LSTM 언어모델 기반 한국어 문장 생성

김양훈 외 3명(서울대), LSTM Language Model Based Korean Sentence Generation, 한국통신학회 논문지, 2016.05

KAIG 세미나 (190218)

논문 구성

- 논문 제목(한글, 영문), 논문 발표자 정보
- 요약 (ABSTRACT)
- 서론: 연구 소개, 필요성 및 논문 구성 소개
- 관련 연구: 유사 연구들 소개 및 부족한 점 서술
- Recurrent Neural Networks
  - Traditional Recurrent Neural Networks
    - Forward Propagation
    - Backward Propagation Through Time – BPTT
    - Vanishing Gradient Problem
  - LSTM
- 시스템 모델
- 실험
- 결론
- 참고문헌 (References)

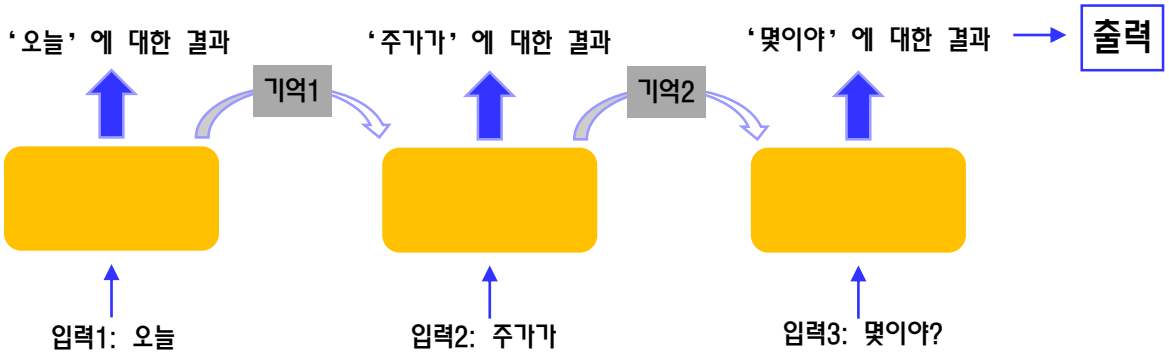
요약

- 순환신경망은 순차적이거나 길이가 가변적인 데이터에 적합한 딥러닝 모델
- LSTM은 순환신경망에서 나타나는 기울기 소멸문제를 해결함으로써 시퀀스 구성 요소간의 장기의존성을 유지
- 본 논문은 LSTM에 기반한 언어모델을 구성
  - 불완전한 한국어 문장이 입력으로 주어졌을 때 뒤 이어 나올 단어들을 예측하여 완전한 문장을 생성할 수 있는 방법을 제안
- 제안된 방법을 평가
  - 여러 한국어 말뭉치를 이용하여 모델을 학습
  - 한국어 문장의 불완전한 부분을 생성하는 실험을 진행
  - 실험 결과, 제시된 언어모델이 자연스러운 한국어 문장을 생성해 낼 수 있음을 확인
  - 문장 최소 단위를 어절로 설정한 모델이 다른 모델보다 문장 생성에서 더 우수한 결과를 보임.

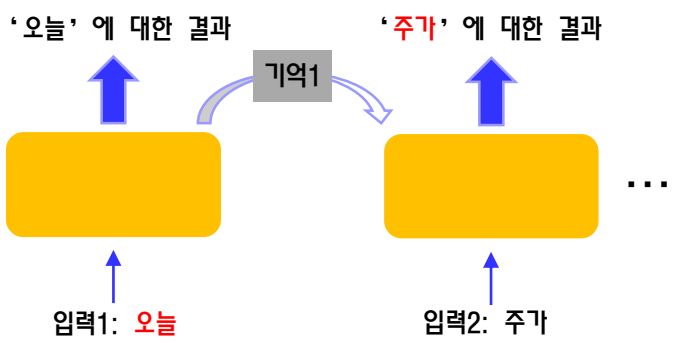
Recurrent Neural Network

- 여러 데이터가 순서대로 입력되었을 때 앞서 입력 받은 데이터를 잠시 기억
- 기억된 데이터가 얼마나 중요한지를 판단하여 별도의 가중치를 줘서 다음 데이터로 넘어감
- 모든 입력 값에 이 작업을 순서대로 실행, 다음 층으로 넘어가기 전에 같은 층을 맴도는 것 처럼 보임 – 같은 층에서 맴도는 성질 – 순환 신경망

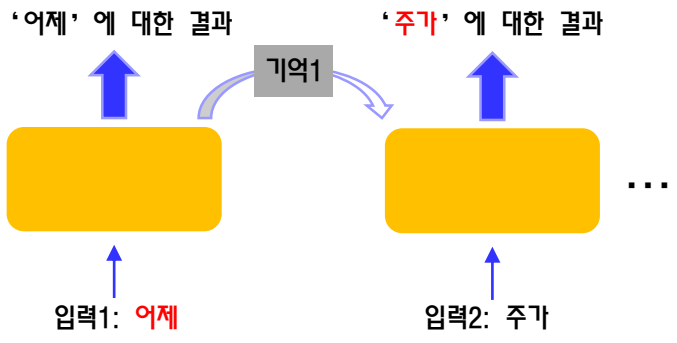
- 인공지능 비서에게, "오늘 주가가 몇이야?"



- 앞서 나온 입력에 대한 결과가 뒤에 나오는 입력 값에 영향을 줌
- 비슷한 문장이 입력되었을 때 그 차이를 구별하여 출력 값에 반영
- 입력 2의 값은 양쪽 모두 '주가' 이지만, 왼쪽의 주가는 오늘을 기준으로, 오른쪽은 어제를 기준으로 계산되어야 함



[오늘 주가를 묻는 경우]



[어제 주가를 묻는 경우]

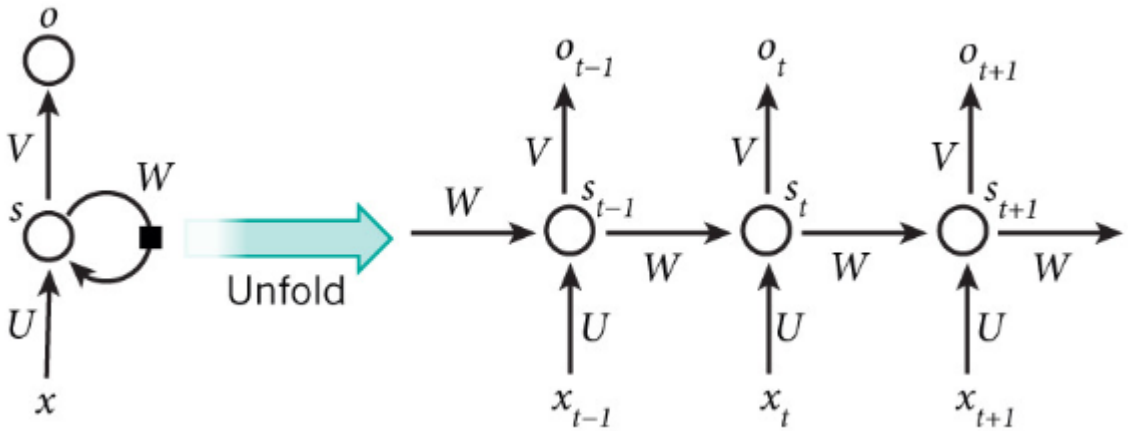
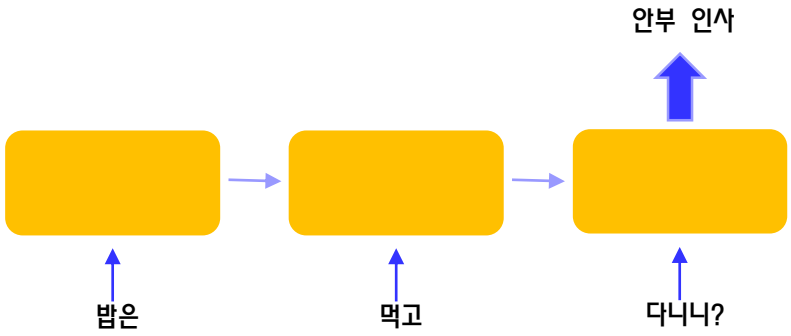


Fig. RNN

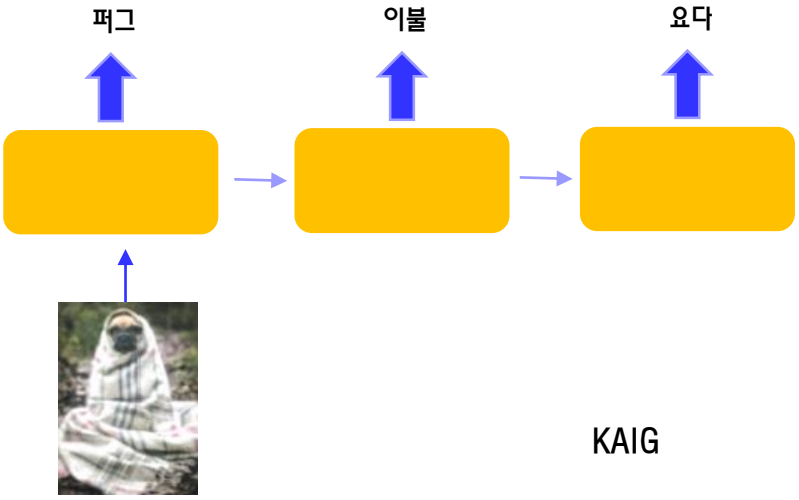
- x_t는 시간 스텝(time step) t에서의 입력값이다.
- s_t는 시간 스텝 t에서의 hidden state이다. 네트워크의 "메모리" 부분으로서, 이전 시간 스텝의 hiddent state 값과 현재 시간 스텝의 입력값에 의해 계산된다: $s_t = f(Ux_t + Ws_t - 1)$. 비선형 함수 f는 보통 tanh나 ReLU가 사용되고, 첫 hidden state를 계산하기 위한 s_1은 보통 0으로 초기화시킨다. $f(Ux_t, Vs_{t-1})$와 같은 표현
- o_t는 시간 스텝 t에서의 출력값이다. 예를 들어, 문장에서 다음 단어를 추측하고 싶다면 단어 수만큼의 차원의 확률 벡터가 될 것이다. $o_t = softmax(Vs_t)$

적용 예

- 입력, 출력 값을 어떻게 설정하느냐에 따라 여러 가지 상황에서 적용 가능
  - 다수 입력 단일 출력: 문장을 읽고 뜻을 파악할 때 활용 (그외 감성 분석 등등)

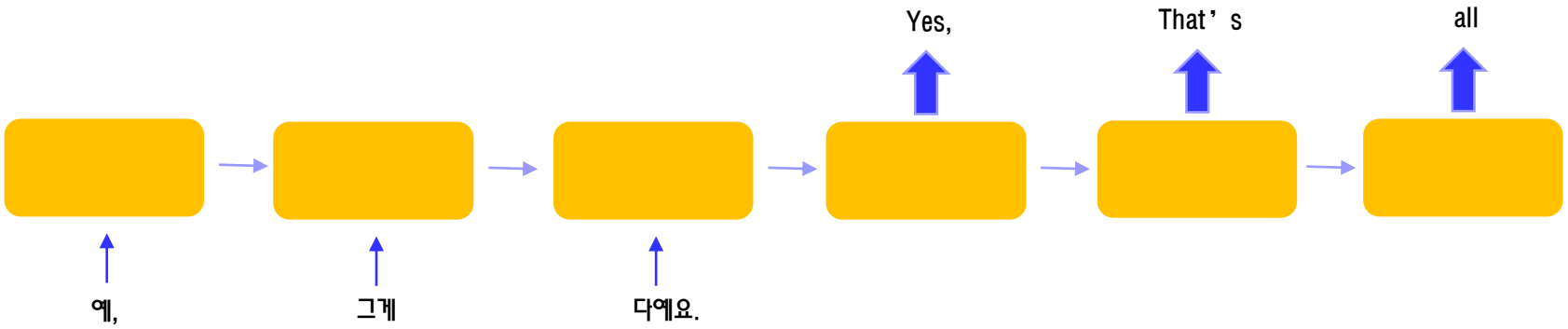


  - 단일 입력 다수 출력: 사진의 캡션을 만들 때 활용 (CNN + RNN)



KAIG

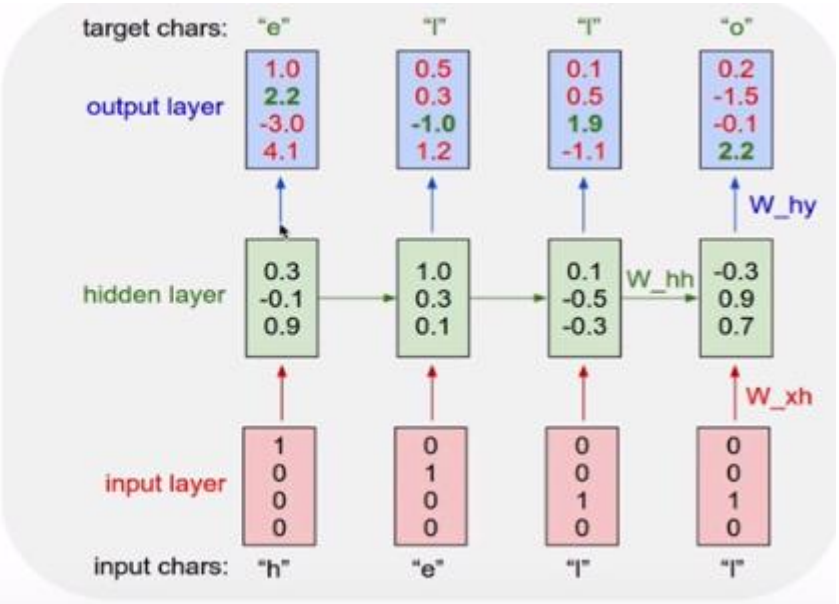
▫ 다수 입력 다수 출력: 문장을 번역할 때 활용



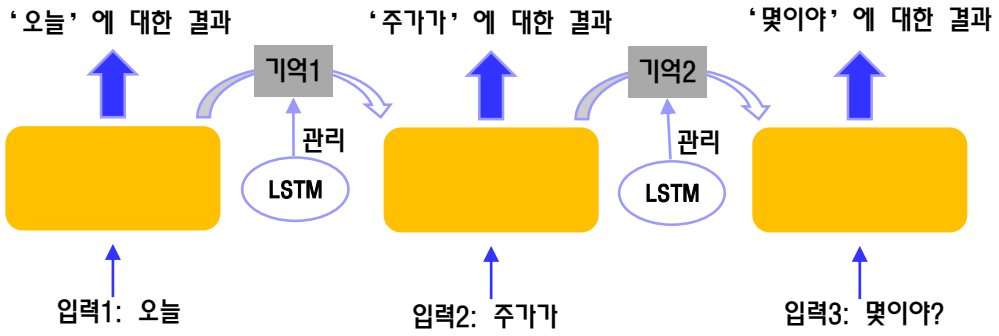
Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
"hello"



