Lecture 18:
The Limits and Future of NLP

Richard Socher
Poster Session

• Everyone expected to attend (or video).
• 5:30pm-8:30pm next Wednesday.
• Check out other student’s posters.
• Dinner on us.
• Stay until end if you think you have a chance of winning one of the awards
• Jobs and funding
• Fun :)}
What has been lost from old NLP work?

• An earlier era of work had lofty goals, but modest realities
• Today, we have much better realities, but often content ourselves with running LSTMs rather than reaching for the stars
Norvig (1986) Ph.D.

Peter Norvig’s thesis – 30th anniversary

A Unified Theory of Inference for Text Understanding

By

Peter Norvig
B.S. (Brown University) 1978

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

in the

GRADUATE DIVISION

OF THE

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved: " "
Chairman
D. A. Zadeh
11/25/86

Date

Chuck Fillmore
11/25/86
The language analyzed

• In a poor fishing village built on an island not far from the coast of China, a young boy named Chang Lee lived with his widowed mother. Every day, little Chang bravely set off with his net, hoping to catch a few fish from the sea, which they could sell and have a little money to buy bread.

(a) There is a sea, which surrounds the island, is used by the villagers for fishing, and forms part of the coast of China
(b) Chang intends to trap fish in his net, which is a fishing net
(c) The word which refers to the fish
(d) The word they refers to Chang and his mother
Basic NLP: Progress has been made!

“Arens and Wilensky’s PHRAN program was used where possible [to convert input sentences to KODIAK knowledge representations]. For some input, PHRAN was not up to the task, so a representation was constructed by hand instead.” (p. 4)
Building elaborations a la Norvig (1986)
What do we still need?

• BiLSTMs with attention seem to be taking over the field and improving our ability to do everything

• Neural methods are leading to a renaissance for all language generation tasks (i.e., MT, dialog, QA, summarization, ...)

• There’s a real scientific question of where and whether we need explicit, localist language and knowledge representations and inferential mechanisms
What do we still need?

• However: We still have very primitive methods for building and accessing memories or knowledge

• Current models have almost nothing for developing and executing goals and plans*
Progress on goals and plans

- Hierarchical and Interpretable Skill Acquisition in Multi-task Reinforcement Learning, Tianmin Shu, Caiming Xiong, and Richard Socher
  International Conference on Learning Representations (ICLR 2018)
What do we still need?

• We still have quite inadequate abilities for understanding and using inter-sentential relationships.

• We still can’t, at a large scale, do elaborations from a situation using common sense knowledge BUT also have bias
The Limits of Single Task Learning

- Great performance improvements
- Projects start from random
- Single unsupervised task can’t fix it
- We will never get to a truly general NLP model this way.
Towards NLP-Complete Super Tasks

• How to express different tasks in the same framework, e.g.

  – Sequence tagging: aspect specific sentiment

  – Text classification: dialogue intent classification

  – Seq2seq: machine translation, summarization, etc.
The 3 Equivalent NLP-Complete Super Tasks

- Language modeling
- Question answering
- Dialogue systems

Usefulness and complexity in their current interpretation
Framework for Tackling NLP

A joint model for comprehensive QA
QA Examples

I: Mary walked to the bathroom.
I: Sandra went to the garden.
I: Daniel went back to the garden.
I: Sandra took the milk there.
Q: Where is the milk?
A: garden
I: Everybody is happy.
Q: What’s the sentiment?
A: positive

I: I think this model is incredible
Q: In French?
A: Je pense que ce modèle est incroyable.

I:

Q: What color are the bananas?
A: Green.

