The STRANGE WORLD of the PRESENT

Cognitive Computing

Chris Welty
Brief Personal History

Google Docs Explore

#originalwatsonteam
(Watson on Jeopardy!)

Natural Language Processing

Semantic Web

Ontology
I amar prestar aen
The world has changed
CogComp: *It's OK to be wrong!*

- Cognitive Systems can fail and still be useful
- A dramatic shift for software systems
- Focus is on improvement, not perfection
- Must be able to answer, “Is it better?”
The 9 pillars of cognitive systems engineering

<table>
<thead>
<tr>
<th>Cognitive</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Measure, measure, measure

- Devise (& believe) a metric
  - e.g. % correct at 70% answered, or % correct at .5 confidence, or # correct at .8 confidence
- Establish baseline performance
  - new development is only added when it improves performance
- If your cognitive system reaches above .8 then probably your metric, data, or truth is inadequate
Data, data, data

- Welcome to the data age
- Cognitive system engineers can’t have enough
- Customers must be prepared to give up their data
- Elevate data development over app development
  - Data is the enterprise
  - Software is one-off
Truth, truth, truth

- Data and metrics are not enough
- There has to be a reference standard, often called *ground truth*, to measure against
  - eg QA: questions with the right answer
  - eg search: search queries and correct pages
  - eg parsing: sentences with correct parse trees
- What happens when problems don’t have a strict right/wrong outcome? see *CrowdTruth* (w/ Dr. Lora Aroyo)
- This is not about supervised learning, this is about measurement & evaluation
The tyranny of “Truth”

Does each sentence express the TREAT relation?

**ANTIBIOTICS** are the first line treatment for indications of **TYPHUS**.

Patients with **TYPHUS** who were given **ANTIBIOTICS** exhibited several side-effects.

With **ANTIBIOTICS** in short supply, DDT was used during World War II to control the insect vectors of **TYPHUS**.
The tyranny of “Truth”

Does each sentence express the TREAT relation?

**ANTIBIOTICS** are the first line treatment for indications of **TYPHUS**. **High Clarity**

Patients with **TYPHUS** who were given **ANTIBIOTICS** exhibited several side-effects. **Medium Clarity**

With **ANTIBIOTICS** in short supply, DDT was used during World War II to control the insect vectors of **TYPHUS**. **Low Clarity**
The tyranny of “Truth”

Does each sentence express the TREAT relation?

**ANTIBIOTICS** are the first line treatment for indications of **TYPHUS**. **99% Agreement**

Patients with **TYPHUS** who were given **ANTIBIOTICS** exhibited several side-effects. **80% Agreement**

With **ANTIBIOTICS** in short supply, DDT was used during World War II to control the insect vectors of **TYPHUS**. **60% Agreement**
The 9 pillars of cognitive systems engineering

<table>
<thead>
<tr>
<th>Cognitive</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Truth</td>
</tr>
<tr>
<td>Truth</td>
<td>Measure</td>
</tr>
<tr>
<td>Measure</td>
<td></td>
</tr>
</tbody>
</table>

The 9 pillars define the cognitive system. If you change one of them, the system should change.
Hypothesis: given some text a user has written, we can predict what people will write about next if we *read the web*

Idea: Google’s “knowledge graph” knows what entities are on the web.
Hypothesis: given some text a user has written, we can predict what they will write about next if we read the web.

Idea: Google’s “knowledge graph” knows what entities are on the web.
Hypothesis: given some text a user has written, we can predict what people will write about next if we *read the web*.

