



Knowledge-incorporating BERT for Response Selection

Janghoon Han

Sogang University

120190211

한장훈

목차

서론

제안모델

실험결과

결론

서론

■ Dialog System

- 인간과 기계가 상호작용 하며 대화를 나눌 수 있게 하는 시스템.
- 여러 문장으로 이루어진 Multi-turn 환경에서 응답을 생성 or 선택

■ Dialog system의 종류

● Generative System

- ◆ 대화 History (Context) 주어졌을 때 단어 단위로 response를 생성.
- ◆ Context는 인코더를 통해 summary vector로 변환되어 디코더를 통해 response를 생성.

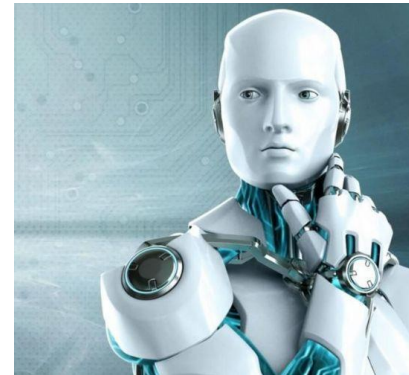
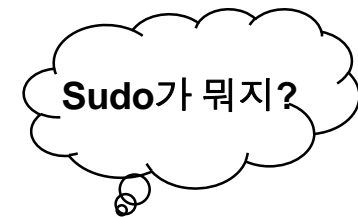
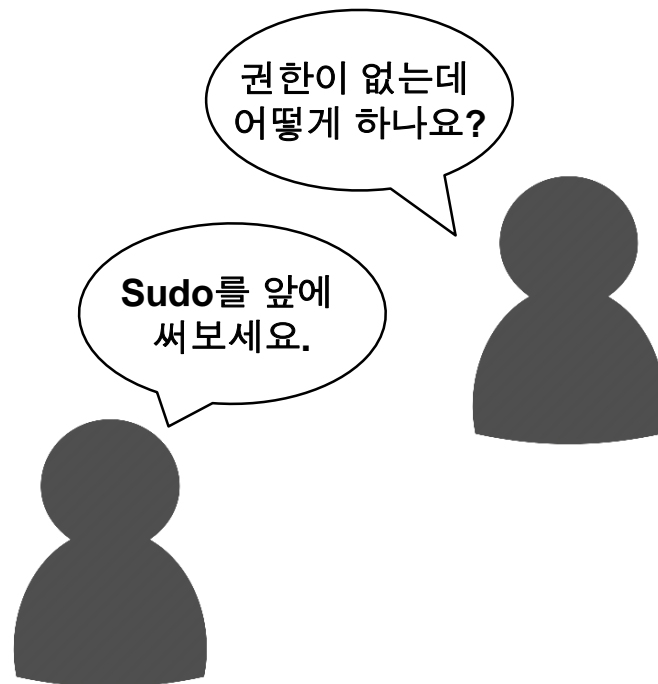
● Retrieval based System

- ◆ Response를 생성하는게 아님.
- ◆ Response candidate 중에서 Best response를 선택함 (selection)

서론

■ 기존 연구의 한계점

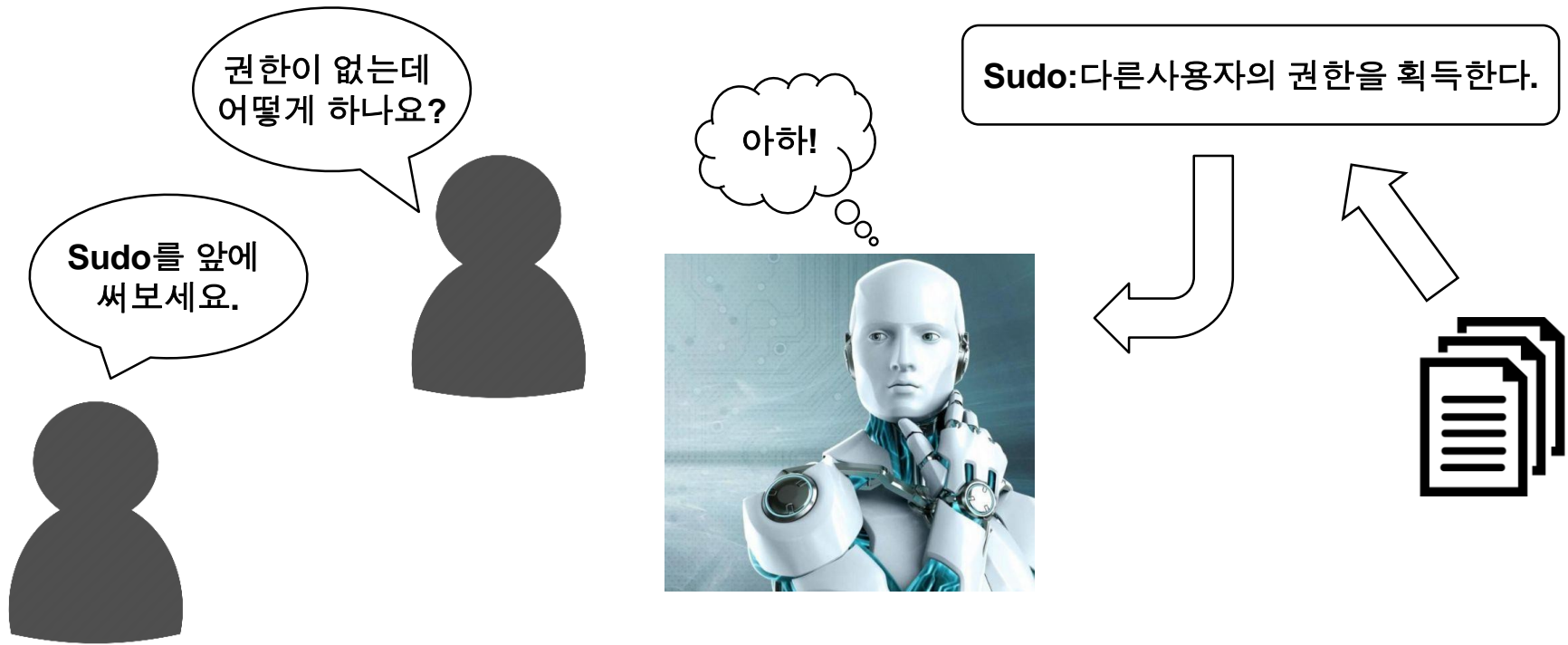
- 기존 연구는 주어진 대화 데이터만 사용함
- 대화를 하는데 외부지식이 필요할 수 있음.



서론

■ 제안모델

- 외부지식을 텍스트로부터 학습하여 response selection Task의 성능을 향상시킴.



Data

■ Dataset

- Ubuntu Corpus V1 사용
- Training data: 100K
context-response pair
- Validation, Test data: 각각
50K context-response pair
- Context는 multi-turn
환경으로 평균 8 turn을 가짐

Context

Utterance 1:

My networking card is not working on my Ubuntu, can somebody help me?

Utterance 2:

What's your kernel version? Run *uname -r* or *sudo dpkg -l |grep linux-headers |grep ii |awk '{print \$3}'* and paste the output here.

Utterance 3:

It's 2.8.0-30-generic.

Utterance 4:

Your card is not supported in that kernel. You need to upgrade, that's like decade old kernel!

Utterance 5:

Ok how do I install the new kernel??

Response

Just do *sudo apt-get upgrade*, that's it.