Move from \( \{x_i, y_i\} \) to \( \{x_i, q_i, y_i\} \)
First of Several Major Obstacles

- For NLP no single model *architecture* with consistent state of the art results across tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>State of the art model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question answering (babI)</td>
<td>Strongly Supervised MemNN (Weston et al 2015)</td>
</tr>
<tr>
<td>Sentiment Analysis (SST)</td>
<td>Tree-LSTMs (Tai et al. 2015)</td>
</tr>
<tr>
<td>Part of speech tagging (PTB-WSJ)</td>
<td>Bi-directional LSTM-CRF (Huang et al. 2015)</td>
</tr>
</tbody>
</table>
Tackling Obstacle: Dynamic Memory Network

But, now known, it’s not enough
Obstacle: Joint Many-task Learning

- Fully joint multitask learning* is hard:
  - Usually restricted to lower layers
  - Usually helps only if tasks are related
  - Often hurts performance if tasks are not related
  - We lose powerful accuracy improvement techniques such as task-specific architecture and hyperparameter tuning

* meaning: same decoder/classifier and not only transfer learning with source target task pairs, no swappable modeling blocks per task
Tackling Joint Training

- A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks
  Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka & Richard Socher

- Final Model →
Model Details

• Include character n-grams and short-circuits
• State of the art purely feedforward parser

**POS Tagging:**

![Diagram of POS Tagging]

**Chunking:**

![Diagram of Chunking]

\[
y_t^{(pos)} = \sum_{j=1}^{C} p(y_t^{(1)} = j|h_t^{(1)})\ell(j),
\]
Dependency Parsing

Figure 3: Overview of dependency parsing in the third layer of the JMT model.

Figure 4: Overview of the semantic tasks in the top layers of the JMT model.

Define $h^{(3)}_{L+1} = r$ as a parameterized vector. To compute the probability that $w_j$ (or the root node) is the parent of $w_t$, the scores are normalized:

$$p_j(h^{(3)}_t) = \frac{\exp (m(t,j))}{\sum_{k=1}^{L} \exp (m(t,k))}$$

where $L$ is the sentence length.

Next, the dependency labels are predicted using $[h^{(3)}_t; h^{(3)}_j]$ as input to a standard softmax classifier with a single ReLU layer. At test time, we greedily select the parent node and the dependency label for each word in the sentence.

At training time, we use the gold child-parent pairs to train the label predictor.

2.5 Semantic Tasks: Semantic Relatedness

The next two tasks model the semantic relationships between two input sentences. The first task measures the semantic relatedness between two sentences. The output is a real-valued relatedness score for the input sentence pair. The second task is a textual entailment task, which requires one to determine whether a premise sentence entails a hypothesis sentence. There are typically three classes: entailment, contradiction, and neutral.

The two semantic tasks are closely related to each other. If the semantic relatedness between two sentences is very low, they are unlikely to entail each other. Based on this intuition and to make use of the information from lower layers, we use the fourth and fifth bi-LSTM layer for the relatedness and entailment task, respectively.

This method currently assumes that each word has only one parent node, but it can be expanded to handle multiple parent nodes, which leads to cyclic graphs.
Multi Sentence Tasks: Semantic Relatedness

The next two tasks model the semantic relationships between two input sentences. The first task measures the semantic relatedness between two sentences. The output is a real-valued relatedness score for the input sentence pair. The second task is a textual entailment task, which requires one to determine whether a premise sentence entails a hypothesis sentence. There are typically three classes: entailment, contradiction, and neutral.

The two semantic tasks are closely related to each other. If the semantic relatedness between two sentences is very low, they are unlikely to entail each other. Based on this intuition and to make use of the information from lower layers, we use the fourth and fifth bi-LSTM layer for the relatedness and entailment task, respectively.

This method currently assumes that each word has only one parent node, but it can be expanded to handle multiple parent nodes, which leads to cyclic graphs.
Training Details: Regularized Idea

Chunking training

\[- \sum_s \sum_t \log p(y_t^{(2)}) = \alpha |h_t^{(2)}) + \lambda \|W_{\text{chunk}}\|^2 + \delta \|\theta_{\text{POS}} - \theta'_{\text{POS}}\|^2,\]

Entailment training

\[- \sum_{(s,s')} \log p(y_{(s,s')}^{(5)}) = \alpha |\hat{h}_s^{(5)}, \hat{h}_{s'}^{(5)}) + \lambda \|W_{\text{ent}}\|^2 + \delta \|\theta_{\text{rel}} - \theta'_{\text{rel}}\|^2,\]
Joint Training Helps Here!

