Top Down Intro to Neural Semantic Parsing

Ofir
Plan

1. Intro to NNs and RNNs
2. Model 1: seq2seq
3. Model 2: seq2tree
4. How to improve these models with attention
5. Results & future work
NNs for supervised learning

Examples and labels: \{((x^{(n)}, y^{(n)}))\}

Neural network: \(f(x^{(n)}; W)\)

Loss on an example: \(L(f(x^{(n)}; W), y^{(n)})\)

Objective: Find \(W\) that minimizes \(\sum L(f(x^{(n)}, W), y^{(n)})\)

Generalization
Language modeling

U.S. intelligence agencies have ...

High Probability:

- a
- received

Low Probability:

- Antarctica
- Dog
- Cat
Language models continuously make predictions

U.S. intelligence agencies have not...

High Probability:

- received
- announced
- discovered

Low Probability:

- a
- the
- Dog
Simple model

Distribution Over Next Word

\[ f(\text{Word}; W) \]

Word
Probability that next word is…
president = 0.07
government = 0.05
the = 0.0003

\[ f(\text{"U.S."} ; W) \]
Simple model: Data

U.S. intelligence agencies have not corroborated the allegations about the president-elect's personal life and finances……

{(U.S., intelligence),
(intelligence, agencies),
(agencies, have),
...
}
Small implementation detail:

Each word has an ID between 1 and $|\text{size of vocab}|$.

Vectors of same length

<table>
<thead>
<tr>
<th>Word</th>
<th>Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>hello</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>the</td>
<td>(0,0,1)</td>
</tr>
</tbody>
</table>

0.03 0.04 0.02 0.03 0.09 0.03 0.04 0.02 0.08
Loss at every timestep of decoder:

\[-\ln(P(\text{correct\_word}))\]

\[P(\text{correct\_word}) = 1 \rightarrow \text{Loss} = -\ln(1) = 0\]

\[P(\text{correct\_word}) = 0 \rightarrow \text{Loss} = -\ln(0) = \infty\]
Problem: No memory

Intelligence = 0.05
Government = 0.3
President = 0.07

Intelligence = 0.09
Government = 0.02
President = 0.04

Intelligence = 0.05
Government = 0.2
President = 0.09

“U.S.”

“intelligence”

“agencies”
Solution

State: represents what we’ve seen until now.

Intelligence = 0.05  
Government = 0.3  
President = 0.07

Intelligence = 0.09  
Government = 0.02  
President = 0.04

Intelligence = 0.05  
Government = 0.2  
President = 0.09

"U.S."  
"intelligence"  
"agencies"
At train time:

\[ \text{target}_t = \text{Intelligence} \]

\[ \text{input}_t = \text{"U.S."} \]

\[ \text{Offset by 1} \]

\[ \text{agencies} \]

\[ \text{"intelligence"} \]

\[ \text{have} \]

\[ \text{"agencies"} \]
At test time: “Who is Matt _ ?”

Damon 0.05
Lauer 0.041
Terry 0.040

input:

“Who”

“is”

“Matt”
Comparison:

1 input, 1 output

2 inputs, 2 outputs
Intro done. Questions?
Semantic parsing as a seq2seq problem:

what microsoft jobs do not require a bscs? →

answer(J,(company(J,'microsoft'),job(J),not((req_deg(J,'bscs'))))))

No knowledge base involved in this problem.
Another example

dallas to san francisco leaving after 4 in the afternoon please →

(lambda $0 e (and ((departure time $0) 1600:ti) (from $0 dallas:ci) (to $0 san francisco:ci)))
input/output is a series of tokens:

["what", "microsoft", "jobs", "do", "not", "require", "a", "bscs", "?", "</s>"] →

["<s>" , "answer" , "(" , "J" , "" , "(" , "company" , "...", "</s>" ]

EOS token
Lets look at a simple example:

My brother →

האח שלי
Solution?

target_{t}: 

input_{t}: 

My brother
What we want

An example run of this model (trained):
Encoder

what
micr-
osoft
jobs
do

...
Representation

what microsof jobs do ...
Decoder

\[
\begin{array}{cccc}
\text{answer} & 0.36 & \text{answer} & 0.01 \\
( & 0.32 & ) & 0.91 \\
J & 0.07 & J & 0.01 \\
\text{answer} & 0.01 & \text{answer} & 0.01 \\
( & 0.21 & ) & 0.81 \\
J & 0.01 & J & 0.01 \\
\end{array}
\]

\[
\begin{array}{cc}
\text{answer} & 0.01 \\
( & 0.02 \\
J & 0.01 \\
\end{array}
\]
How is this trained?

