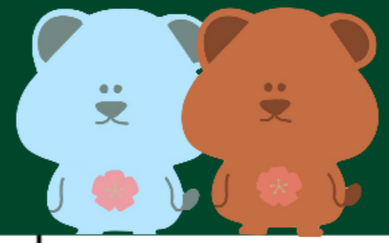


Comparative Study of CNN and RNN for Natural Language Processing

5기 차수빈

EWHA EURON

1. Introduction

EWHA EURON

1. Introduction

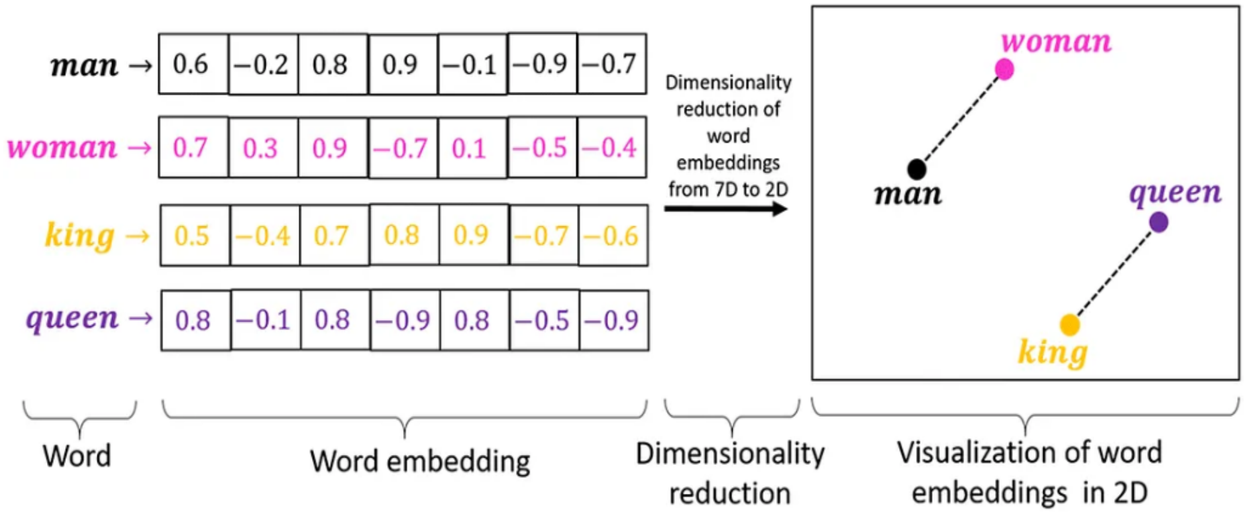
https://arxiv.org/abs/1708.02709

1. Embedding

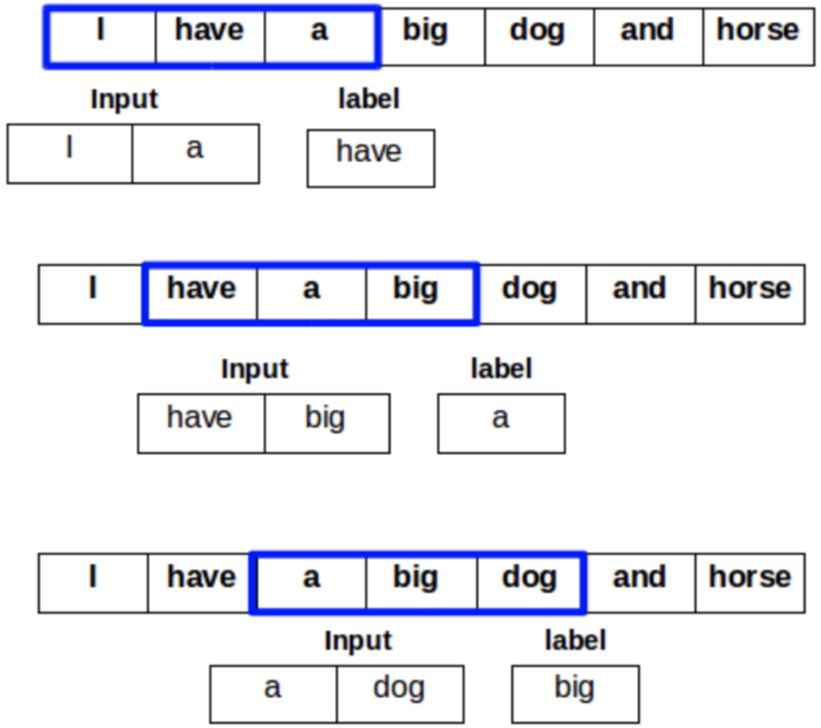
- 단어 혹은 문장이 가지고 있는 상관관계를 벡터의 형태로 그 상관관계가 보존될 수 있도록 변형시키는 작업

- NLP는 문장/단어에 섞여 있는 속뜻이나 단어들 사이의 상관관계를 추측하는 과정이 필수적!

- 임베딩 종류
  - 단어 임베딩: 벡터에 단어의 문맥적 의미를 함축
    Word2vec, GloVe, FastText 등
  - 문장 임베딩: 단어/문장의 순서 및 단어/문장 간 상호 정보를 고려



▲ Word Embedding



▲ 문장 Embedding

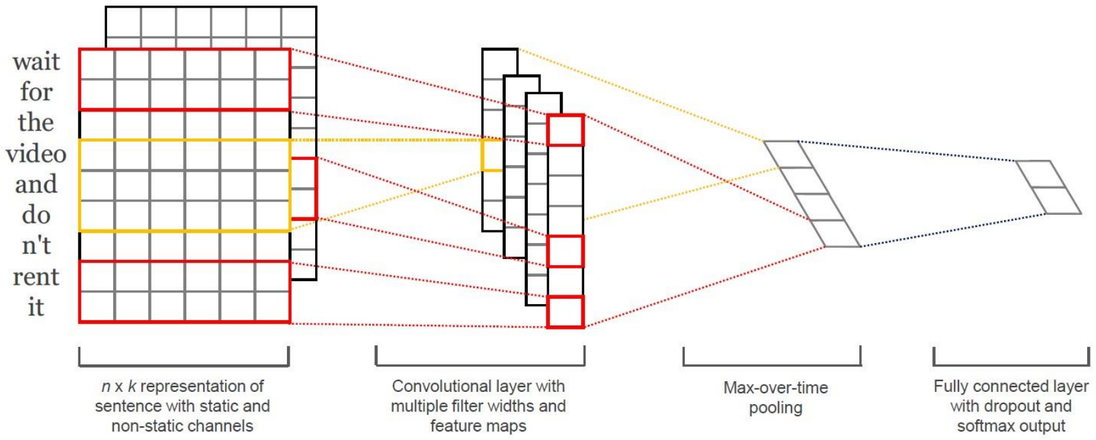
1. Introduction

2. CNN(=> 분류 작업)

- 임베딩을 통해 얻은 embedding vector matrix를 input으로 한 CNN 네트워크 생성

- 위치에 무관한 특징을 추출하는 데 유리

- 가장 높은 값을 가지는 value가 자연어의 추상성을 대표하는 값이 될 가능성이 높음
  -> Max-pooling 활용

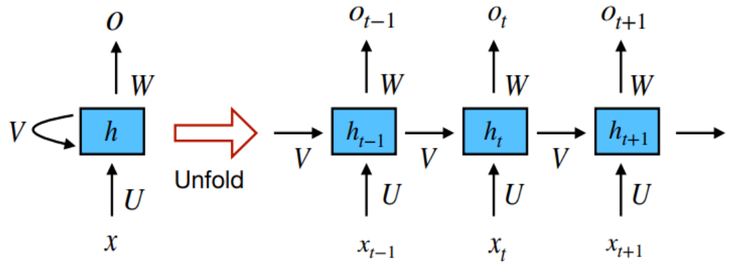


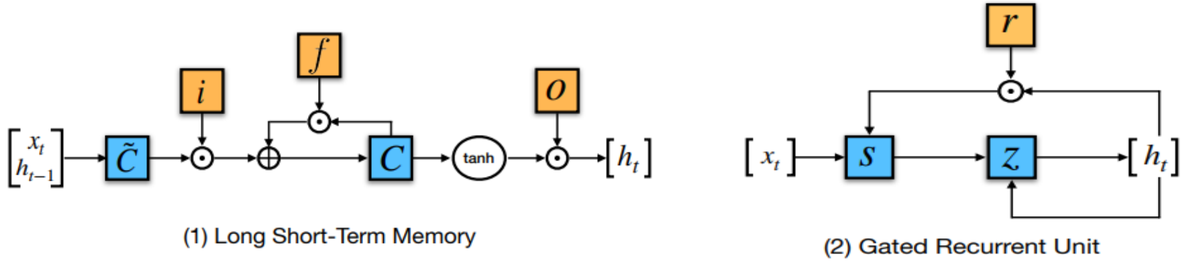
3. RNN(=> 시퀀스 모델링 작업)

- 연속적인 정보를 처리하는 데 유용한 딥러닝 구조
  크기가 정해지지 않은 출력을 얻는 데 적합

- 순서대로 들어오는 input에 따라 output을 만들고, 이전에 얻은 hidden state를 다음 RNN layer의 input으로 활용
  ➔네트워크가 이전에 들어온 문장의 숨은 특성에 대해 학습 가능

