COVID-19 Accommodations

• Classes, assignments, exams, etc. all remote through the rest of the semester
  • For this class, this will mean diligence in working remotely with teammates
  • PC5 (Cooperative Testing) has been moved back another week (now due 4/6)
  • PC6 (Sprint Review 3) will now be delivered as a YouTube video (now also 4/6)
  • PC7 (Final Presentations) will be a scheduled telecon with all of your team members, me, and one of the IAs (forthcoming)
    • Look at the Piazza post; you can schedule a 30 minute block on my calendar via the link there
    • Try to have most/all your team members present for that

• Grades now P/NRC with option to uncover letter grade
Recap

Natural Language Processing can be broken into several concepts:

• **Data**: Examples with labels
  • e.g., the tuple (“I want a burger” -> “order_food”) is an intent classification data point

• **Model**: A method for quantifying data
  • Features and Weights can be used
  • Contrived Example: “I want a burger”
    • “want” and “burger” => +2 for *order_food* intent
    • “burger” => +1 ; “want” => -1 for *get_nutrition* intent
  • Metrics like [tf-idf](https://en.wikipedia.org/wiki/Tf–idf) or [n-gram frequency](https://en.wikipedia.org/wiki/N-gram) can be useful for modeling

• **Inference**: deciding based on output from a model
  • We take concrete action based on numerical outputs
  • e.g., we infer the *intent* based on the model’s highest output value (the +2 above)

• **Learning**: revising model based on new data
  • How do you decide rules for the model?
Recap: Applying to Conversational AI

• **Intent Classification**
  • **Data:** tuples of (utterance, intent class)
  • **Model:** clustering, SVM, rules;
  • **Inference:** mapping from model output to intent class label

• **Slot Extraction**
  • **Data:** tuples of (token position, slot label)
  • **Model:** n-grams, **RNN**
  • **Inference:** RNN output mapped back to a vocabulary
One Slide Summary: Deep Learning and Embeddings

• **Machine Learning** is driven by applied **statistics**
  • Simple linear models are more interpretable (e.g., best-fit line)
  • More complex models yield better accuracy (trading off interpretability)

• **Deep Learning** is used in the NLP space to accurately represent language and classify intents and slots
  • Deep learning allows black-boxing of inputs to eliminate the need to derive costly features or rules
  • In particular, **Recurrent Neural Networks** and derivatives are state-of-the-art for NLU tasks

• **Embeddings** are numerical representations of NLU elements
  • Expressed as **fixed-dimensional vectors**
  • We say that we **embed** a token, sentence, or utterance into a **vector space** called the embedding space
Machine Learning

- **AI** is an application of **Machine Learning**
- **ML** is an application of statistics to **make predictions from existing data**
Machine Learning

- AI is an application of **Machine Learning**
- ML is an application of statistics to **make predictions from existing data**
Machine Learning

- **AI** is an application of **Machine Learning**
- **ML** is an application of statistics to **make predictions from existing data**
Machine Learning

• Must manually
  • Select features (e.g., age)
  • Hypothesize relationship
    (e.g., linear, piecewise, quadratic…)

• Time consuming, but interpretable

• Relies on domain knowledge
Machine Learning

• **Decision Trees** can be used to classify inputs (e.g., tall vs. not tall; high risk vs. low risk)

• **Example:** cardiovascular risk
  • Perhaps doctors have access to tons of old medical histories.
  • Might notice clusters in data (i.e., *domain expertise*):
    • Minimum systolic <= 90 -> high risk of death
    • Old with sinus tachycardia rhythm -> high risk
We use ML to teach software to make predictions.

Software learns from existing data.

