ADL—BERT

2019年4月9日　上午 10:58

ELMo: 要依據context跟字義，去學LM，每個時間點出來layer的資訊，這個字在這個context下面的關係
BERT: bidirectional encoder representations from transformer

- 每個字要有一個represntation跟contexnt 有關，從tr4ansformer而來
- BERT只拿了encoder部分，bert不是要做generation，只用了encoder block
- 想要用在language understanding上面，所以通常是左右兩邊的資訊
- 因為不是generation task，所以沒有後面字的情況，隨機遮掉15%的token，取代一個字變成mask，predict被遮住的字是甚麼
- 最後產生mask那個字是誰，遮掉的字機率要最高
- 如果只遮5%，一樣data只會訓練到很少的target word
- 如果遮掉太多，context資訊不夠，可能會不知道那個字是誰
- input是一連串的字，可以先放入one hot或是任何embedding，在經過transformer block (encoder部分)
- 每個input時間點都會有對應的output
- OpenAI GPT
  - 只有forward，去做self-attention，沒有看後面的人，比較類似transformer decoder，multi-attention只做前面人
- ELMo
  - LSTM去做的，會把左到右跟又到左對應同一個時間點的concate起來當成embedding

第二個task:

- next sentence prediction: 很多task都是句子跟句子之間關係而來的
  - EX: QA、NLI(給兩句話看關聯)
- 知道是否連續出現，sentence-level
- 藉由data產生很多句子
- 把兩句話用[sep]分開，希望predict結果是否為next sentence關係

- 兩個task 去做optimize

  - input包含三種embedding ，concate的結果
    - token embedding
    - sentence-level embedding，會去加上哪些字是第一句話的，那些字是第二句話的
    - positional embedding
  - 會去做segment，word-piece

  - transformer encoder
  - 第一個output是next sentence prediction
  - 也要update maskLM，任意mask掉的字
  - 訓練不用label

BERT要固定長度I/O?來fine tune padding?
NER

有了訓練好的BERT，使用方式

- 有understanding task可以直接拿BERT fine tune
  - 把要做的task，在最後加上clf

BERT Large太大了，大部分用Base就夠了

Bert只拿出中間layer的embedding，拿去target task

- BERT有12層，每個字都會有12個vector
- 第一層的最爛，concate最後四層最好，但是embed size比較大
- 最後四層sum起來的資訊最豐富，可以sum或是concate
- 整個bert透過target task更新的performace比較好

結論

- 克漏字，透過transformer encoder
- 希望學到字跟字之間的關係
- 做到transfer learning
- contextual embedding approach