Data

■ External Knowledge Data

- Ubuntu command 관한 외부 지식
- Ubuntu man page의 manual description을 web crawling 하여 정제
- Ubuntu dataset에 나타난 명령어에 관한 8313개 문서
- 문서는 web의 Name 항목과 Description 항목을 사용

NAME

sudo, sudoedit - execute a command as another user

SYNOPSIS

```
sudo -h | -K | -k | -V
sudo -v [-AknS] [-a type] [-g group] [-h host] [-p prompt] [-u user]
sudo -l [-AknS] [-a type] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
sudo [-AbEHnPS] [-a type] [-C num] [-c class] [-g group] [-h host] [-p prompt] [-r role]
      [-t type] [-u user] [VAR=value] [-i | -s] [command]
sudoedit [-AknS] [-a type] [-C num] [-c class] [-g group] [-h host] [-p prompt] [-u user]
      file ...
```

DESCRIPTION

sudo allows a permitted user to execute a command as the superuser or another user, as specified by the security policy. The invoking user's real (not effective) user ID is used to determine the user name with which to query the security policy.

sudo supports a plugin architecture for security policies and input/output logging. Third parties can develop and distribute their own policy and I/O logging plugins to work seamlessly with the **sudo** front end. The default security policy is sudoers, which is configured via the file /etc/sudoers, or via LDAP. See the Plugins section for more information.

제안모델

■ Knowledge-incorporating BERT

- Pre-trained 된 언어모델인 BERT를 Dialog Selection에 적용하여 성능을 향상
- 기존 Ubuntu Dialog Corpus 뿐만 아니라 Manual Description data를 확보하여 외부지식을 사용
- 외부지식을 학습하기 위해 BERT를 Post-training 한 후, Post-trained 된 BERT 모델을 통해 Dialog Selection Task를 해결.

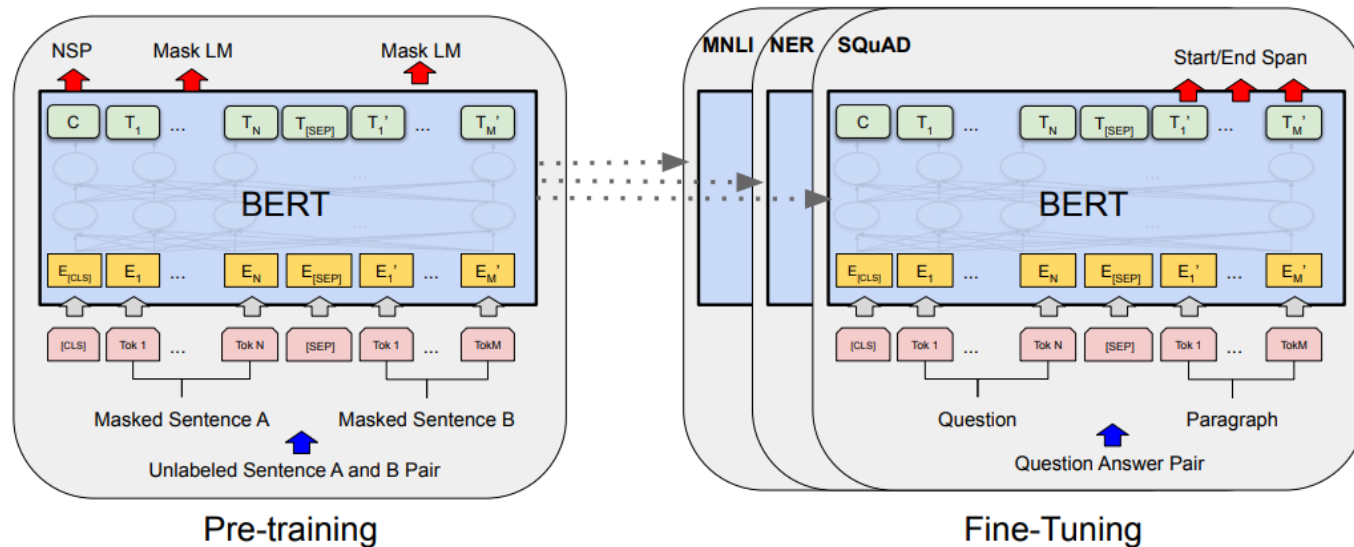
이전연구

■ BERT(Bidirectional Encoder Representations from Transformers)

- 언어모델
 - 단어 시퀀스의 확률을 할당 – 이전 단어가 주어졌을 때 다음 단어가 무엇인지 예측
 - 위키피디아, 뉴스 데이터 등 대용량 코퍼스로 학습
- 사전 학습된 BERT를 다양한 Task에 적용하여 좋은 성능을 보임



BERT



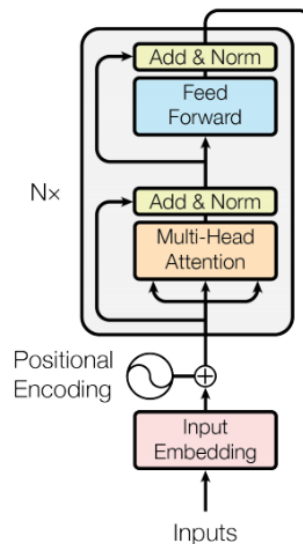
BERT

■ 구조

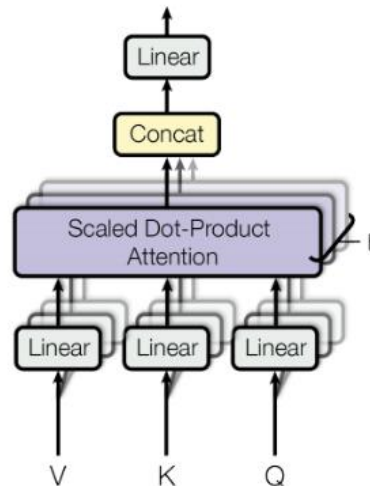
- 12개의 트랜스포머 인코더 블록, 히든레이어 크기는 768차원
- 멀티 헤드 어텐션 : 768차원을 12개로 나누어 각각 어텐션을 계산
- 셀프 어텐션 사용 (어텐션 스코어 함수는 Scaled dot product)
- 입력 : 토큰 임베딩, 포지션 임베딩, 세그먼트 임베딩



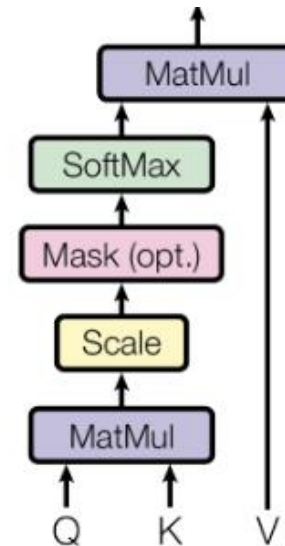
BERT



**Transformer
encoder**



**MultiHead
Attention**



**Self
Attention**

제안모델

■ Knowledge-incorporating Post-Training

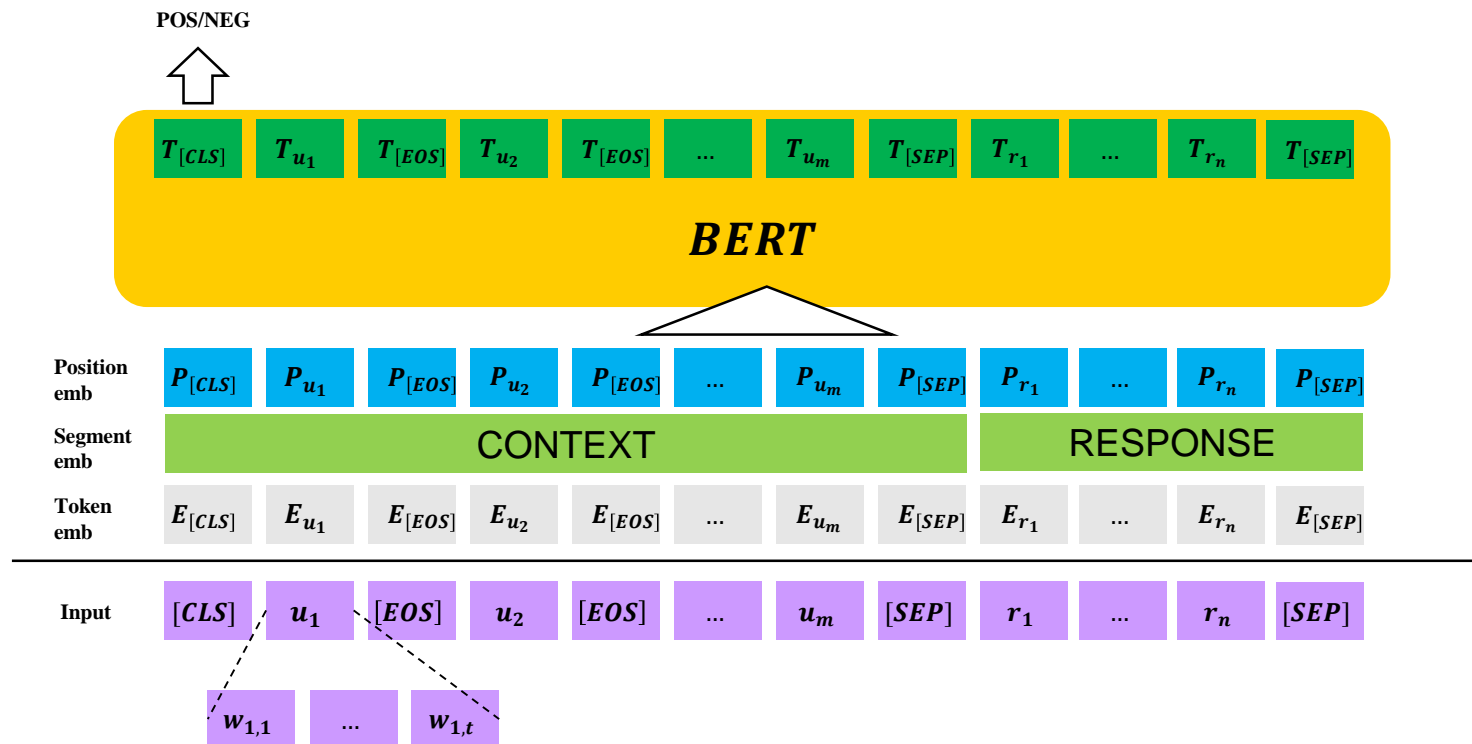
- Post-Training 데이터로 manual description data 사용
- 학습 방법으로 masked language model (MLM) 방식과 next sentence prediction (NSP) 을 사용
- **Masked Language Model (MLM)**
 - ◆ 특정 토큰을 mask 하고 mask 된 토큰을 예측하는 방식.
 - ◆ 기존 방식과는 다르게 명령어 토큰만 mask 하여 학습 - 일반적인 코퍼스에서 나타나지 않는 정보를 학습
- **Next Sentence Prediction (NSP)**
 - ◆ 두 문장을 입력으로 두번째 문장이 첫번째 문장의 다음문장인지 예측
 - ◆ 연속된 문장은 한 문서 내에서 추출하여 사용.
 - ◆ 다음문장과 관련성을 학습.
- 최종적으로 2가지 방식을 융합하여 손실함수로 학습함. 이때 Cross Entropy Loss 사용

$$L_{final} = L_{MLM} + L_{NSP}$$

제안모델

■ BERT for Response Selection

- 외부지식을 학습한(Post-trained) BERT를 사용하여 Response Selection Task를 수행함.



BERT for Response Selection

■ Problem Formalization

- Conversation dataset $D = \{(y_i, c_i, r_i)\}_{i=1}^N$, $c_i = \{u_{i,1}, \dots, u_{i,l_i}\}$, $y_i \in \{0,1\}$
- y 는 정답, c 는 conversation context, r 은 response
- c_i (Context)는 여러 개의 문장 $u_{i,1}$ 으로 이루어짐.
- Task는 matching model $g(\cdot, \cdot)$ 을 찾는 것, 즉 $g(c, r)$ 의 Score가 Context – response 매칭 정도를 나타냄

■ BERT Input

- 기존의 BERT 입력은 $x = [CLS], sentence1, [SEP], sentence2, [SEP]$
- Context와 response의 관계를 분류하는 문제이므로 다음과 같이 사용.

$$x = [CLS], u_1, [EOS], \dots, u_m, [EOS], [SEP], r_1, \dots, r_n, [SEP]$$

- CLS 토큰 : 입력된 문장들의 관계정보를 파악하기위해 넣어줌.