- 기본 RNN 구조에서 LSTM, GRU 등으로 발전





1. Introduction

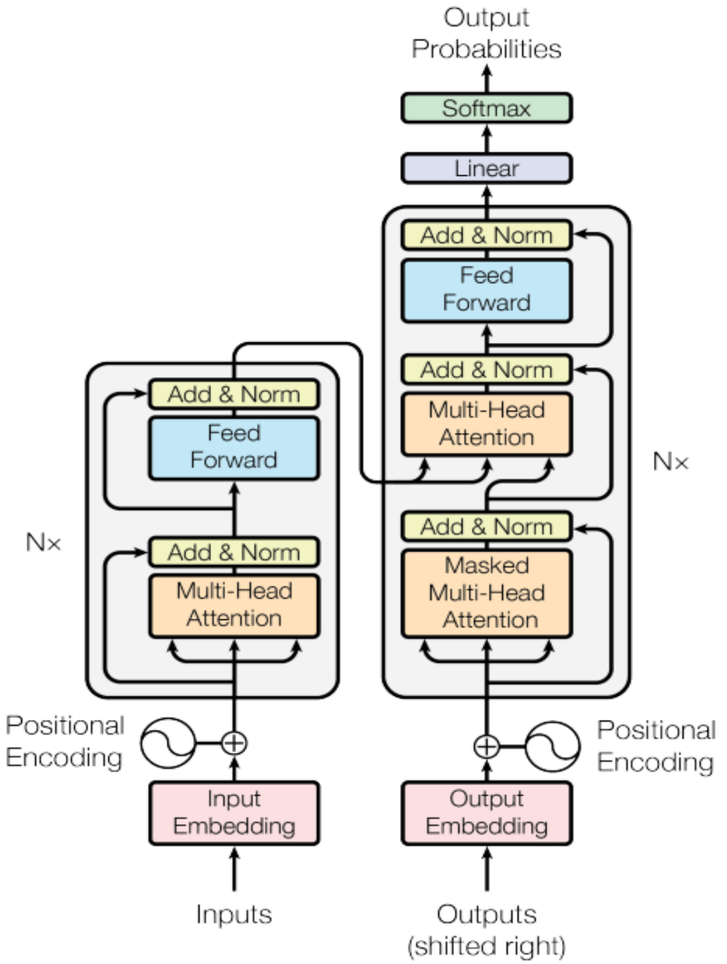
4. Attention(Transformer)

- RNN 기반 모델은 마지막 은닉상태(hidden state)만을 decoder에 전달
  -> 입력된 “모든” 단어의 정보가 decoder에 제대로 전달되지 못함
  -> 입력된 단어가 많을수록 앞쪽에서 입력된 단어는 거의 전달이 되지 못함

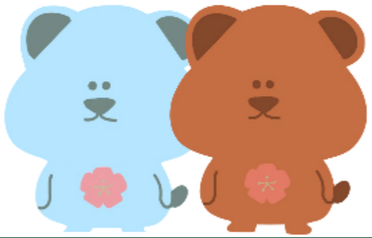
- Attention 기법은 각 단어에 대한 은닉상태 정보를 모두 decoder로 전달
  -> 예측하고자 하는 단어와 관련이 높은 단어에 더 많은 주의(attention)를 기울여 가중치를 부여

- 단어들 사이의 실제 연관성과 가중치를 파악할 수 있음

- BERT(Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), T5 등



2. 선행 연구



EWHA EURON

2. 선행 연구 – CNN vs RNN

1. Vu 등(2016)
- 관계 분류 작업에서 CNN과 기본 RNN을 조사
- CNN이 RNN보다 더 높은 성능을 보였음
- 또한 CNN과 RNN이 상호 보완적인 정보를 제공한다는 증거를 제시

2. Wen et al. (2016)와 Adel 및 Schütze (2017)
- 긴 문장의 분류 작업에서 CNN을 GRU/LSTM보다 선호

3. Yin et al. (2016)
- 답변 선택 작업에서 어텐션 기반 CNN이 어텐션 기반 LSTM보다 더 나은 성능

4. Dauphin et al. (2016)
- fine-tuned gated CNN은 긴 문맥 의존성을 모델링 할 수 있으며, 최첨단 성능을 달성한다고 주장함

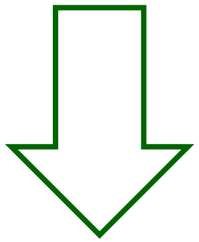
2. 선행 연구 – CNN vs RNN

5. Arkhipenko et al. (2016)

- 러시아 트윗의 감정 분석에서 GRU가 LSTM과 CNN을 능가한다는 것을 확인

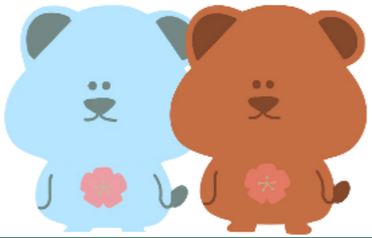
6. Chung et al. (2014)와 Jozefowicz et al. (2015)