These are the true labels

- answer 0.36
- (J 0.32 J 0.07)
- answer 0.01
- (J 0.91 J 0.01)
- <s> 0.96
- (J 0.21 J 0.81)

what ... <s>
Loss at every timestep of decoder:

\[-\ln(P(\text{correct\_word}))\]

\[P(\text{correct\_word}) = 1 \rightarrow \text{Loss} = -\ln(1) = 0\]

\[P(\text{correct\_word}) = 0 \rightarrow \text{Loss} = -\ln(0) = \infty\]
what
...
</s>

<s>
answer 0.36
( 0.32
J 0.07
)
answer 0.01
( 0.91
J 0.01
)

answer

L = 3
L = 0.5
L = 7

….

L = 3
answer 0.36
( 0.32
J 0.07
)
answer 0.01
( 0.91
J 0.01
)

L = 0.5

L = 7
</s> 0.96
( 0.21
J 0.81
)

)
Small note about the network:

Also for decoder.

Same function
Top prob isn’t guaranteed to be optimal, so use beam search
End of seq2seq model

Questions?
We modeled both sentences & logical forms as sequences.
Logical forms as trees

But logical forms represent trees:

Lambda expression:

\(((\lambda \ x \ . \ (\lambda \ y \ . \ x)) \ (\lambda \ z \ . \ z)) \ (\lambda \ a \ . \ a)\)
So let's build a decoder that instead of outputting a sequence would output a tree.
What we want

[“what”, “microsoft”, “jobs”, “do”, “not”, “require”, “a”, “bscs”, “?”, “<s>”]
We will use a sequence model to generate a tree
How?

Layer by layer, starting from the top.
Preprocessing

Take output sequence, replace all “( … )” with “<n>”.

Decoding: A B (C)
Parent feeding

Every unit gets not only the output of its previous timestep, but also its parent’s output.
End of tree model

Questions?
Attention

“You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!” -Raymond Mooney
Reminder

The length is not constant


[“answer”, “(“, “J” “), “(“, “company”, ….., “</s>” ]

Encoder

Decoder

This is always the same size.

[“answer”, “(“, “J” “), “(“, “company”, ….., “</s>” ]
Bahdanu et. al. (2014) showed that as sentence length passes 20 words, translation quality drops.
How can we fix this?

Let’s “attend” at every timestep only to the relevant words.
Simple example

She wanted a green bicycle. →

היא רצתה אופניים ירוקים
Simple example

She wanted a green bicycle. →

היא רצתה אופניים ירוקים
This also occurs in logical forms!

what microsoft jobs do not require a bscs? →

answer(J,(company(J,'microsoft'),job(J),not((req_deg(J,'bscs'))))))
Reminder:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;s&gt;</td>
<td>answer 0.36</td>
<td>answer 0.01</td>
</tr>
<tr>
<td></td>
<td>( 0.32</td>
<td>( 0.91</td>
</tr>
<tr>
<td>J 0.07</td>
<td>J 0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;&lt;/s&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;s&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>answer 0.01</td>
</tr>
<tr>
<td></td>
<td>( 0.96</td>
<td></td>
</tr>
<tr>
<td>J 0.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>)</td>
</tr>
</tbody>
</table>

what ... </s>
Attention decoder:

\[ h_{t-1} \rightarrow \text{context}_t \rightarrow \text{Distribution for next word} \rightarrow h_t \]

\[ y_{t-1} \rightarrow h_t \]
How do we calculate the context vector?
Encode the words once at the beginning

She wanted a green...
Now for every word we have a representation

$v_1$    $v_2$    ....    $<$s$>$
t=1

60 3 1 3 1 ...

v_1 v_2 ....