### Supervised Learning

- Labeled data (e.g., tall vs. short)
  - (20 y.o., 6ft, tall)
  - (20 y.o., 5ft, short)

### Unsupervised Learning

- Unlabeled data (e.g., just points of data)
  - (20 y.o., 6ft)
  - (20 y.o., 5ft)
A **Model** allows us to quantify utterances. Depending on the specific model, we can visualize data.
Machine Learning in an NLU Context

A Model allows us to quantify utterances. Depending on the specific model, we can visualize data.
Machine Learning in an NLU Context

A **Model** allows us to quantify utterances. Depending on the specific model, we can visualize data.

How do we pick features? (hint: it’s hard)
Machine Learning in an NLU Context

Input Utterance: “I want a burger.”

Does it contain ‘want’?

- No
  - Contains ‘nutrition’?
    - X
    - A

- Yes
  - Contains ‘burger’, ‘sandwich’, ... ?
    - B
    - A

Decisions Trees can be used for intent classification; but ordinarily too many complex interactions exist.
Deep Learning Crash Course

• **Deep Learning** is a catch-all phrase that refers to **Neural Networks** that have multiple layers (c.f. deep pipeline from architecture)
Deep Learning Crash Course

- **Deep Learning** is a catch-all phrase that refers to **Neural Networks** that have multiple layers (c.f. deep pipeline from architecture)

![Diagram showing Machine Learning and Deep Learning processes]

- “depth” = more layers
Neural Network

• A Neural Network is a structure that feeds data through layers of simple mathematical operations (neurons), producing some set of numerical outputs that have some useful interpretation.

“Things that are lies”
Neural Network

• We use Deep Neural Networks (DNNs) to perform classification of intents, slot mapping, and slot-value pairing
  • DNNs can **learn from** (or “notice”) patterns in data that are not immediately obvious to human domain knowledge experts

• DNNs benefit from data
  • As long as features are represented, DNNs can learn which ones are important
Deeper in NNs

• Each cell in a NN is a simple combination of floating-point inputs
Tradeoffs in ML

- Deep neural networks **perform better**, but are **less explainable**
- Decision trees, linear models are all far easier to **explain**, but lack **expressive power**
Learning in NNs

• NNs can be characterized by the weights of connections from one layer to the next

• We use a loss function that captures the difference between known labels (i.e., what the output should be) and the output produced by the untrained NN

• We adjust the weights of the NN based on the loss function
  • Over time, the NN learns to capture patterns in the training data
DNNs for Classification

• DNN output layer can be interpreted as some class (e.g., “elephant” vs. “cat” or “noun” vs. “verb”)

• **Key idea:** with enough data, the NN can learn weights that can make future classifications

• Instead of developing complex rules or criteria for a **model**, the weights fall out of the learning process automatically!
DNNs for NLP

• DNNs are continuous mathematical structures
  • Lots of floating point operations

• Natural language is made up of letters
  • We need to represent natural language in some vector space
Recurrent Neural Networks

• RNNs are a type of NN that allow feeding information within a layer
  • (as opposed to feed-forward-only)

• Beneficial for **sequential data** (like sequences of tokens)
Why DNNs for NLP?

<table>
<thead>
<tr>
<th>Model</th>
<th>Slot F1 Score</th>
<th>Intent Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-model with decoder</td>
<td>96.89</td>
<td>98.99</td>
</tr>
<tr>
<td>Stack-Propagation + BERT</td>
<td>96.10</td>
<td>97.50</td>
</tr>
<tr>
<td>Stack-Propagation</td>
<td>95.90</td>
<td>96.90</td>
</tr>
<tr>
<td>Attention Encoder-Decoder NN</td>
<td>95.87</td>
<td>98.43</td>
</tr>
<tr>
<td>SF-ID (BLSTM) network</td>
<td>95.80</td>
<td>97.76</td>
</tr>
<tr>
<td>Capsule-NLU</td>
<td>95.20</td>
<td>95.00</td>
</tr>
<tr>
<td>Joint GRU model(W)</td>
<td>95.49</td>
<td>98.10</td>
</tr>
<tr>
<td>Slot-Gated BLSTM with Attention</td>
<td>95.20</td>
<td>94.10</td>
</tr>
<tr>
<td>Joint model with recurrent slot label context</td>
<td>94.64</td>
<td>98.40</td>
</tr>
<tr>
<td>Recursive NN</td>
<td>93.96</td>
<td>95.40</td>
</tr>
<tr>
<td>Encoder-labeler Deep LSTM</td>
<td>95.66</td>
<td>NA</td>
</tr>
<tr>
<td>RNN with Label Sampling</td>
<td>94.89</td>
<td>NA</td>
</tr>
<tr>
<td>Hybrid RNN</td>
<td>95.06</td>
<td>NA</td>
</tr>
<tr>
<td>RNN-EM</td>
<td>95.25</td>
<td>NA</td>
</tr>
<tr>
<td>CNN-CRF</td>
<td>94.35</td>
<td>NA</td>
</tr>
</tbody>
</table>
Neural Networks, Deep Learning, RNNs