- Long Short Term Memory
  - 한 층 안에서 반복을 많이 해야 하는 RNN
  - 기울기 소실 문제가 많이 발생
  - 반복되기 전에 다음 층으로 기억된 값을 넘길지 안 넘길지를 관리하는 단계 추가



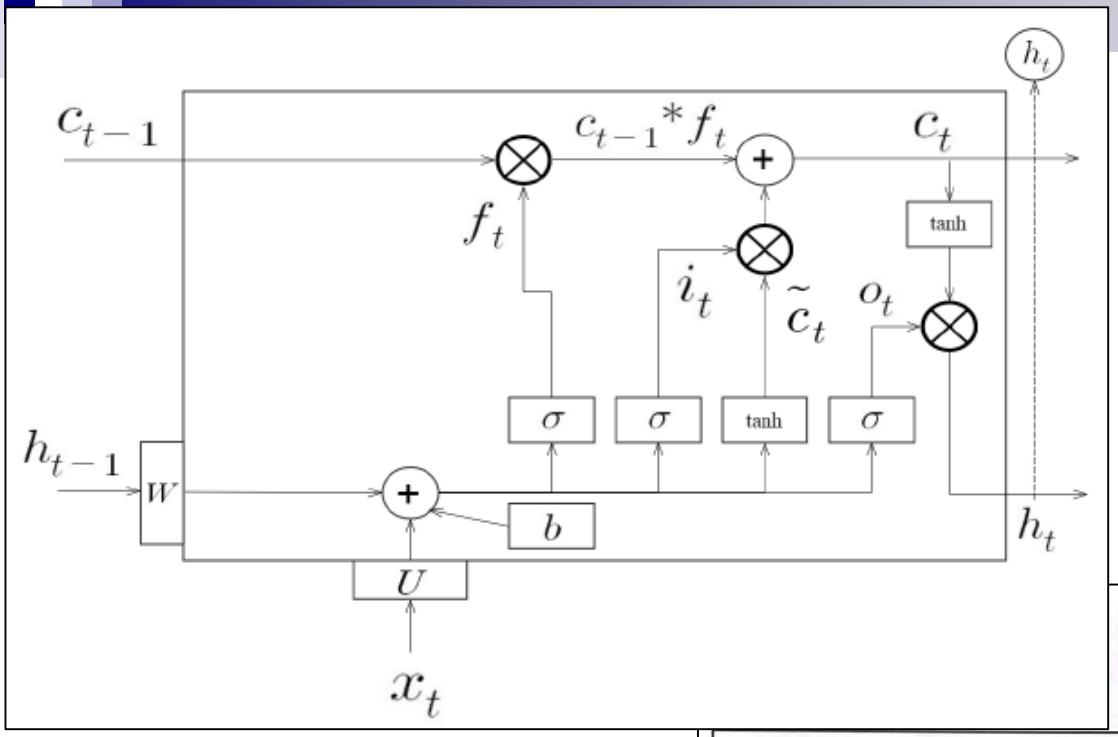


Fig. LSTM

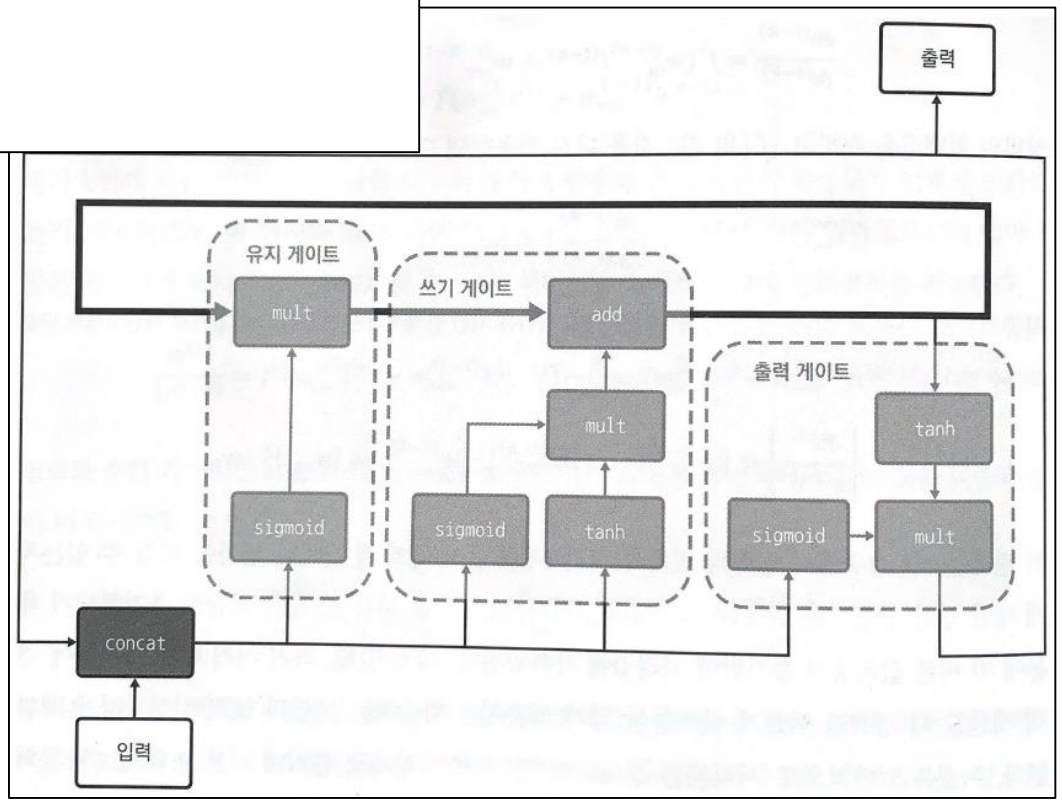
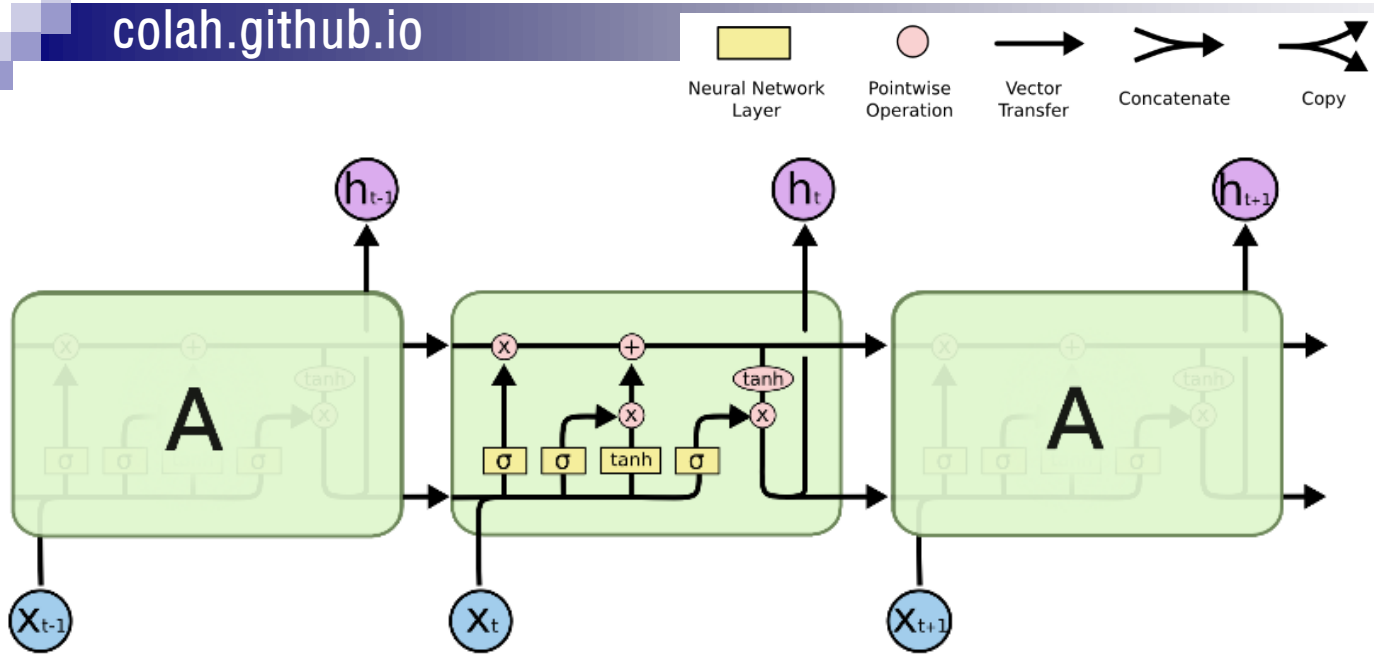
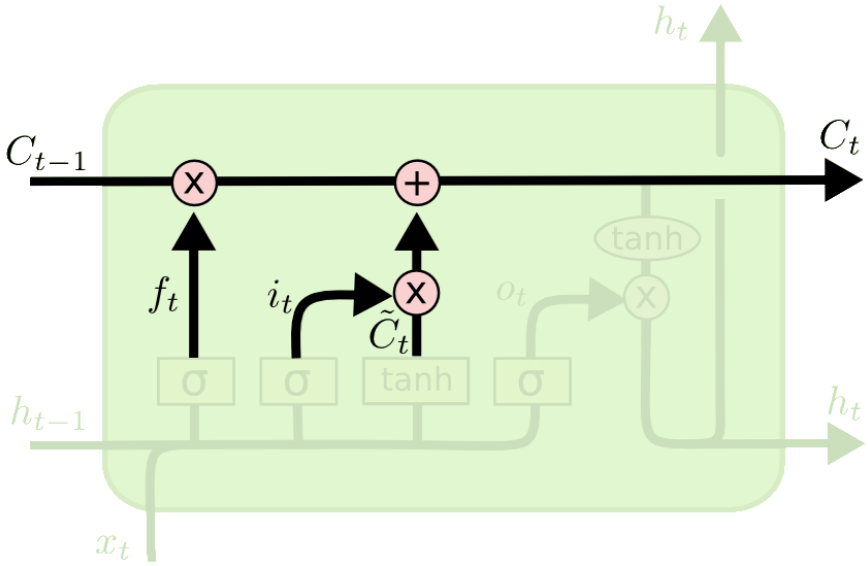


Fig. LSTM



(*Cell* : 기억/망각 정도를 학습)



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$ 망각 정도

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$ 기억 정도

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$ 현재 셀 상태

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

KAIG

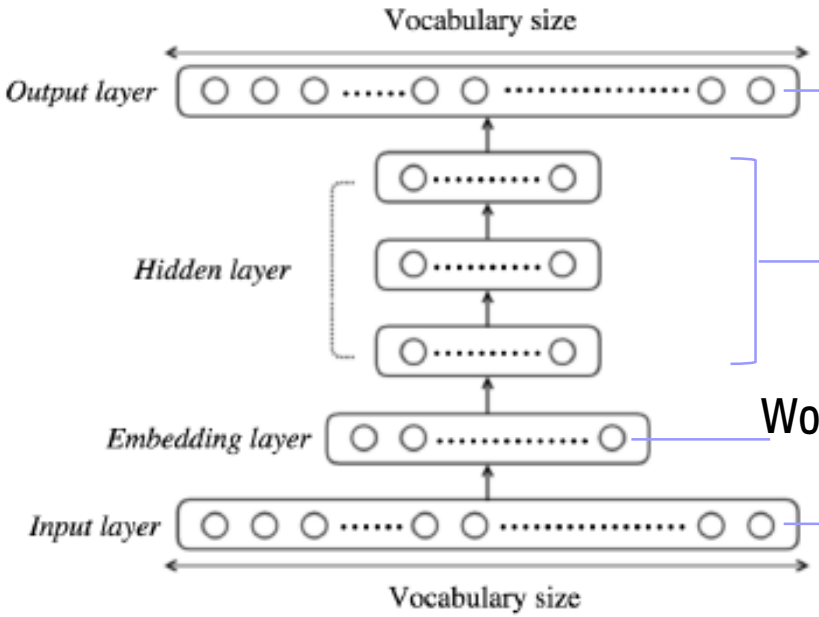
시스템 모델

- 본 논문에서는 입력 값으로 불완전한 한국어 문장이 단어 단위로 주어졌을 때, LSTM을 통하여 뒤이어 나올 단어들을 예측하는 문제를 해결
- 입력 값 $w=(w_1, \ldots, w_T)$가 앞서 입력된 단어 배열을 통해 완전한 문장이 구성되기까지의 문맥(Context)를 학습
- 출력계층(Output Layer)에서 단어사전(Dictionary)의 모든 단어에 대해 매 단어가 불완전한 한국어 문장을 뒤 이을 단어 w_{T+1} 로 나올 확률 값 출력

$$
\begin{aligned}
&P(x_{t+1}, x_t, \cdots, x_2, x_1) \\
&= P(x_{t+1}|x_1, x_2, \cdots, x_{t-1}, x_t) \times P(x_1, x_2, \cdots, x_{t-1}, x_t) \\
&= P(x_{t+1}|x_1, x_2, \cdots, x_{t-1}, x_t) \\
&\times P(x_t|x_1, x_2, \cdots, x_{t-2}, x_{t-1}) \times P(x_1, x_2, \cdots, x_{t-2}, x_{t-1})
\end{aligned}
$$

- 즉, 계산된 확률값 중 가장 높은 값을 가진 단어를 w_{T+1} 로 출력

사용된 LSTM 구조



사전에 있는 모든 단어들에 대한 확률값 출력

각각이 LSTM 구조, 게이트 연산은 Peephole Connection

Word Embedding (최근 Word2Vec 기법 활용)

One-Hot Encoding

LSTM 기반 언어모델을 이용한 단어 생성 예제

$Input$ = "나는 지금 집에"

$Preprocessing$: "나", "는", "지금", "집", "에" ——— **형태소 분석**

w_0 = "나", w_1 = "는", w_2 = "지금", w_3 = "집", w_4 = "에"

$\Rightarrow At\ t = 5,\quad P(w_5|$"나", "는", "지금", "집", "에"$)$

$o_4 = \mathrm{argmax}\, P(w_5|$"나", "는", "지금", "집", "에"$)$ = "간다"

$Input$ = "나는 지금 집에 간다"

$\Rightarrow At\ t = 6,\quad P(w_6|$"나", "는", "지금", "집", "에", "간다"$)$

$o_5 = \mathrm{argmax}\, P(w_6|$"나", "는", "지금", "집", "에", "간다"$)$ = "$< eos >$"

$If(o_t \equiv$ "$< eos >$"$)$: $print$ "나는 지금 집에 간다" ——— (End of Sentence) **까지 반복**

실 험

- 실험 세팅 - 데이터: 성경 텍스트와 뉴스 텍스트
  - Korean Bible Society, *Bible*, Retrieved 7th, Dec, 2015, http://www.bskorea.or.kr
  - jungyeul, *korean-parallel-corpora*(2014), Retrieved 22th, Oct, 2015, https://github.com/jungyeul/korean-parallel-corpora/tree/master/korean-english-v1

Data	Bible			News		
Sentences	22964			97123		
Preprocess	word	morpheme	character	word	morpheme	character
Vocabulary	28132	17129	1246	45085	19857	2203
Embedding Size	500/1000			1000		
Hidden Size	100					
Hidden stack	3					
mini-Batch	400					
Max sentence length	30	50	100	30	50	100

- 실험 세팅
  - 어절, 형태소, 음절 단위로 분할하는 전처리 과정 거쳐 각각 최대 50,000, 20,000, 3000개의 최대 어휘 개수를 가지도록 설정
  - 형태소 분석기: Twitter, *twitter-korean-text*, Retrieved 10th, Nov. 2015, https://github.com/twitter/twitter-korean-text
  - 출현빈도가 1인 문장 최소 단위 혹은 최대 어휘사이즈를 벗어난 문장들은 “**unknown**” 으로 처리하여 제외
  - Python 언어 및 Theano 라이브러리 이용
  - Intel Xeon CPU 2개(32 Cores), NVIDIA GTX TITAN X GPU 1개가 장착된 PC에서 진행

실험 결과 – 학습 시간

- 데이터 당 최대 60 epoch를 mini-batch를 이용하여 학습

표 2. 학습 시간
Table 2. Training Time

Data	Preprocess	Training time(hours)
Bible	word	7.5
	morpheme	6.1
	character	2.0
News	word	21.6
	morpheme	17.1
	character	5.4

- 학습 시간은 사전의 크기와 문장 당 학습하는 문장 최소 단위의 개수에 따라 좌우됨. 두 가지 변수 모두 커질수록 학습에 필요한 연산량이 증가
  - 단어를 형태소 및 음절로 분할할 경우 사전 크기는 감소하지만 문장 당 문장 최소 단위의 개수는 증가
  - 형태소 단위의 모델이 어절 단위의 모델보다 학습 시간이 약 10시간 정도 추가 소요, 이는 감소한 사전 크기의 영향보다 증가한 문장 당 평균 형태소 개수가 문장 당 평균 어절 개수보다 많은 것이 더 큰 영향을 준 것으로 풀이된다.

■ 실험 결과 – 최대 확률 문장 생성 결과: 입력된 문장은 말뭉치의 주제를 고려하여 랜덤 입력

표 3. 뉴스 데이터로 학습된 모델의 문장 생성 결과
Table 3. Sentence generation result using News data

Preprocess	Output Sentence
word	"정부는 전에도 이 같은 계획을 갖고 있는 것으로 알려졌다."
morpheme	"정부는 전에도 미국이 이라크에서 열린 우리당이 북한이 핵 프로그램을 위해 미국이 핵 프로그램을 비난했다."
character	"정부는 전에도 각각 8명의 상태가 목을 창출할 수 있는 서비스를 공개했다."

표 4. 성경 데이터로 학습된 모델의 문장 생성 결과
Table 4. Sentence generation result using Bible data

Preprocess	Output Sentence
word	"이는 혈통으로나 육정으로나 내가 나를 위하여 그 모든 말을 인하여 내 모든 말을 인하여 ..."
morpheme	"이는 혈통으로나 육정으로나 의 하나님께로 행하는것이 아니요 오직 하나님의 말씀을 알지 못함이니라. "
character	"이는 혈통으로나 육정으로나 사람이 그 아들 이스마엘과 아비야를 낳았고 그 아들 요나단이 그 아들 이스마엘과 아비야를 낳았고"

■ 전반적으로 뉴스 데이터의 학습 결과가 성경 데이터보다 자연스러운 문장을 생성

- 뉴스 데이터의 경우 성경 데이터의 약 4배 정도의 크기로 모델의 학습 결과에 큰 영향을 미친 것으로 생각
- 음절 단위보다는 형태소 단위가, 형태소 단위 보다는 어절 단위를 채택한 모델이 뛰어난 성능, 문장 최소 단위 및 최대 어휘 사이즈의 영향을 받은 것

- 실험 결과 – N-gram, vanilla RNN 과의 비교
  - 최대 50,000개의 사전 크기, 어절 단위 뉴스 데이터를 사용
  - N-Gram: NLTK 라이브러리로 구현
  - vanilla RNN: layer 수, 임베딩 크기 등의 모델 설정을 LSTM 기반 언어모델과 동일하게 설정
  - Perplexity(혼잡도): 테스트 셋 400 문장에 대해 측정한 perplexity의 평균값
  - 정량적(Perplexity)비교
    - LSTM 언어모델이 주어진 테스트 문장에서 나타나는 단어의 확률 분포와 가장 유사, 즉 성능이 가장 뛰어남
  - 정성적 비교
    - LSTM 기반 언어모델이 보다 자연스러운 의미의 문장 생성

Model	Perplexity	Sample sentences
N-gram	61	미국 정부는 오늘 아침 하락세로 개장했습니다.
		로스엔젤레스 남부의 한 작은 도시에서 총격전이 발생했다는 신고를 받고 출동한 경찰에 의해 수배 중이었다.
		우리는 그들의 도움이 필요하기 때문에 바로 착수할 수 있다고 말했다.
Vanilla RNN	64	미국 정부는 오늘 아침 처음으로 발표한 성명을 통해 이번 사건이 발생했다.
		로스엔젤레스 남부의 한 명의 미국 정부가 이끄는 기자회견에서 이번 사건에 대해 미국의 입장을 더 많은 조사를 하지 않을 것이라고 밝혔다
		우리는 그들의 경제 관련 문제를 해결할 수 있는 질문에 말했다.
LSTM	46	미국 정부는 오늘 초 오전 기자회견에서 이번 사건에 대해 더 많은 것을 발표했다.
		로스엔젤레스 남부의 한 종교단체 본부를 급습, 다른 세 명의 죽음에 이르게 한 20명의 여성을 체포했다.
		우리는 그들의 이런 모습을 본 적이 없다고 말했다.

- 실험 결과 - Embedding layer의 영향 분석
  - 언어모델은 우선적으로 임베딩 계층을 통해 각각의 단어를 특정 방식을 이용하여 단어벡터로 재표현 (Word Embedding)
    - 이는 사전의 크기가 달라지게 되고,
    - 재표현된 벡터들은 해당 단어의 함축적인 의미를 내포하기 때문에

    언어모델의 최종 학습 결과에 영향을 미치게 된다
  - 아래는 어절, 형태소, 음절을 기본 사전 단위로 사용한 성경데이터를 각각 1000-Embedding size, 500-Embedding size로 맵핑 시켰을 때, 언어모델의 성능 분석 결과
    - 어절, 형태소, 음절 순서로 사전 크기가 작아짐

표 7. 임베딩 크기에 따른 perplexity 비교
Table 7. perplexity comparison between different embedding size

Perplexity	500 embedding	1000 embedding
어절	100	79
형태소	122	112
음절	111	141

- 실험 결과 - Embedding layer의 영향 분석 (계속)
  - 성능 분석 결과 (계속)
    - 사전 크기가 비교적 큰 어절 단위 언어모델의 경우 임베딩을 적절하게 크게 선택하면 더 좋은 결과를 얻을 수 있으며,
    - 반대로 음절 단위 언어모델의 경우 임베딩을 상대적으로 작게 선택했을 때의 perplexity 값이 작은 값을 기록
    - 위 실험 결과를 통해 사전 크기에 적합한 임베딩 크기 선정이 언어 모델의 성능에 중요한 역할을 하는 것을 확인.

결론

- 본 논문에서는 LSTM 기반 언어모델을 통하여 문장의 일부분이 주어졌을 때 문장의 나머지 부분을 생성하여 문장을 완성시키는 시스템을 제안
- 제시된 LSTM 기반 언어모델이 각 말뭉치의 주제, 말뭉치의 크기, 학습 문장 최소 단위가 모델 학습 성능 및 문장 생성 결과에 어떠한 영향을 미치는지 살펴보았다.
- 특히 학습 문장 최소 단위의 설정과 말뭉치의 크기가 언어모델의 성능에 중요한 것으로 판단되며,
- 실험 결과 약 9만 문장으로 구성된 뉴스 말뭉치에서 **어절 단위**로 학습 및 문장 생성을 했을 때 가장 좋은 문장 생성 결과를 얻을 수 있었다.

References

[1] S. H. Gil and G. H. Kim, "Vision-based vehicle detection and tracking using online learning," J. KICS, vol. 39A, no. 1, pp. 1-11, 2014.

[2] J. H. Moon, et al., "Case study of big data-based agri-food recommendation system according to types of customers," J. KICS, vol. 40, no. 5, pp. 903-913, 2015. [3] O. Russakovskym, et al., "ImageNet large scale visual recognition challenge," Int. J. Comput. Vis., vol. 115, no. 3, pp. 211-252, Dec. 2015.

[4] S. Kumar, et al., "Localization estimation using artificial intelligence technique in wireless sensor networks," J. KICS, vol. 39C, no. 9, pp. 820-827, 2014.

[5] Y. Bengio, et al., "Learnring long-term dependencies with gradient descent is difficult," IEEE Trans. Neural Netw., vol. 5, no. 2, pp. 157-166, 1994.

...

[24] S. Bird, E. Klein, and, E. Loper Natural Language Processing with Python, O´Reilly Media Inc., Jun. 2009.