Parsing

Neural Transition-Based Dependency Parsing

b

"I parsed this sentence correctly"

Stack	Buffer	New dependency	Transition
[ROOT]	[I, parsed, this, sentence, correctly]		Initial conf
[ROOT, I]	[parsed, this, sentence, correctly]		shift
[ROOT, I, parsed]	[this, sentence, correctly]		shift
[ROOT, parsed]	[this, sentence, correctly]	parsed→I	left-arc
[ROOT, parsed, this]	[sentence, correctly]		shift
[ROOT, parsed, this, sentence]	[correctly]		shift
[ROOT, parsed, sentence]	[correctly]	sentence→this	left-arc
[ROOT, parsed]	[correctly]	parsed→sentence	right-arc
[ROOT, parsed, correctly]	[]		shift
[ROOT, parsed]	[]	parsed→correctly	right-arc
[ROOT]	[]	ROOT→parsed	right-arc

c

A sentence containing n words will be parsed in $n + n = 2n$ steps.

In order to parse a sentence we must use SHIFT for each word once to push them into the stack - that's n steps. Words are taken out of the stack by the LEFT-ARC/RIGHT-ARC transition - each transition takes out one word out of the stack, so n transitions are needed to remove all the words back from the stack.

g.

```
Epoch 10 out of 10
100%|
Average Train Loss: 0.06758635246710956
Evaluating on dev set
1445850it [00:00, 41293865.08it/s]
- dev UAS: 88.44
New best dev UAS! Saving model.


==================================================
TESTING
==================================================
Restoring the best model weights found on the dev set
Final evaluation on test set
2919736it [00:00, 59530225.72it/s]
- test UAS: 89.08
```
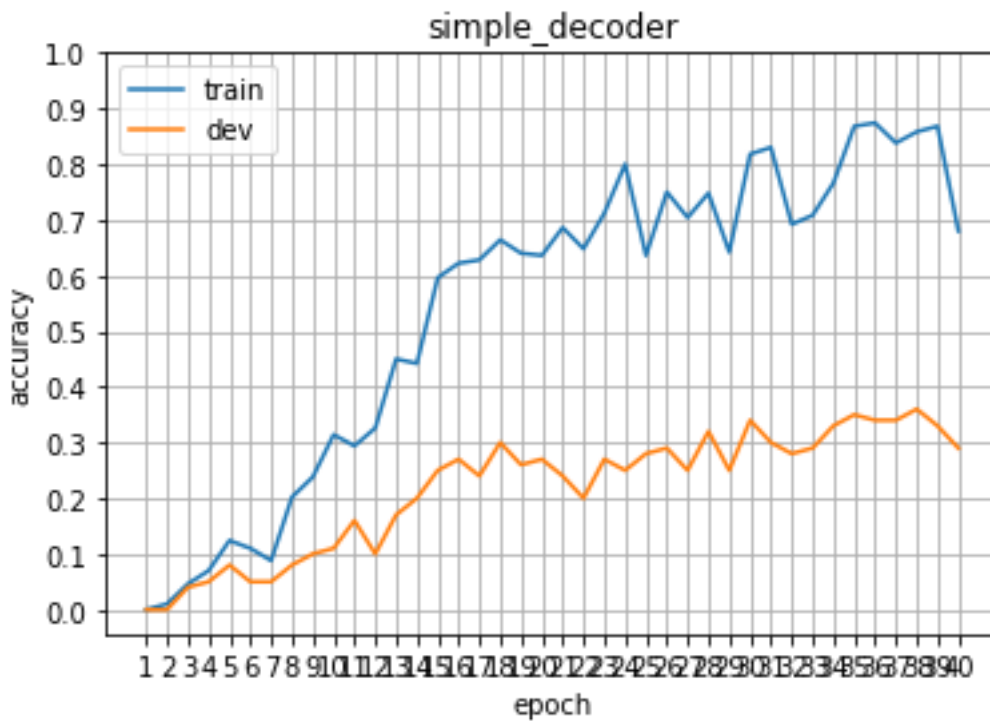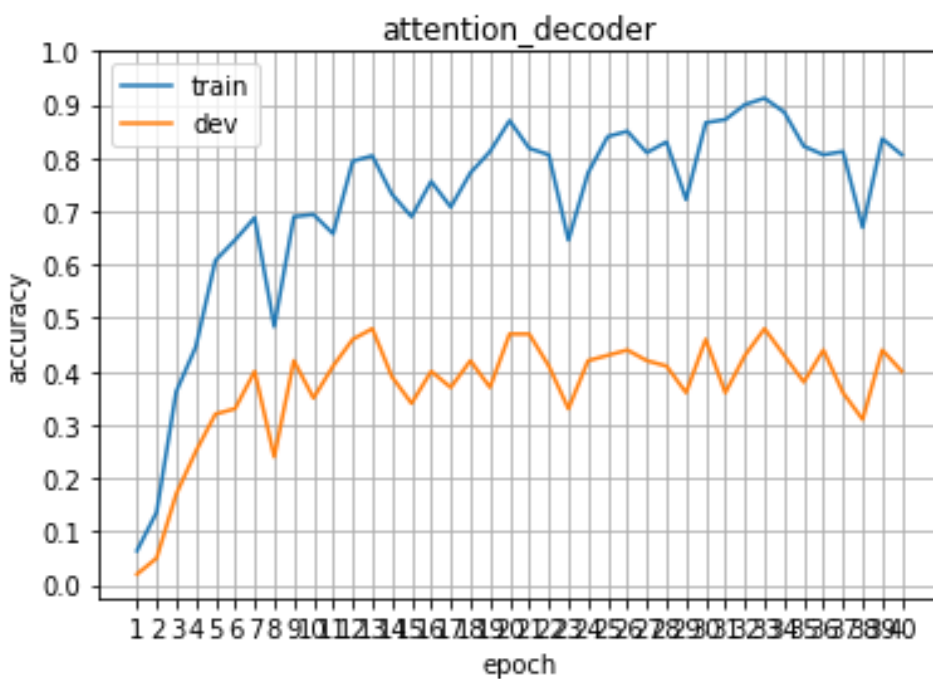
h.

  i. **Error type:** Prepositional Phrase Attachment Error
      Incorrect dependency: guy → from
      Correct dependency: loan → from

  ii. **Error type:** Modifier Phrase Attachment Error
      Incorrect dependency: elements → most
      Correct dependency: crucial → most

  iii. **Error type:** Verb Phrase Attachment Error
       Incorrect dependency: wedding → fearing
       Correct dependency: heading → fearing

  iv. **Error type:** Coordination Attachment Error
      Incorrect dependency: makes → rescue
      Correct dependency: rush → rescue

Q2 – Semantic Parsing

Graph Q1:

Graph Q2:

Graph Q3: