Misspelling Correction with Pre-trained Contextual Language Model

Abstract

- Computer는 사람과 다르게 자동으로 spelling mistakes를 잡아내기 힘들다.
- 이 논문에서는 pretrained language model인 BERT를 사용해서 spelling error를 잡아내고있다.
- 2개의 실험을 진행했다.
    1. Based on BERT
    2. Based on the edit distance algorithm

Introduction

- NER task를 예시로
    - “this shirt was bought at Grandpa Joe’s in LA.”
    - at가 ta로 misspelled 되었다면 ner을 유추해내기 쉽지 않다.
- 현대의 많은 NLP model들은 잘못된 data들에 대해 robust 하지 않다.

  이러한 잘못된 data들은 noise로 작용하기 때문에, 이러한 noise를 잡아내는게 성능에 매우 중요한 factor이다
- 문맥정인 정보로 misspelled words를 잡아낸다. Misspelled words는 2가지로 나뉨
    1. Non-word error (영단어가 아닌)

       사전을 뒤져서 해당되는 단어가 없다면 Non-word error이기 때문에 잡아내기 easy
    2. Real-word error (사용되는 영단어 이지만 문맥상 맞지 않는)

       이 문제가 어렵다.

       "access"

       "assess"와 "acsese" 는 둘다 2개의 character가 틀렸다.

       하지만 2개의 접근방식은 완전히 identical 해야한다.

       이러한 문제를 해결하기 위해 1. BERT 기반 model 2. distance 기반 algorithm
- BERT model은 transformer를 사용하여 masking된 단어를 문맥으로 맞춘다. (Attention score를 구해서) 이러한 특성을 이용해서 masking 된 spelling error를 잡아내는 방식이다.

Dateset

- CLC (Cambridge Learner Corpus) FCE (First Certificate in English) Dateset을 사용했다.

  제 2외국어가 English인 학습자들이 쓴 5124개의 sentence를 포함하고 있다.

CLC FCE dataset은 75개 종류의 annotated error로 이루어져있고, 적합한 correction을 포함하고있다. 이중 오직 1개의 spelling error만을 가지고있는 2075개 sentence를 dataset으로 사용한다.

"S" : Spelling

"SA" : American spelling

"SX" : Spelling confusion

1개의 오류만을 가지고있는 이유 : BERT에 이 error를 mask 해서 넣어줄것이기 때문에

error가 real-word인가를 구분하기 위해 NLTK라는 corpus를 사용했다.

TABLE I
DATASET STATISTICS

sentence statistics	average sentence length	20.68
	std of sentence length	11.00
	maximum length	1
	minimum length	99
error types	number of real-word errors	225
	number of non-word errors	1850

1. Pretrianed 된 BERT는 masking된 character의 가능성을 가지는 일렬의 list를 예측해낸다
2. Distance algorithm은 misspelled word와 가장 유사한 real word의 단어를 찾아낸다.

- A. Treating Misspelled Words as Masked Words

  1) Raw sentence: “How aer you today?”

  2) Replace a word with a MASK: “How [MASK] you today?”

  3) Predictions (with probability) of the MASK: “are (0.88)”, “do (0.04)”, “about (0.03)”, “have (0.003)”

  이러한 BERT의 mechanism을 기반으로 작동한다.

  Candidate words를 나타낼 수 있는 parameter은 vocab 사이즈인 N을 가진다.

  BERT의 문제점은 만약 BERT의 vocab안에 우리가 원하는 output이 없다면, BERT는 학습이 불가능하다.

  하지만 BERT는 아아주 많은 문장을 학습했기 때문에 OOV현상은 일어나기가 힘들다.

- B. Damerau - Levenshtein Distance

  이논문에서 사용한 distance관련 algorithm은 Damerau - Levenshtein Distance를 기반으로 하고있다.

  4개의 다른 연산이 존재하는데

  1. Insertion : "wat" -> "what"
  2. Deletion : "whaat" -> "what"
  3. Substitution : "wgat" -> "what"

4. Transpostion : "waht" -> "what"

Edit distance는 다른단어로 바꾸기 위해 가장 적은 연산의 횟수를 계산하는 방식으로 이루어진다

BERT의 잘못된 character prediction 결과와 잘못된 word를 Compare

이둘은 BERT의 예측결과이기 때문에 매우 비슷하다.

BERT에서 예측한 word 중에 가장 edit distance가 작은 단어를 선택한다.

인간이 misspelled된 단어를 알아차릴때 문맥보다는 단어의 유사성을 기반으로 알아차리기 때문에 Context를 담고있는 BERT에서 예측하고 있는 확률들보다는 distance를 기반으로 단어를 뽑아낸다.

example of K=2) annoying -> enjoying, annoying

2가지 방법 : BERT로 예측하고 이것에 edit distance를 적용

edit distance를 적용한뒤 BERT의 예측결과와 일치하는지 비교

Experiments and Results

1. edit distance이전에 BERT를 적용

   BERT로 예측한 결과중 N개 를 뽑아서 이들과 misspelled token 간의 distance를 비교해서 오름차순으로 정렬. 이중 가장 가까운 거리가 바로 final prediction. 만약 edit distance가 같다면 BERT의 확률값이 높은걸 final prediction으로

TABLE II
RESULTS FOR EXPERIMENT 1, BASED ON VARIOUS NUMBER OF BERT PREDICTIONS, N

top-N	accuracy top@1	accuracy top@N	P(top@1|top@N)
10	48.77%	49.64%	98.25%
50	64.34%	66.80%	96.32%
100	68.34%	71.71%	95.30%
200	71.08%	75.52%	94.13%
300	72.14%	77.20%	93.45%
400	72.92%	78.17%	93.28%
500	**73.25%**	78.84%	92.91%

Accuracy top 1 : top 1 prediction과 얼마나 label이 match 하는지

Accuracy top 2 : N개의 BERT output과 label이 얼마나 match하는지

Accuracy top 1 | top N : N개의 BERT결과와 match 혹은 edit distance의 top1과 match

N = 500일때 가장 높은 정확도를 보임 N이 500보다 커지면 distance 알고리즘이 좀 틀린다

2. edit distance이후에 BERT와 matching하는지

   Edit distance K 만큼 뽑은다음에 BERT로 이들의 순위를 매긴다.