BERT for Response Selection

■ Initial Representations

- 입력은 포지션 임베딩, 세그먼트 임베딩, 토큰임베딩으로 표현됨
- 포지션 임베딩: 위치정보를 표현
- 세그먼트 임베딩: SEP로 나뉘며 첫번째, 두번째 문장 여부를 나눔
- 토큰임베딩: Base 가 되는 토큰 단위의 임베딩. 입력은 tokenizer에 의해 token 단위로 나눔

■ Training

- Token 단위로 출력이 되면 $T_{[CLS]}$ 을 사용하여 최종 스코어 계산
- 최종 스코어는 CLS 토큰을 FFN을 통과시켜 구함. $g(c, r) = \sigma(WT_{[CLS]} + b)$
- 손실 함수는 *Binary cross entropy* 사용

$$Loss = - \sum_{(c_i, r_i, y_i) \in D} y \log(g(c, r)) \\ + (1 - y) \log(1 - g(c, r))$$

실험결과

■ 다른 모델들과 비교

Model	$R_{10}@1$	$R_{10}@2$	$R_{10}@2$
DAM	0.767	0.874	0.969
IoI-local	0.796	0.894	0.974
ESIM	0.796	0.894	0.975
MSN	0.800	0.899	0.978
$BERT_{Base}$	0.810	0.900	0.977
$BERT_{Knowledge}$	0.824	0.908	0.978

Table 1: Model comparison on Ubuntu Corpus V1.

- Metric으로 10개의 상위 k개의 후보 중에 답이 존재하는지 평가하는 Recall 사용 ($R_{10}@k$)
- 기존 모델 보다 BERT 기반 모델이 성능이 좋음, $R_{10}@k$ 기준 2.4% 향상
- 외부지식을 학습한 $BERT_{knowledge}$ 가 그렇지 않는 $BERT_{base}$ 모델보다 성능이 1.4 % 향상

실험결과

■ Post-Training 학습 방법에 따른 비교

Loss	$R_{10}@1$	$R_{10}@2$	$R_{10}@2$
<i>MLM</i>	0.818	0.905	0.978
<i>MLM_{command}</i>	0.821	0.906	0.978
<i>NSP</i>	0.817	0.904	0.978
<i>MLM + NSP</i>	0.822	0.906	0.978

Table 2: ablation with Loss function.

- 기본 *MLM* 방식보다 명령어 토큰만 mask를 한 *MLM_{command}* 가 성능이 0.3% 향상
- 단일 *MLM*, *NSP* 방식보다 모두 사용했을 경우 성능이 약 0.5% 향상

결론

■ Ubuntu corpus Leader board 에서 2위의 성능을 달성

RANK	METHOD	R10@1	R10@2	R10@5	R2@1	PAPER	CODE	RESULT	YEAR
1	BERT-DPT	0.851	0.924	0.984		Domain Adaptive Training BERT for Response Selection			2019
2	MSN	0.800	0.899	0.978		Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots			2019
3	ESIM	0.796	0.894	0.975		Sequential Attention-based Network for Noetic End-to-End Response Selection			2019
4	lol-local	0.796	0.894	0.974	0.947	One Time of Interaction May Not Be Enough: Go Deep with an Interaction-over-Interaction Network for Response Selection in Dialogues			2019
5	IMN	0.794	0.889	0.974	0.946	Interactive Matching Network for Multi-Turn Response Selection in Retrieval-Based Chatbots			2019
6	TripleNet	0.790	0.885	0.970	0.943	TripleNet: Triple Attention Network for Multi-Turn Response Selection in Retrieval-based Chatbots			2019

0.824 0.908 0.978

- 추가적으로 implicit 하게 외부지식을 학습하는것이 아닌 explicit하게 필요한 외부지식을 추출하여 모델에 직접 사용하는 방법을 연구 중

결론

감사합니다

APPENDIX