<table>
<thead>
<tr>
<th></th>
<th>Single</th>
<th>JMT\textsubscript{all}</th>
<th>JMT\textsubscript{AB}</th>
<th>JMT\textsubscript{ABC}</th>
<th>JMT\textsubscript{DE}</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>POS</td>
<td>97.45</td>
<td>97.55</td>
<td>97.52</td>
<td>97.54</td>
</tr>
<tr>
<td>B</td>
<td>Chunking</td>
<td>95.02</td>
<td>(97.12)</td>
<td>95.77</td>
<td>(97.28)</td>
</tr>
<tr>
<td>C</td>
<td>Dependency UAS</td>
<td>93.35</td>
<td>94.67</td>
<td>n/a</td>
<td>94.71</td>
</tr>
<tr>
<td></td>
<td>Dependency LAS</td>
<td>91.42</td>
<td>92.90</td>
<td>n/a</td>
<td>92.92</td>
</tr>
<tr>
<td>D</td>
<td>Relatedness</td>
<td>0.247</td>
<td>0.233</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>E</td>
<td>Entailment</td>
<td>81.8</td>
<td>86.2</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
New State of the Art on 4 of 5 Tasks

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMT_all</td>
<td>97.55</td>
</tr>
<tr>
<td>Ling et al. (2015)</td>
<td>97.78</td>
</tr>
<tr>
<td>Kumar et al. (2016)</td>
<td>97.56</td>
</tr>
<tr>
<td>Ma &amp; Hovy (2016)</td>
<td>97.55</td>
</tr>
<tr>
<td>Søgaard (2011)</td>
<td>97.50</td>
</tr>
<tr>
<td>Collobert et al. (2011)</td>
<td>97.29</td>
</tr>
<tr>
<td>Tsuruoka et al. (2011)</td>
<td>97.28</td>
</tr>
<tr>
<td>Toutanova et al. (2003)</td>
<td>97.27</td>
</tr>
</tbody>
</table>

Table 2: POS tagging results.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMT_all</td>
<td>0.233</td>
</tr>
<tr>
<td>JMT_all</td>
<td>0.238</td>
</tr>
<tr>
<td>Zhou et al. (2016)</td>
<td>0.243</td>
</tr>
<tr>
<td>Tai et al. (2015)</td>
<td>0.253</td>
</tr>
</tbody>
</table>

Table 5: Semantic relatedness results.

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMT_all</td>
<td>95.77</td>
</tr>
<tr>
<td>Søgaard &amp; Goldberg (2016)</td>
<td>95.56</td>
</tr>
<tr>
<td>Suzuki &amp; Isozaki (2008)</td>
<td>95.15</td>
</tr>
<tr>
<td>Collobert et al. (2011)</td>
<td>94.32</td>
</tr>
<tr>
<td>Kudo &amp; Matsumoto (2001)</td>
<td>93.91</td>
</tr>
<tr>
<td>Tsuruoka et al. (2011)</td>
<td>93.81</td>
</tr>
</tbody>
</table>

Table 3: Chunking results.

<table>
<thead>
<tr>
<th>Method</th>
<th>UAS</th>
<th>LAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMT_all</td>
<td>94.67</td>
<td>92.90</td>
</tr>
<tr>
<td>Single</td>
<td>93.35</td>
<td>91.42</td>
</tr>
<tr>
<td>Andor et al. (2016)</td>
<td>94.61</td>
<td>92.79</td>
</tr>
<tr>
<td>Alberti et al. (2015)</td>
<td>94.23</td>
<td>92.36</td>
</tr>
<tr>
<td>Weiss et al. (2015)</td>
<td>93.99</td>
<td>92.05</td>
</tr>
<tr>
<td>Dyer et al. (2015)</td>
<td>93.10</td>
<td>90.90</td>
</tr>
<tr>
<td>Bohnet (2010)</td>
<td>92.88</td>
<td>90.71</td>
</tr>
</tbody>
</table>

Table 4: Dependency results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMT_all</td>
<td>86.2</td>
</tr>
<tr>
<td>JMT_all</td>
<td>86.8</td>
</tr>
<tr>
<td>Yin et al. (2016)</td>
<td>86.2</td>
</tr>
<tr>
<td>Tai et al. (2015)</td>
<td>84.6</td>
</tr>
</tbody>
</table>

Table 6: Textual entailment results.
Progress of Recent Weeks

- Joint model trained on harder tasks
- Single task models
- Joint many-task model