- GRU와 LSTM 사이에 명확한 승자가 없다는 결과를 발견
- 많은 작업에서 기본 알고리즘은 비슷한 성능을 나타내며, 하이퍼 파라미터 튜닝이 중요한 역할을 한다고 지적



어떤 아키텍처가 어떤 NLP 작업에서 더 나은 성능을 보이는지 일관된 결론을 내리기 어려우며, 작업 및 데이터에 따라 다를 수 있음

3. Experiments



EWHA EURON

3-1. Tasks

- 여러 NLP 관련 작업들을 네 가지 범주로 구성함

1. TextC(텍스트 분류)
- 감정 분류(SentiC)
- 관계 분류(RC)

2. SemMatch(의미 일치)
- 텍스트 수반(TE)
- 정답 선택(AS)
- 질문-관계 매칭(QRM)

3. SeqOrder(시퀀스 순서)
- PQA

4. ContextDep(문맥 종속성)
- 품사 태깅

3-2. Setup

1. 변인 통제

- 항상 처음부터 훈련하며, pre-trained 단어 임베딩 등의 추가 지식을 사용하지 않음
- 항상 배치 정규화와 같은 복잡한 트릭 없이 기본 설정만을 사용하여 훈련
- 각 작업 및 각 모델에 대한 최적 하이퍼 파라미터를 개별적으로 탐색
  → 모든 결과가 최적 하이퍼 파라미터를 기반으로 함
- 각 모델의 기본 아키텍처와 활용을 조사

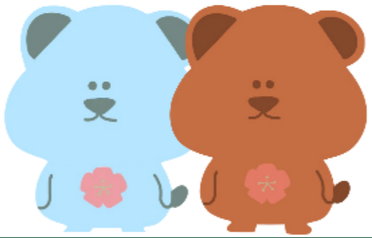
2. 구조

- CNN: 합성곱 레이어 + max-pooling 레이어로 구성
- RNN(GRU 및 LSTM)
  - 입력을 왼쪽에서 오른쪽으로 모델링하고 항상 입력의 마지막 숨겨진 상태를 최종 표현으로 사용
    단 POS 태깅의 경우, 양방향 RNN도 보고
    -> 이렇게 함으로써 각 단어의 표현이 양쪽 모두의 문맥을 인코딩할 수 있게 함

3. 하이퍼 파라미터 튜닝

- 검증 데이터에서 조정
- hidden size, minibatch size, learning rate, 최대 문장 길이, 필터 크기(CNN only), 그리고 AS, QRM 및 PQA 작업에서 랭킹 손실 마진 등을 포함

4. 결과 & 분석



EWHA EURON

4-1. 결과

			performance	lr	hidden	batch	sentLen	filter_size	margin
TextC 텍스트 분류	SentiC (acc)	CNN	82.38	0.2	20	5	60	3	–
		GRU	**86.32**	0.1	30	50	60	–	–
		LSTM	84.51	0.2	20	40	60	–	–
	RC (F1)	CNN	68.02	0.12	70	10	20	3	–
		GRU	**68.56**	0.12	80	100	20	–	–
		LSTM	66.45	0.1	80	20	20	–	–
SemMatch 의미 일치	TE (acc)	CNN	77.13	0.1	70	50	50	3	–
		GRU	**78.78**	0.1	50	80	65	–	–
		LSTM	77.85	0.1	80	50	50	–	–
	AS (MAP & MRR)	CNN	**(63.69,65.01)**	0.01	30	60	40	3	0.3
		GRU	(62.58,63.59)	0.1	80	150	40	–	0.3
		LSTM	(62.00,63.26)	0.1	60	150	45	–	0.1
	QRM (acc)	CNN	**71.50**	0.125	400	50	17	5	0.01
		GRU	69.80	1.0	400	50	17	-	0.01
		LSTM	71.44	1.0	200	50	17	-	0.01
SeqOrder 시퀀스 순서	PQA (hit@10)	CNN	54.42	0.01	250	50	5	3	0.4
		GRU	**55.67**	0.1	250	50	5	–	0.3
		LSTM	55.39	0.1	300	50	5	–	0.3
ContextDep 문맥 종속성	POS tagging (acc)	CNN	94.18	0.1	100	10	60	5	–
		GRU	93.15	0.1	50	50	60	–	–
		LSTM	93.18	0.1	200	70	60	–	–
		Bi-GRU	94.26	0.1	50	50	60	–	–
		Bi-LSTM	**94.35**	0.1	150	5	60	–	–

