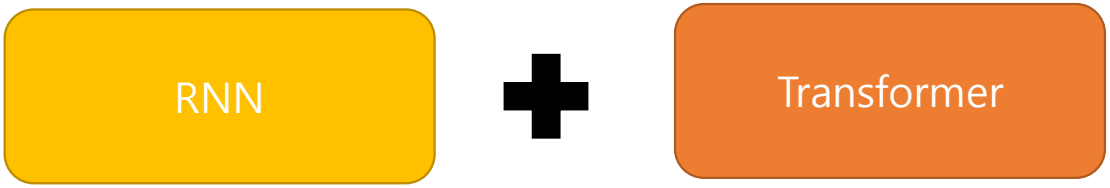
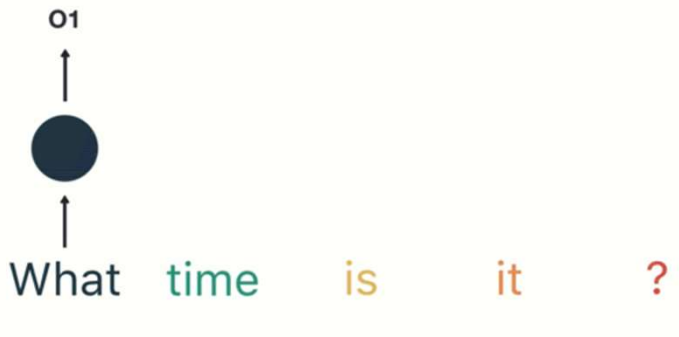
Universal Transformers

Mostafa Dehghani University of Amsterdam
Stephan Gouws DeepMind
Orial Vinyals DeepMind
Jakob Uszkoreit Google Brain
Lukasz Kaier Google Brain

Introduction



- RNN
  Sequential => Slow
- Transformer
  - Parallelization => Fast and Global Receptive Field
- Goal
  - RNN의 재귀구조를 Transformer와 결합
  - Algorithmic 문제와 Natural Language Understanding 문제에서 성능이 좋아졌다.



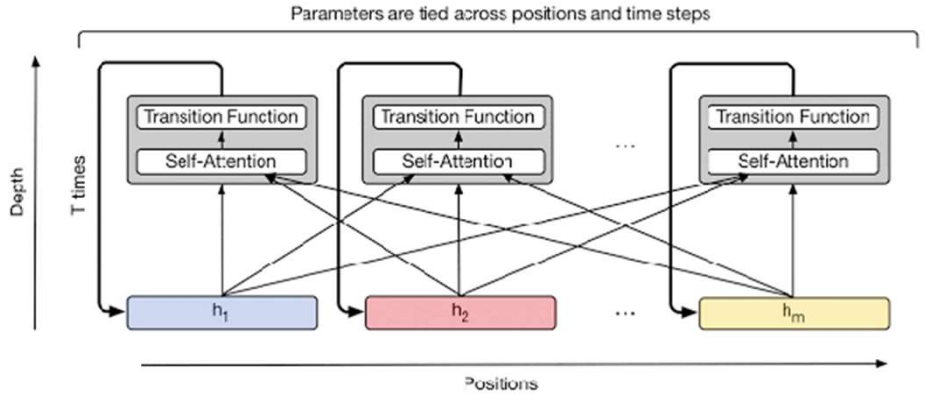
Model

RNN

Recur over positions in the sequence

Universal Transformer

Recur over each position

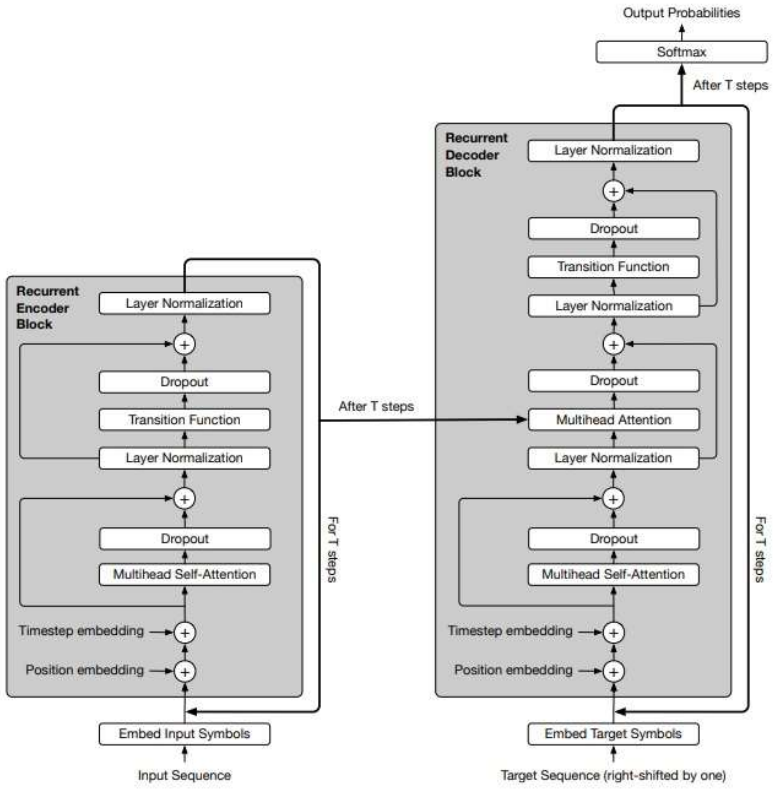


UT VS Transformer

$$1 \leq i \leq m, 1 \leq t \leq T, 1 \leq j <= d/2$$

$$P^t_{i,2j} = \sin(i/10000^{2j/d}) + \sin(t/10000^{2j/d})$$

$$P^t_{i,2j+1} = \cos(i/10000^{2j/d}) + \cos(t/10000^{2j/d})$$



	Transformer	UT
Weight Sharing	X	Self-Attention, Transition Function
# Updates	# Stacks	Variable Length
	Position Embedding	Position Embedding + Timestep Embedding
Transition Function	Feed Forward Network	Feed Forward Network or Convolution

Dynamic Halting

- Adaptation Computation Time (Graves, 2016)
- Transformer or RNN => # Stacks에 의해 연산량 결정
- Dynamic Halting => 연산량을 모델 스스로 결정
- 어렵거나 중요한 symbol운 더 많은 연산
- 중간에 중단되면 그 symbol은 다음 step에도 같은 state를 사용
- Halting probability가 threshold를 넘으면 중단
- 모든 symbol이 중단될 때까지 Computation

(1/6) Experiments

Story

Sandra journeyed to the hallway.
Mary went to the bathroom.
Mary took the apple there.
Mary dropped the apple.

Question

Where is the apple?

Output

bathroom

- bAbi 데이터셋
  - 모델의 Reasoning 능력 측정

(1/6) Experiments

Model	10K examples		1K examples	
	train single	train joint	train single	train joint
Previous best results:				
QRNet (Seo et al., 2016)	0.3 (0/20)	-	-	-
Sparse DNC (Rae et al., 2016)	-	2.9 (1/20)	-	-
GA+MAGE Dhingra et al. (2017)	-	-	8.7 (5/20)	-
MemN2N Sukhbaatar et al. (2015)	-	-	-	12.4 (11/20)
Our Results:				
Transformer (Vaswani et al., 2017)	15.2 (10/20)	22.1 (12/20)	21.8 (5/20)	26.8 (14/20)
Universal Transformer (this work)	0.23 (0/20)	0.47 (0/20)	5.31 (5/20)	8.50 (8/20)
UT w/ dynamic halting (this work)	**0.21 (0/20)**	**0.29 (0/20)**	**4.55 (3/20)**	**7.78 (5/20)**

- Metric
  - Error rate
  - # Failure Task (Total 20 Task , failure = error rate> 5%)
- Transformer는 잘 못함
- UT는 SOTA

(1/6) Experiments

Story

Sandra journeyed to the hallway.
Mary went to the bathroom.
Mary took the apple there.
Mary dropped the apple.

Question

Where is the apple?