Idea: Google's "knowledge graph" knows what entities are on the web.
Hypothesis: given some text a user has written, we can predict what people will write about next if we *read the web*

Idea: Google’s “knowledge graph” knows what entities are on the web
Hypothesis: given some text a user has written, we can predict what people will write about next if we *read the web*

Idea: Google’s “knowledge graph” knows what entities are on the web
Hypothesis: given some text a user has written, we can predict what people will write about next if we **read the web**

Idea: Google’s “knowledge graph” knows what entities are on the web
Hypothesis: given some text a user has written, we can predict what people will write about next if we *read the web*

Idea: Google’s “knowledge graph” knows what entities are on the web.
Hypothesis: given some text a user has written, we can predict what people will write about next if we read the web

Idea: Google’s “knowledge graph” knows what entities are on the web

If two entities co-occur and a user writes about one, predict they will mention the other, as long as they haven’t already done that.
Hypothesis: given some text a user has written, we can predict what people will write about next if we *read the web*

Idea: Google’s “knowledge graph” knows what entities are on the web

If two entities co-occur and a user writes about one, predict they will mention the other, *as long as they haven’t already done that*.

Simple.
Hypothesis: given some text a user has written, we can predict what people will write about next if we read the web

Idea: Google’s “knowledge graph” knows what entities are on the web

If two entities co-occur and a user writes about one, predict they will mention the other, as long as they haven’t already done that.

Simple.

Make it work for 240+ million users, on the web, in all languages, get them to use it, and demonstrate that it has value.
Cognitive Systems Engineering
A 12-step program
Explore, explore, explore

"Equipped with his five senses, man explores the universe around him and calls the adventure science."

Edwin Hubble

- Traditional software development cycle is Design, Implement, Test, Modify (Refine, Repair, Enhance)
- Cognitive systems software development is data science: exploring ideas, connections, discovering "signals" in the data
Fail, fail, fail

“La science, mon garçon, est faite d’erreurs, mais d’erreurs qu’il est bon de commettre, car elles mènent peu à peu à la vérité.”

Jules Verne, Journey to the Center of the Earth

- You should expect 60-80% of cognitive systems development effort (exploration) to have no impact
  ○ In my personal experience, 66% of my work was not included in the system
  ○ that’s because I’m good
- This is a necessary part of exploration
- *Invest in failure to succeed*
"Each demon may be assigned, for example, a letter of the alphabet, and it his job to shout as loud as the amount of A-ness he sees..."

Oliver Selfridge, Pandemonium

- Cognitive systems combine multiple approximate methods
- Jeopardy! Watson had more than 100 different software components
  - No single component does the whole job
  - Many of them do very similar jobs
  - 12 typing components, 8 passage alignment components, 10 n-gram components, ...
- These methods are not integrated with each other beyond that they each produce a score for each hypothesis
- A algorithm learns how to combine them to produce a final score
Follow, Follow, Follow

“Try to remember and if you remember then follow, follow, follow…”  
Tom Jones, The Fantastiks

- Cognitive systems are never finished
- They evolve & improve over time
- Establish baseline, run follow-ons
  - A follow-on experiment adds some new component to the system
  - Performance is compared to the baseline
- Only components that improve over the baseline are included in the system
- There will eventually be a levelling-off in improvement
Measure, measure, measure

- Figuring out *what to measure* is often the most difficult part of building any cognitive system.
- Deep Learning success stories have mostly been due to:
  - Massive amounts of data
  - With built-in associations
  - E.g. massive crowd face tagging, massive opinion mining, click-through rates, etc.
- ...what do you do when the correct associations are not in big data?
Metrics and measurements

- A good metric must be
  - Repeatable, reproducible
  - Aligned with human performance
  - Automatic
- To align with human performance
  - Gather some data from people
    - Hire annotators, use crowdsourcing
  - As much as you can afford
  - Tweak the metric until it correlates
- Gather data in stages as your understanding of the problem increases
  - Avoid the one-off, prefer incremental gathering
PUT PRIVACY FIRST

- A good metric must
  - Respect people’s privacy expectations
Don’t make it too easy

- Good metrics should be hard to “beat”
  - E.g. Bleu
Cognitive systems can be wrong and be useful!

The world has changed

<table>
<thead>
<tr>
<th>Cognitive</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Truth</td>
<td>Truth</td>
</tr>
<tr>
<td>Measure</td>
<td>Measure</td>
</tr>
</tbody>
</table>

Explore | Fail | Combine | Follow | 12 STEPS |