◀ Hyper Parameters

Table 1: Best results or CNN, GRU and LSTM in NLP tasks

4-1. 결과

			performance	lr	hidden	batch	sentLen	filter_size	margin
SeqOrder 시퀀스 순서	PQA (hit@10)	CNN	54.42	0.01	250	50	5	3	0.4
		GRU	**55.67**	0.1	250	50	5	–	0.3
		LSTM	55.39	0.1	300	50	5	–	0.3
ContextDep 문맥 종속성	POS tagging (acc)	CNN	94.18	0.1	100	10	60	5	–
		GRU	93.15	0.1	50	50	60	–	–
		LSTM	93.18	0.1	200	70	60	–	–
		Bi-GRU	94.26	0.1	50	50	60	–	–
		Bi-LSTM	**94.35**	0.1	150	5	60	–	–

- RNN은 순서 정보와 장거리 문맥 종속성 인코딩에 유리 ⇒ 예상대로 더 좋은 성능!

4-1. 결과

			performance	lr	hidden	batch	sentLen	filter_size	margin
TextC 텍스트 분류	SentiC (acc)	CNN	82.38	0.2	20	5	60	3	–
		GRU	**86.32**	0.1	30	50	60	–	–
		LSTM	84.51	0.2	20	40	60	–	–
	RC (F1)	CNN	68.02	0.12	70	10	20	3	–
		GRU	**68.56**	0.12	80	100	20	–	–
		LSTM	66.45	0.1	80	20	20	–	–
SemMatch 의미 일치	TE (acc)	CNN	77.13	0.1	70	50	50	3	–
		GRU	**78.78**	0.1	50	80	65	–	–
		LSTM	77.85	0.1	80	50	50	–	–
	AS (MAP & MRR)	CNN	(**63.69,65.01**)	0.01	30	60	40	3	0.3
		GRU	(62.58,63.59)	0.1	80	150	40	–	0.3
		LSTM	(62.00,63.26)	0.1	60	150	45	–	0.1
	QRM (acc)	CNN	**71.50**	0.125	400	50	17	5	0.01
		GRU	69.80	1.0	400	50	17	-	0.01
		LSTM	71.44	1.0	200	50	17	-	0.01

- CNN은 로컬 및 위치 불변 특징을 추출하는 데 유리 ⇒ 예상과 다른 결과!

4-2. 분석

- **정량 분석**

- 다양한 NLP 작업에서 GRU와 CNN의 성능 비교
- 각 작업의 복잡성/ 요구사항에 따라 차이
  - GRU: 전체 문장 또는 장거리 의미 종속성이 중요한 작업
  - CNN: local 및 위치 불변 특징 추출

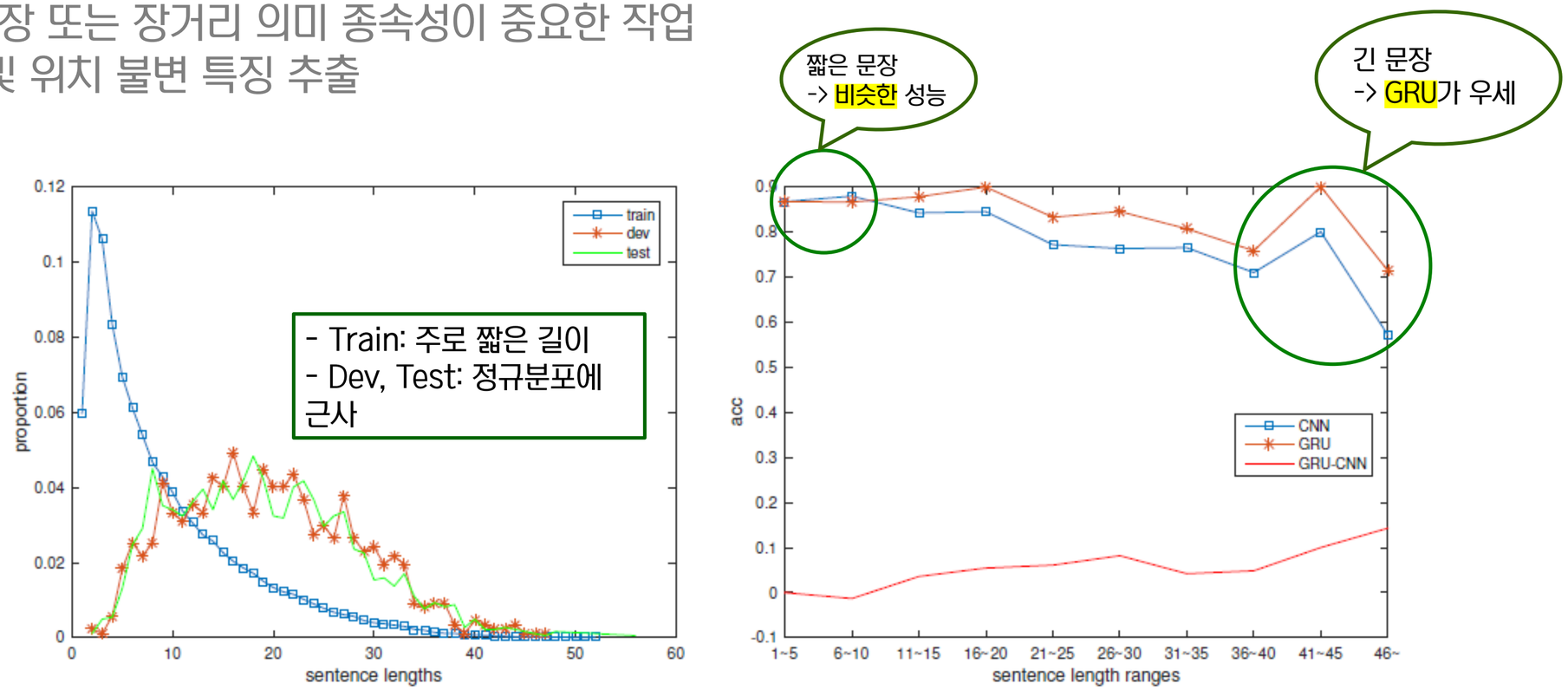


Figure 2: Distributions of sentence lengths (left) and accuracies of different length ranges (right).

4-2. 분석

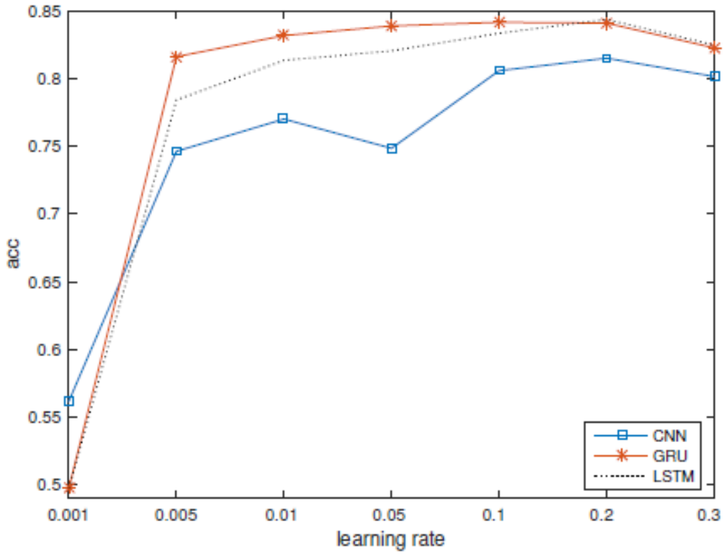
- **하이퍼 파라미터에 대한 민감도**

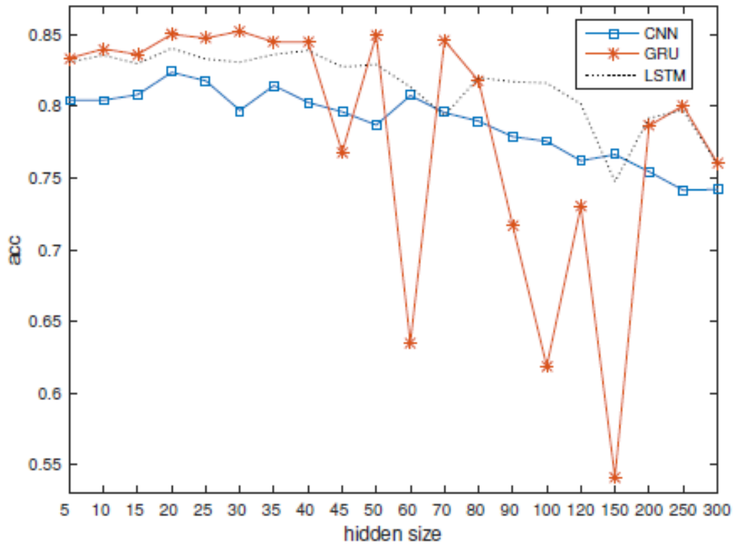
- 하이퍼 파라미터 값이 변할 때 CNN과 GRU의 성능이 얼마나 안정적인지를 확인