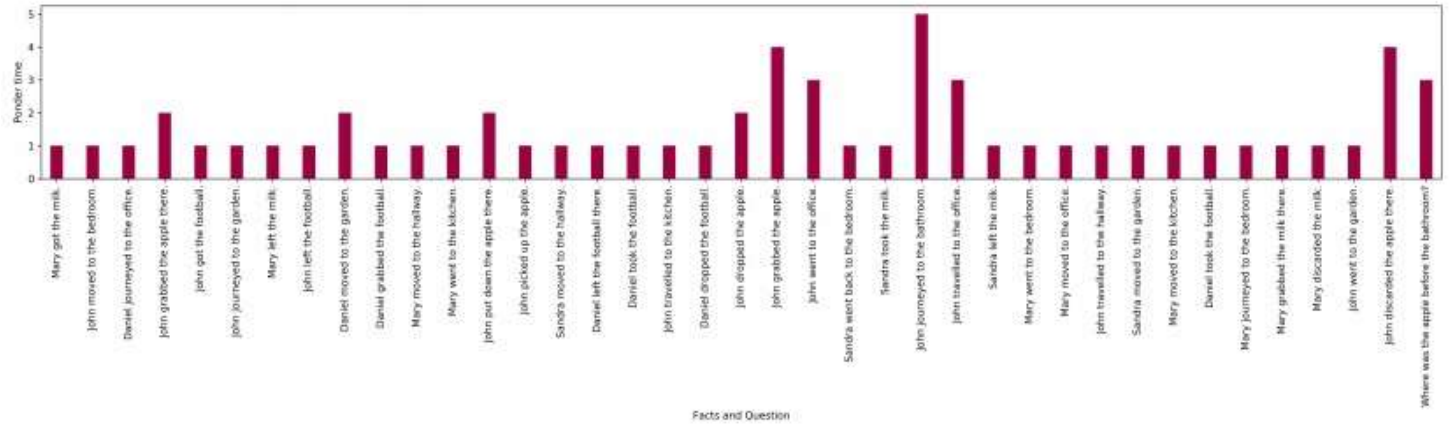
Output

bathroom

- Supporting Fact
  - 문제에 답하기 위해 필요한 문장의 수
  - 난이도 증가
- Supporting Fact 증가함에 Step 수 증가
- 문제 난이도에 따라 모델이 스스로 연산량 조절

Supporting Fact	# steps
1	2.3±0.8
2	3.1±1.1
3	3.8±2.2

(1/6) Experiments



- 각 Sentence가 몇 번의 step 이후에 중단되는지
- 대부분의 문장은 한 두 step만 봄
  - 중요하지 않은 문장은 무시
  - 중요한 문장에 집중

(2/6) Experiments

Model	Number of attractors						
	0	1	2	3	4	5	Total
Previous best results (Yogatama et al., 2018):							
Best Stack-RNN	*0.994*	0.979	0.965	0.935	0.916	0.880	0.992
Best LSTM	0.993	0.972	0.950	0.922	0.900	0.842	0.991
Best Attention	**0.994**	**0.977**	0.959	0.929	0.907	0.842	**0.992**
Our results:							
Transformer	0.973	0.941	0.932	0.917	0.901	0.883	0.962
Universal Transformer	0.993	0.971	**0.969**	0.940	0.921	0.892	**0.992**
UT w/ ACT	**0.994**	0.969	0.967	**0.944**	**0.932**	**0.907**	**0.992**
Δ (UT w/ ACT - Best)	0	-0.008	0.002	0.009	0.016	0.027	-

- Subject-Verb Agreement
  - 주어가 주어졌을 때 동사가 단수형/복수형
  - 문법 구조 파악
  - Attractors ↑ == 난이도 ↑
  - Transformer는 LSTM with Attention 보다 못함
  - UT는 Attractors ↑ 할수록 다른 모델보다 좋음

No attractor: The **boy** **smiles**.
One attractor: The **number** of men **is** not clear.
Two attractors: The **ratio** of men to women **is** not clear.
Three attractors: The **ratio** of men to women and children **is** not clear.

(3/6) Experiments

Context:

"Yes, I thought I was going to lose the baby."
"I was scared too," he stated, sincerity flooding his eyes.
"You were?" "Yes, of course. Why do you even ask?"
"This baby wasn't exactly planned for."

Target sentence:

"Do you honestly think that I would want you to have a _____?"

