edge

Current 1-best tree

Winner edge

Loser edge

(He+Eisner+D, EMNLP 2013)
The system we learn to control
Static versus dynamic feature selection

![Graph showing comparison between static and dynamic feature selection. The x-axis represents runtime (s), and the y-axis represents unlabeled attachment score (UAS). The static feature selection line starts at a lower score and reaches a higher score over time compared to the dynamic feature selection line, which starts at a higher score and reaches a lower score over time.](https://example.com/figure.png)
Looking inside the box...

- **Later features are helpful**
  - Most edges win or lose early
  - Linear increase in runtime

The graph shows the relationship between feature selection stage and time/accuracy/edge percentage. The blue line represents 'runtime %', the green line represents 'UAS %', the black line represents 'remaining edge %', and the turquoise line represents 'winner edge %'. The stages of feature selection are labeled from 0 to 6 on the x-axis, and the percentage values are shown on the y-axis.
Accuracy is (essentially) unaffected

![Bar chart showing accuracy for Bulgarian, Chinese, English, German, Japanese, Portuguese, and Swedish languages. The chart compares DynFS, VineP, and Baseline models.](image-url)
...but we get a lot faster
Algorithms that think on their feet

Hal Daum

e III (me@hal3.name)

NLP algorithms use a kitchen sink of features

Feature-Frugal Dependency Parsing

Simultaneous Machine Interpretation

Quizbowl (Incremental Question Answering)

Outline
Simultaneous (machine) interpretation

➢ Dozens of defendants
➢ Judges from four nations (three languages)
➢ Status quo: speak, then translate
➢ After Nuremberg, simultaneous translations became the norm
➢ Long wait → bad conversation
Why simultaneous interpretation is hard

- Human languages have vastly different word orders
  - About half are OV, the other half are VO
  - This comes with a lot more baggage than just verb-final

Running (German/English) Example:

Ich bin mit dem Zug nach Ulm gefahren
I am with the train to Ulm traveled
I (...... waiting......) traveled by train to Ulm
We have a set of actions (predict / translate)

- Wait
- Predict clause-verb
- Predict next word
- Commit ("speak")

In a changing environment (state)

- The words we've seen so far
- Our models' internal predictions

With a well defined oracle
Example of interpretation trajectory

Ich bin mit dem Zug nach Ulm gefahren
I am with the train to Ulm traveled
I (…… waiting ……) traveled by train to Ulm
Evaluating performance and baselines
Evaluating performance and baselines
Evaluating performance and baselines

Source Sentence: Er ist

Psychic: He went to the store

Monotone: He

Batch

Policy Prediction: He

Good Translation

Bad Translation

Good Translation

Bad Translation

Good Translation

Bad Translation

Good Translation

Bad Translation
Evaluating performance and baselines
Evaluating performance and baselines
Evaluating performance and baselines
Evaluating performance and baselines
Evaluating performance and baselines

% of Sentence

Smoothed Average

Batch, Monotone, Optimal, Learned
Feature-Frugal Dependency Parsing

NLP algorithms use a kitchen sink of features

Simultaneous Machine Interpretation

Outline

Quizbowl (Incremental Question Answering)
Humans doing incremental prediction

- Game called “quiz bowl”
- Two teams play each other
  - Moderator reads a question
  - When a team knows the answer, they buzz in
  - If right, they get points; otherwise, rest of the question is read to the other team
- Hundreds of teams in the US alone
- Example . . .
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory.
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so
With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients for spontaneous and stimulated emission, the subject of one of his multiple groundbreaking 1905 papers. He further developed the model of statistics sent to him by
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients for spontaneous and stimulated emission, the subject of one of his multiple groundbreaking 1905 papers. He further developed the model of statistics sent to him by Bose to describe particles with integer spin. For 10 points, who is this German physicist best known for formulating
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients for spontaneous and stimulated emission, the subject of one of his multiple groundbreaking 1905 papers. He further developed the model of statistics sent to him by Bose to describe particles with integer spin. For 10 points, who is this German physicist best known for formulating the special and general theories of relativity?
Solving incrementally

- Action: buzz now or wait
  - Content Model is constantly generating guesses
  - Oracle provides examples where it is correct
  - The Policy generalizes to test data
  - Features represent our state

---

Qatar

From Wikipedia, the free encyclopedia

For other places with the same name, see Qatar (disambiguation).

Qatar ( winger, /kɑːˈtɑːr/ or /kɑːˈtɛr/; Arabic: قطر [qatˈʔar]; local the State of Qatar (Arabic: دولة قطر دولة قطر), is a sovereign Arab the small Qatar Peninsula on the northeastern coast of the Arabian Penin to the south, with the rest of its territory surrounded by the Persian Gulf. from the nearby island kingdom of Bahrain. In 2013, Qatar's total populat and 1.5 million expatriates.

---

- arabian
- persian
- gulf
- kingdom
- expatriates

---

- Gummibears
- Pokemon
- Qatar
- Bahrain

?
Evaluation methodology

- Mechanical Turk to collect human data
- 7000 questions were answered in the first day
- Over 43000 questions were answered in the space of two weeks
- Total of 461 unique users
- Leaderboard to encourage users

**Big problem:**

“this man shot at Aaron Burr”  
*is very different from*  
“Aaron Burr shot at this man”
Challenge: modeling compositionality

invented

he

fridge

a
double-e

with

parts

no

moving

GummiBears

Pokemon

Qatar

Bahrain

?
Challenge: modeling compositionality

He invented a fridge no moving parts.
Challenge: modeling compositionality

Algorithms that think on their feet

Hal Daumé III (me@hal3.name)
Challenge: modeling compositionality

\[ e(\text{fridge}) = f( w_{\text{fridge}} + \forall \left[ e( a) \cdot e(\text{double-e}) \cdot e(\text{with}) \right] ) \]
Results on question-answering task

History: Model vs. Human

Score Difference

-200 -150 -100 -50 0 50 100 150 200

Model loses
Model wins

Full

BOW (QB)
RNN (QB)
BOW (wiki)
Combined
Results on question-answering task

Literature: Model vs. Human

Score Difference

-400 -300 -200 -100 0 100 200

Model loses

Model wins

Full

25 35 45 55 65 75 85

BOW (QB)

RNN (QB)

BOW (wiki)

Combined
Moving to more general frameworks

- Lots of NLP (+al) problems can be cast at test time as integer linear programs
- ILPs are usually solved using branch and bound

Branch and bound involves a complex heuristic search
Can we learn to perform this search efficiently?

Min: \(-2x - y\)
subject to:
1. \(3x - 5y \leq 0\)
2. \(3x + 5y \leq 15\)
3. \(x \geq 0, y \geq 0\)
4. \(x, y \in \mathbb{Z}\)
Some intuition

- A good search strategy should:
  - find a good incumbent solution early
  - identify non-promising nodes before expansion

- “Good” varies depending on your position in the tree:
  - DFS should only be used at nodes that promise to lead to a good feasible solution that may replace the incumbent
  - Best-bound-first search can quickly discard unpromising nodes, but should not be used frequently at the top

- We will learn a heuristic that can capture this intuition
We achieve less than 1.2 optimality gap while exploring 0.05%, 1.5%, 5.1% and 47% of the nodes explored by Gurobi!

**Measures:**
- Optimality Gap, Integrality Gap, and improvement from initial heuristic solution

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ours (DAgger training)</th>
<th>DFS</th>
<th>Gurobi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OGap</td>
<td>IGap</td>
<td>Impr</td>
</tr>
<tr>
<td>MIK</td>
<td>0.23</td>
<td>16.63</td>
<td>4.39</td>
</tr>
<tr>
<td>Regions1</td>
<td>0.54</td>
<td>4.53</td>
<td>10.57</td>
</tr>
<tr>
<td>Regions2</td>
<td>1.22</td>
<td>6.76</td>
<td>19.36</td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.87</td>
<td>20.28</td>
<td>24.46</td>
</tr>
</tbody>
</table>
Some other fun stuff...

```python
class SequenceLabeler(pyvw.SearchTask):
    def __init__(self, vw, srn):
        # you must must must initialize
        # this will automatically
        pyvw.SearchTask.__init__(self, vw)
        srn.set_options(srn.A)

    def _run(self, sentence):
        output = []
        for tag, word in sentence:
            pred = self.srn.
            output.append(pred)
        return output

for curPass in range(10):
    sequenceLabeler.learn(my_data)
```

Named entity recognition (tuned hps)

- F-score (per entity)
- Training Time (minutes)

- 73.3
- 79.2
- 80.0
- 76.5
- 78.3
- 74.6
- 75.9
- 78.3
• Reasoning with incomplete information is useful for speed and modeling

• *Imitation learning* can help us build such systems

• Wide range of new, interesting problems to work on!

Thanks! Questions?