She wanted ...

\[ a(v_i, h_{t-1}) \]
t=2

70  70  1 3 1 ...

v_1  v_2  ....

She  wanted  ...  </s>

a(v_i, h_{t-1})
t=1

\[ a(v_i, h_{t-1}) \]

\[
\begin{array}{cccc}
0.90 & 0.02 & 0.001 & 0.02 & 0.001 \\
60 & 3 & 1 & 3 & 1 \ldots
\end{array}
\]

\[ \text{normalize} \]

\[ \text{context}_t = \sum \text{normalized}(a(v_i, h_{t-1})) \cdot v_i \]

She wanted \ldots \langle/s\rangle
Attention decoder:

\[
\begin{align*}
    h_{t-1} & \quad \rightarrow \quad h_t \\
    y_{t-1} & \quad \rightarrow \quad h_t \\
    \text{context}_t & \quad \rightarrow \quad h_t
\end{align*}
\]

Distribution for next word
Decoder: now with attention

answer 0.36  answer 0.01  answer 0.01  answer 0.01
( 0.32  ( 0.91  ( 0.21  ( 0.02
J 0.07  J 0.01  J 0.81  J 0.01

<s> answer ( J

</s> 0.96
( 0.02
J 0.01

...
(c) what is the earliest flight from ci0 to ci1 tomorrow
(b) what’s first class fare round trip from ci0 to ci1
Accuracy: The proportion of the input sentences that are correctly parsed to their gold standard logical forms

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCISSOR (Ge and Mooney, 2005)</td>
<td>72.3</td>
</tr>
<tr>
<td>KRISP (Kate and Mooney, 2006)</td>
<td>71.7</td>
</tr>
<tr>
<td>WASP (Wong and Mooney, 2006)</td>
<td>74.8</td>
</tr>
<tr>
<td>λ-WASP (Wong and Mooney, 2007)</td>
<td>86.6</td>
</tr>
<tr>
<td>LNLZ08 (Lu et al., 2008)</td>
<td>81.8</td>
</tr>
<tr>
<td>ZC05 (Zettlemoyer and Collins, 2005)</td>
<td>79.3</td>
</tr>
<tr>
<td>ZC07 (Zettlemoyer and Collins, 2007)</td>
<td>86.1</td>
</tr>
<tr>
<td>UBL (Kwiatkowski et al., 2010)</td>
<td>87.9</td>
</tr>
<tr>
<td>FUBL (Kwiatkowski et al., 2011)</td>
<td>88.6</td>
</tr>
<tr>
<td>KCASL3 (Kwiatkowski et al., 2013)</td>
<td>89.0</td>
</tr>
<tr>
<td>DCS+L (Liang et al., 2013)</td>
<td>87.9</td>
</tr>
<tr>
<td>TISP (Zhao and Huang, 2015)</td>
<td>88.9</td>
</tr>
<tr>
<td>SEQ2SEQ</td>
<td>86.6</td>
</tr>
<tr>
<td>− attention</td>
<td>72.9</td>
</tr>
<tr>
<td>− argument</td>
<td>68.6</td>
</tr>
<tr>
<td>SEQ2TREE</td>
<td>87.1</td>
</tr>
<tr>
<td>− attention</td>
<td>76.8</td>
</tr>
</tbody>
</table>

Table 3: Evaluation results on GEO. 10-fold cross-validation is used for the systems shown in the top half of the table. The standard split of ZC05 is used for all other systems.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZC07 (Zettlemoyer and Collins, 2007)</td>
<td>82.6</td>
</tr>
<tr>
<td>UBL (Kwiatkowski et al., 2010)</td>
<td>71.4</td>
</tr>
<tr>
<td>FUBL (Kwiatkowski et al., 2011)</td>
<td>82.8</td>
</tr>
<tr>
<td>GUSP-FULL (Poon, 2013)</td>
<td>74.8</td>
</tr>
<tr>
<td>GUSP++ (Poon, 2013)</td>
<td>83.5</td>
</tr>
<tr>
<td>TISP (Zhao and Huang, 2015)</td>
<td>84.2</td>
</tr>
<tr>
<td>SEQ2SEQ</td>
<td>84.2</td>
</tr>
<tr>
<td>− attention</td>
<td>75.7</td>
</tr>
<tr>
<td>− argument</td>
<td>72.3</td>
</tr>
<tr>
<td>SEQ2TREE</td>
<td>84.6</td>
</tr>
<tr>
<td>− attention</td>
<td>77.5</td>
</tr>
</tbody>
</table>

Table 4: Evaluation results on ATIS.

<table>
<thead>
<tr>
<th>Method</th>
<th>Channel</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieval</td>
<td>28.9</td>
<td>20.2</td>
</tr>
<tr>
<td>phrasal</td>
<td>19.3</td>
<td>11.3</td>
</tr>
<tr>
<td>sync</td>
<td>18.1</td>
<td>10.6</td>
</tr>
<tr>
<td>classifier</td>
<td>48.8</td>
<td>35.2</td>
</tr>
<tr>
<td>posclass</td>
<td>50.0</td>
<td>36.9</td>
</tr>
<tr>
<td>− attention</td>
<td>54.0</td>
<td>37.9</td>
</tr>
<tr>
<td>− argument</td>
<td>53.9</td>
<td>38.6</td>
</tr>
<tr>
<td>SEQ2TREE</td>
<td>55.2</td>
<td>40.1</td>
</tr>
<tr>
<td>− attention</td>
<td>54.3</td>
<td>38.2</td>
</tr>
</tbody>
</table>

(a) Omit non-English.

<table>
<thead>
<tr>
<th>Method</th>
<th>Channel</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieval</td>
<td>36.8</td>
<td>25.4</td>
</tr>
<tr>
<td>phrasal</td>
<td>27.8</td>
<td>16.4</td>
</tr>
<tr>
<td>sync</td>
<td>26.7</td>
<td>15.5</td>
</tr>
<tr>
<td>classifier</td>
<td>64.8</td>
<td>47.2</td>
</tr>
<tr>
<td>posclass</td>
<td>67.2</td>
<td>50.4</td>
</tr>
<tr>
<td>− attention</td>
<td>68.8</td>
<td>50.5</td>
</tr>
<tr>
<td>− argument</td>
<td>68.7</td>
<td>48.9</td>
</tr>
<tr>
<td>SEQ2TREE</td>
<td>69.6</td>
<td>51.4</td>
</tr>
<tr>
<td>− attention</td>
<td>68.7</td>
<td>49.5</td>
</tr>
</tbody>
</table>

(b) Omit non-English & unintelligible.

<table>
<thead>
<tr>
<th>Method</th>
<th>Channel</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieval</td>
<td>43.3</td>
<td>32.3</td>
</tr>
<tr>
<td>phrasal</td>
<td>37.2</td>
<td>23.5</td>
</tr>
<tr>
<td>sync</td>
<td>36.5</td>
<td>24.1</td>
</tr>
<tr>
<td>classifier</td>
<td>79.3</td>
<td>66.2</td>
</tr>
<tr>
<td>posclass</td>
<td>81.4</td>
<td>71.0</td>
</tr>
<tr>
<td>− attention</td>
<td>88.3</td>
<td>73.8</td>
</tr>
<tr>
<td>− argument</td>
<td>86.8</td>
<td>74.9</td>
</tr>
<tr>
<td>SEQ2TREE</td>
<td>89.7</td>
<td>78.4</td>
</tr>
<tr>
<td>− attention</td>
<td>87.6</td>
<td>74.9</td>
</tr>
</tbody>
</table>

(c) ≥ 3 turkers agree with gold.

Table 5: Evaluation results on IFTTTT.
Problems

- Rare words
- Cant remember past attention values
End. Questions?