Target word:

miscarriage

- LAMBADA
  - Context가 주어졌을 때 Target sentence의 마지막 단어 예측
  - Target sentence만 이용해서는 풀 수 없음
  - 컨텍스트 전반을 이해해야 풀 수 있음

(3/6) Experiments

Model	LM Perplexity & (Accuracy)			RC Accuracy		
	control	dev	test	control	dev	test
Neural Cache (Grave et al., 2016)	**129**	139	-	-	-	-
Dhingra et al. Dhingra et al. (2018)	-	-	-	-	-	0.5569
Transformer	142 (0.19)	5122 (0.0)	7321 (0.0)	0.4102	0.4401	0.3988
LSTM	138 (0.23)	4966 (0.0)	5174 (0.0)	0.1103	0.2316	0.2007
UT *base*, 6 steps (fixed)	131 (0.32)	279 (0.18)	319 (0.17)	**0.4801**	0.5422	0.5216
UT w/ dynamic halting	130 (0.32)	**134** (0.22)	**142** (0.19)	0.4603	**0.5831**	**0.5625**
UT *base*, 8 steps (fixed)	129(0.32)	192 (0.21)	202 (0.18)	-	-	-
UT *base*, 9 steps (fixed)	**129(0.33)**	214 (0.21)	239 (0.17)	-	-	-

- UT가 Transformer 보다 좋음
- Dynamic halting 평균 스텝 – 8.2
- Step 크기 8, 9로 고정했을 때 보다 Dynamic Halting이 좋음
- 단순히 연산을 많이 해서 좋은 것이 아님
- 중요치 않은 symbol은 halting 하고 중요한 symbol은 더 많은 computation 하는 것이 중요한 역할

(4/6) Experiments

Input	1	0	1	0	+	0	1	1	1
Output	1	1	0	0	1				

- Algorithmic Tasks
  - Copy, Reverse and Addition
  - Train at sequences of length 40
  - Evaluate at sequences of length 400

Model	Copy		Reverse		Addition	
	char-acc	seq-acc	char-acc	seq-acc	char-acc	seq-acc
LSTM	0.45	0.09	0.66	0.11	0.08	0.0
Transformer	0.53	0.03	0.13	0.06	0.07	0.0
Universal Transformer	0.91	0.35	0.96	0.46	0.34	0.02
Neural GPU*	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

(5/6) Experiments

Input:

```
j=8584
for x in range(8):
  j+=920
b=(1500+j)
print((b+7567))
```

Target:

```
25011
```

- Learning to Execute(LTE)
  - Memorization
    - Copy, double and reverse
  - Program Evaluation
    - Program, control and addition

	Copy		Double		Reverse	
Model	char-acc	seq-acc	char-acc	seq-acc	char-acc	seq-acc
LSTM	0.78	0.11	0.51	0.047	0.91	0.32
Transformer	0.98	0.63	0.94	0.55	0.81	0.26
Universal Transformer	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 5: Character-level (*char-acc*) and sequence-level accuracy (*seq-acc*) results on the Memorization LTE tasks, with maximum length of 55.

	Program		Control		Addition	
Model	char-acc	seq-acc	char-acc	seq-acc	char-acc	seq-acc
LSTM	0.53	0.12	0.68	0.21	0.83	0.11
Transformer	0.71	0.29	0.93	0.66	**1.0**	**1.0**
Universal Transformer	**0.89**	**0.63**	**1.0**	**1.0**	**1.0**	**1.0**

Table 6: Character-level (*char-acc*) and sequence-level accuracy (*seq-acc*) results on the Program Evaluation LTE tasks with maximum nesting of 2 and length of 5.

(6/6) Experiments

Model	BLEU
Universal Transformer *small*	26.8
Transformer *base* (Vaswani et al., 2017)	28.0
Weighted Transformer *base* (Ahmed et al., 2017)	28.4
Universal Transformer *base*	**28.9**

- WMT 2014 English-German translation task
  - Transition function – FC Layer
  - Without Dynamic Halting

Conclusion

- RNN의 재귀구조를 Transformer와 결합시켰더니 Algorithmic 문제와 Natural Language Understanding 문제에서 성능이 좋아졌다.