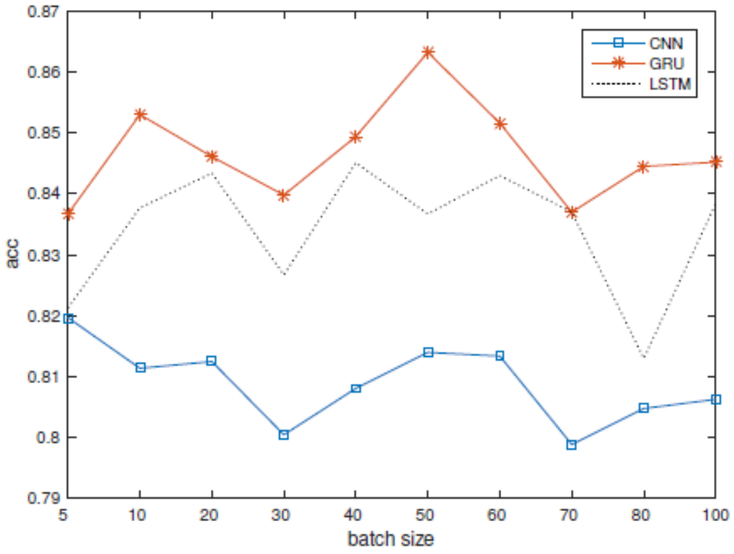
- 각각 다른 학습률, hidden size, 은닉 크기, 배치 크기에 대해 성능 비교