- First solution described in class
Obstacle: Duplicate Word Representations

- Different encodings for encoder (Word2Vec and GloVe word vectors) and decoder (softmax classification weights for words)
- Duplicate parameters/meaning
Tackling Obstacle by Tying Word Vectors

- Simple but theoretically motivated idea: tie word vectors and train single weights jointly

# Language Modeling With Tying Word Vectors

<table>
<thead>
<tr>
<th>MODEL</th>
<th>PARAMETERS</th>
<th>VALIDATION</th>
<th>TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>KN-5 (Mikolov &amp; Zweig)</td>
<td>2M</td>
<td>-</td>
<td>141.2</td>
</tr>
<tr>
<td>KN-5 + Cache (Mikolov &amp; Zweig)</td>
<td>2M</td>
<td>-</td>
<td>125.7</td>
</tr>
<tr>
<td>RNN (Mikolov &amp; Zweig)</td>
<td>6M</td>
<td>-</td>
<td>124.7</td>
</tr>
<tr>
<td>RNN+LDA (Mikolov &amp; Zweig)</td>
<td>7M</td>
<td>-</td>
<td>113.7</td>
</tr>
<tr>
<td>RNN+LDA+KN-5+Cache (Mikolov &amp; Zweig)</td>
<td>9M</td>
<td>-</td>
<td>92.0</td>
</tr>
<tr>
<td>Deep RNN (Pascaru et al., 2013a)</td>
<td>6M</td>
<td>-</td>
<td>107.5</td>
</tr>
<tr>
<td>Sum-Prod Net (Cheng et al., 2014)</td>
<td>5M</td>
<td>-</td>
<td>100.0</td>
</tr>
<tr>
<td>LSTM (medium) (Zaremba et al., 2014)</td>
<td>20M</td>
<td>86.2</td>
<td>82.7</td>
</tr>
<tr>
<td>LSTM (large) (Zaremba et al., 2014)</td>
<td>66M</td>
<td>82.2</td>
<td>78.4</td>
</tr>
<tr>
<td>VD-LSTM (medium, untied) (Gal, 2015)</td>
<td>20M</td>
<td>81.9 ± 0.2</td>
<td>79.7 ± 0.1</td>
</tr>
<tr>
<td>VD-LSTM (medium, untied, MC) (Gal, 2015)</td>
<td>20M</td>
<td>-</td>
<td>78.6 ± 0.1</td>
</tr>
<tr>
<td>VD-LSTM (large, untied) (Gal, 2015)</td>
<td>66M</td>
<td>77.9 ± 0.3</td>
<td>75.2 ± 0.2</td>
</tr>
<tr>
<td>VD-LSTM (large, untied, MC) (Gal, 2015)</td>
<td>66M</td>
<td>-</td>
<td>73.4 ± 0.0</td>
</tr>
<tr>
<td>CharCNN (Kim et al., 2015)</td>
<td>19M</td>
<td>-</td>
<td>78.9</td>
</tr>
<tr>
<td>VD-RHN (Zilly et al., 2016)</td>
<td>32M</td>
<td>72.8</td>
<td>71.3</td>
</tr>
<tr>
<td>Pointer Sentinel-LSTM(medium) (Merity et al., 2016)</td>
<td>21M</td>
<td>72.4</td>
<td>70.9</td>
</tr>
<tr>
<td>38 Large LSTMs (Zaremba et al., 2014)</td>
<td>2.51B</td>
<td>71.9</td>
<td>68.7</td>
</tr>
<tr>
<td>10 Large VD-LSTMs (Gal, 2015)</td>
<td>660M</td>
<td>-</td>
<td>68.7</td>
</tr>
<tr>
<td>VD-LSTM +REAL (medium)</td>
<td>14M</td>
<td>75.7</td>
<td>73.2</td>
</tr>
<tr>
<td>VD-LSTM +REAL (large)</td>
<td>51M</td>
<td><strong>71.1</strong></td>
<td><strong>68.5</strong></td>
</tr>
</tbody>
</table>
Obstacle: Necessary Inputs to QA

• We need to be able to understand text, **images and databases** to really answer all kinds of questions
Database QA

