

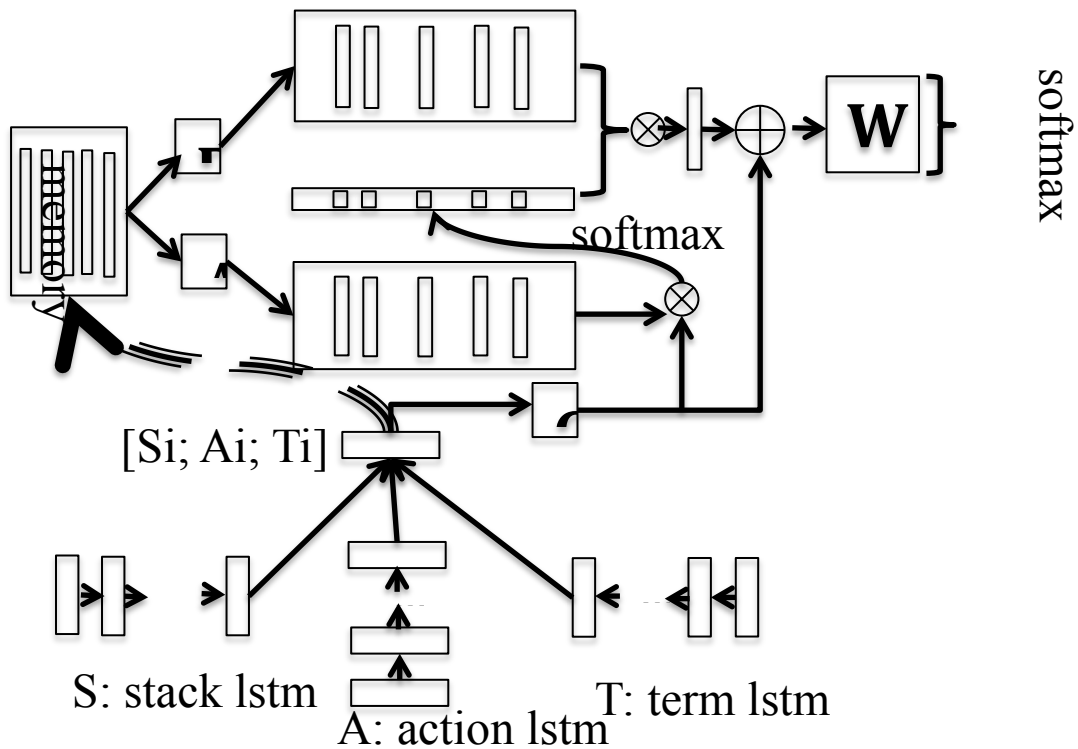
Progress of RNNG with memory network

Yang Feng

RELEASE ON 11-07-2016

The current version

Cascade training of lstm and memory network, which means first train lstm as shown in the paper (Dyer et al. 2016) and then train memory network with the parameters of lstm fixed.



Experiment configuration

Memory size	Memory dimension	nhops	Graph memory	Whether succeed	Reason
100000	768	2	512M	No	Graph is out of memory
10000	768	2	2048M	No	Ditto
5000	768	2	10240M	No	The cluster node is out of memory
5000	768	2	4096M	Under training	---

Also, I optimized my code to reduce memory use, and now the running status is

System	Total memory	Memory for lstm states	Speed
Rnng-gen-mem	20G	5~8G	4.5s/sentence
Rnng-gen	2G	--	0.3s/sentence

There are 39832 sentences in the training set and it takes about 50 hours to run a round and it takes at least 10 rounds to converge, so the training time is at least 500 hours. We need to speed up!

To do list:

- To speed up. Change Concatenation lstm states to linear transform. In this way, the memory can reduce from 5000×768 to 5000×256 .
- To store in memory actions of subtrees instead of the whole sentences.
- To try different strategies to select memory contents;
- To try different memory size;
- To implement batch training;
- To jointly train memory networks and lstm models.

[RELEASE ON 11-14-2016](#)

Test the speed of rnng+mn

System	Memory size	Memory dimension	layers	Memory	Speed ms/sentence	F1
Rnng	-	-	-	-	300	
Rnng+mn	1	768	1	9.7G	148 (heavy load node)	
Rnng+mn	10	768	1	9.2G	80	
Rnng+mn	100	768	1	9.7G	114	
Rnng+mn	1000	768	1	9.7G	400	
Rnng+mn	2000	768	1	9.7G	900	
Rnng+mn	5000	768	1	9.7G	1400	

Experiments ongoing:

Rnng+mn training=39832 memory=1000,random

Epoch 14.8 llh: 190535 → 173904, not converge yet

Test: killed, reduce data size from 2416 → 400

Rnng+mn training=39832 memory=2000,random

Epoch 4.0 llh: 189953 → 179480

Rnng training=5000

Failed when test, as it doesn't allow unk.

Modified the code and retrained to allow unk.

Rnng+mn training=5000 memory=1000,random

Retrained as it uses the model produced by rnng.

Rnng+mn training=5000 memory=1000,wrong

Retrained as it uses the model produced by rnng.

Rnng+mn training=5000 memory=1000,wrong, swap frequently

Ready for training

Rnng+nm training=5000 memory=random-selected

Ready for training