learning rate: 비교적 안정적

hidden size, batch size: 변동이 심함





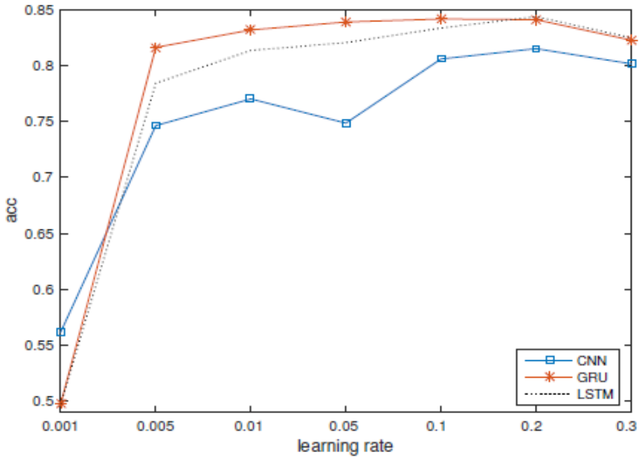


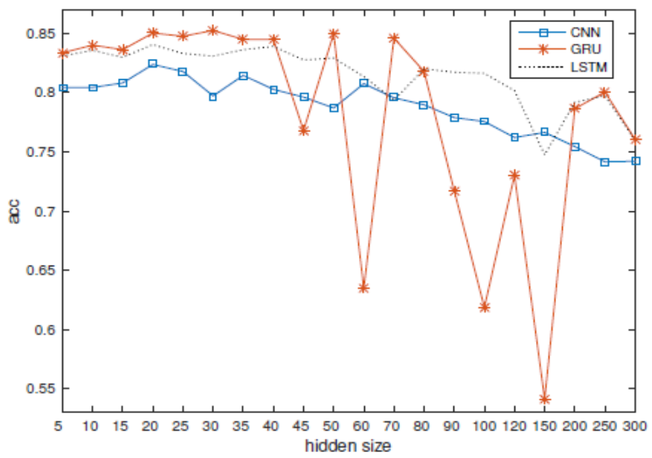
4-2. 분석

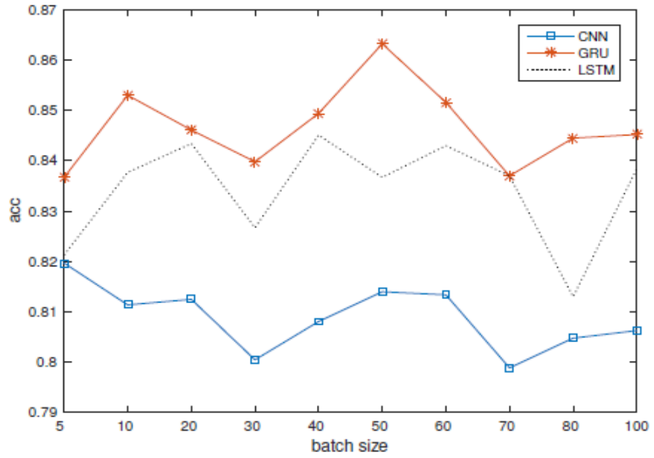
- **하이퍼 파라미터에 대한 민감도**

- 여전히 작업에 따른 성능 차이 존재

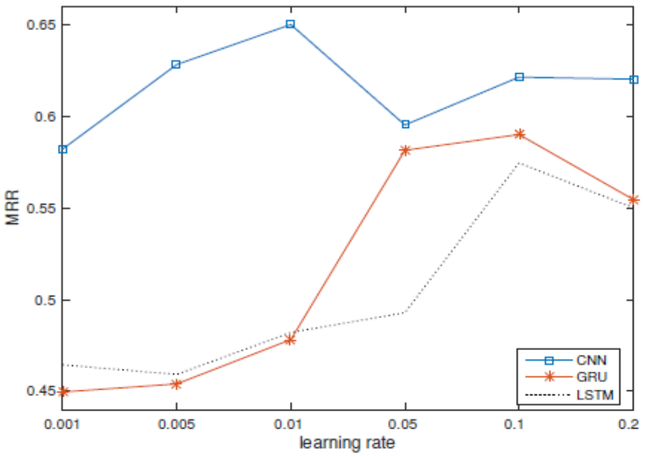
Sentic

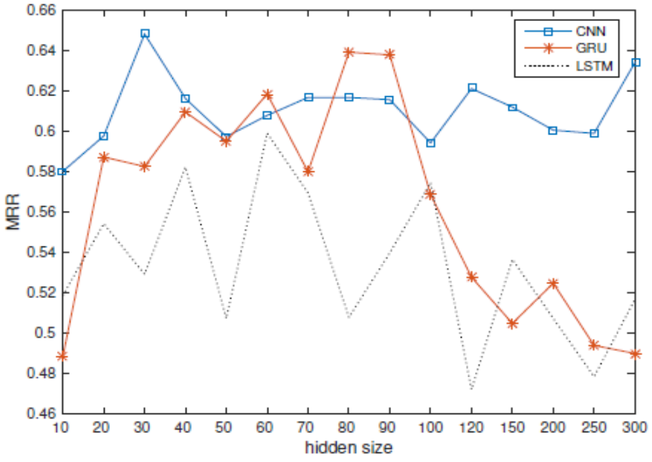


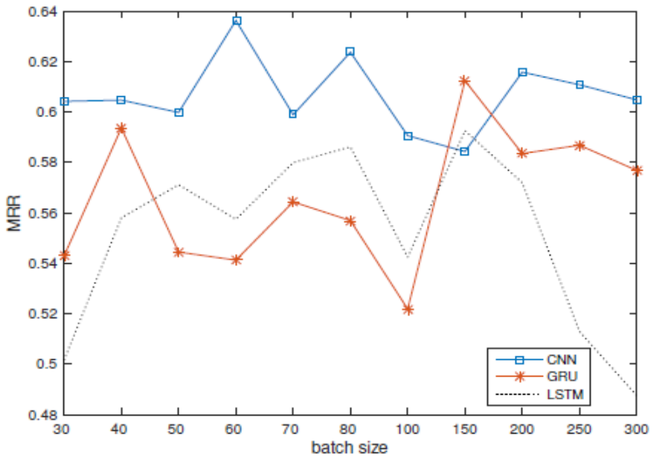




AS







5. Conclusion

EWHA EURON

5. Conclusion

- NLP 작업에서 가장 널리 사용되는 세 가지 DNN(**CNN**, **GRU** 및 **LSTM**)을 비교

- **RNN**은 다양한 작업에서 잘 수행되며 견고한 결과를 보임
  - 그러나 작업이 본질적으로 몇 가지 주요 어구(key phrase) 인식 작업인 경우(≒ 감정 감지 및 질문-답변 일치)에는 성능이 떨어질 수 있다는 것을 발견

- hidden size와 batch size는 DNN 성능을 크게 변동시킬 수 있음
  - 두 매개변수의 최적화가 CNN 및 RNN의 좋은 성능에 중요하다는 것을 시사

+ Discussion

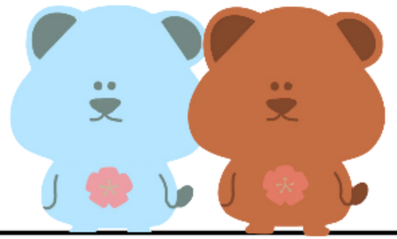
EWHA EURON

+ Discussion

- NLP에서 CNN은 local 및 위치 불변 특징 추출에, RNN은 연속적인 정보를 처리하는 데 유용하다는 특징을 가지고 있습니다. 이후 등장한 Transformer는 CNN과 RNN의 특징을 모두 반영하고 있는 것 같습니다.

Q> 논문에서 제시된 4가지 작업(텍스트 분류, 의미 일치, 시퀀스 순서, 문맥 종속성)에 대해 Transformer로도 실험을 진행한다면, Transformer가 모든 작업에서 CNN이나 RNN보다 우수한 성능을 보일까요?

THANK YOU



EWHA EURON