- **Question:** Who was drafted with the 3rd pick of the 1st round?
- **Answer:** Jayson Tatum

<table>
<thead>
<tr>
<th>Rnd</th>
<th>Pick</th>
<th>Player</th>
<th>Pos</th>
<th>Nationality</th>
<th>Team</th>
<th>School/club</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Markelle Fultz</td>
<td>PG</td>
<td>United States</td>
<td>Philadelphia 76ers (from Brooklyn via Boston)</td>
<td>Washington</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Lonzo Ball</td>
<td>PG</td>
<td>United States</td>
<td>Los Angeles Lakers</td>
<td>UCLA</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Jayson Tatum</td>
<td>SF</td>
<td>United States</td>
<td>Boston Celtics (from Sacramento via Philadelphia)</td>
<td>Duke</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>Josh Jackson</td>
<td>SF</td>
<td>United States</td>
<td>Phoenix Suns</td>
<td>Kansas</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>De'Aaron Fox</td>
<td>PG</td>
<td>United States</td>
<td>Sacramento Kings</td>
<td>Kentucky</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>Jonathan Isaac</td>
<td>SF/PF</td>
<td>United States</td>
<td>Orlando Magic</td>
<td>Florida State</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>Lauri Markkanen</td>
<td>PF/C</td>
<td>Finland</td>
<td>Minnesota Timberwolves (traded to Chicago Bulls)</td>
<td>Arizona</td>
</tr>
</tbody>
</table>
Seq2SQL Overview

How many engine types did Val Musetti use?

Seq2SQL
- Aggregation classifier
- SELECT column pointer
- WHERE clause pointer

SELECT
- COUNT
- Engine
- WHERE Driver = Val Musetti

Entrant
- Constructor
- Chassis
- Engine
- No
- Driver
Seq2SQL for DB QA
## Database QA Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Logical Form Accuracy</th>
<th>Test Execution Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attentional Seq2Seq</td>
<td>23.4</td>
<td>35.9</td>
</tr>
<tr>
<td>Augmented Pointer Network</td>
<td>42.8</td>
<td>52.8</td>
</tr>
<tr>
<td>Seq2SQL (no RL)</td>
<td>47.2</td>
<td>57.6</td>
</tr>
<tr>
<td>Seq2SQL</td>
<td>49.2</td>
<td>60.3</td>
</tr>
</tbody>
</table>
Recent Visual Question Answering

• Interpretable Reinforcement Learning Counter
• Good for discrete reasoning over images
• Interpretable Counting for Visual Question Answering, Alexander Trott, Caiming Xiong, Richard Socher. ICLR 2018
Results

How many people are in the picture?
ground truth = 4

How many people are wearing hats?
ground truth = 1

How many people are wearing glasses?
ground truth = 1

How many boats in the photo?
ground truth = 2
Results

How many motorcycles are in the picture?  
ground truth = 5

How many utility light poles are pictured?  
ground truth = 2

How many drains are visible in the picture?  
ground truth = 1

How many lamps are there?  
ground truth = 1
Obstacle: Architecture Engineering

• We don’t yet know the right model architecture for comprehensive QA & joint multitask learning

• Architecture Search is an active area of research but usually applied to simpler/known tasks, e.g.

• A Flexible Approach to Automated RNN Architecture Generation, Stephen Merity, Martin Schrimpf, James Bradbury, Richard Socher. (ICLR 2018 Workshop Track)
Architecture Search Overview

Candidate Architecture Generation
- Random Architecture Generator
- Incremental RL Generator

Ranking Function

Evaluator
- Full Training
  - Evaluation
  - 107.3
- Evaluation
  - 116.4
- Evaluation
  - 128.7

Architecture Results
When trained on language modeling
Interesting new building blocks like cos
Very different activation patterns!
Recent Work on Architecture Search

- **Efficient Neural Architecture Search via Parameter Sharing**
- Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, Jeff Dean
- Mon, 12 Feb 2018 :)
- **1000x more efficient and finds better models**
- Shares parameters between models instead of training from scratch
Lots of Limits for deepNLP

- Comprehensive QA
- Multitask learning
- Combined multimodal, logical and memory-based reasoning
- Learning from few examples
DeepNLP

Congratulations!

Good luck with the projects