- **Neural Networks** underly the majority of modern AI techniques
- NNs allow **black-boxing** a lot of **domain-expertise** required in other ML techniques
- **DNNs** are merely **bigger NNs** that have lots of intermediate layers
  - Requirement: need a **lot** of data
- **RNNs** are a type of NN that have a particular property: **loops in the graph**
  - NB: loops imply **statefulness**

---

<table>
<thead>
<tr>
<th>Consumption</th>
<th>CO$_2$e (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air travel, 1 passenger, NY↔SF</td>
<td>1984</td>
</tr>
<tr>
<td>Human life, avg, 1 year</td>
<td>11,023</td>
</tr>
<tr>
<td>American life, avg, 1 year</td>
<td>36,156</td>
</tr>
<tr>
<td>Car, avg incl. fuel, 1 lifetime</td>
<td>126,000</td>
</tr>
</tbody>
</table>

| Training one model (GPU)                      |               |
| NLP pipeline (parsing, SRL)                   | 39            |
| w/ tuning & experimentation                   | 78,468        |
| Transformer (big)                             | 192           |
| w/ neural architecture search                 | 626,155       |

Table 1: Estimated CO$_2$ emissions from training common NLP models, compared to familiar consumption.\(^1\)

---

\(^1\)Sources: (1) Air travel and per-capita consumption: https://bit.ly/2Hw0xWc; (2) car lifetime: https://bit.ly/2Qbr0w1.
Embeddings

• A **word embedding** is a way of mapping words to vectors
  • word2vec/GLoVE: unsupervised methods based on corpus statistics
  • Built-in embedding layers: tensorflow/keras support “Embedding layers”
Desired Properties of Embeddings

• **Semantic Relationships Represented**
  • Related words should be “Close” in a Euclidean sense
    • “cloud” – “sky” < “cloud” – “steak”
  • Unrelated words should be far away
  • Arithmetic should be possible
    • “cloud” + “sky” – “sun” might yield something near “rain”

• **Compact Representation**
  • We want to do quick math to compute word relationships
  • We need a representation suitable for DNNs

• **Mappable**
  • We need to move to and from word and embedding space quickly
  • A DNN layer may be an embedding... how do we turn it into a word?
Visualizing Word Embeddings

- Let’s embed the word “King”
Visualizing Word Embeddings

• Let’s embed the word “King”
Visualizing Word Embeddings

• Let’s embed the word “King”
BERT: State-of-the-art embeddings
DNNs for NLP

• We can use DNNs for
  • **Classification**: Fixed number of intents (once you build your state graph)
    • Embed utterances; model can learn words (and neighbors in the embedding space!) that distinguish intents
    • The state graph just makes this part simpler (you only need to consider between the intent classes of child nodes in any given state)
  
• **Slot Extraction**: Train by labeling portions of utterance
  • Yo fam get me a burger.
  • O B:person O B:person O B:food

  • Model learns a combination of vocabulary and contextual hints