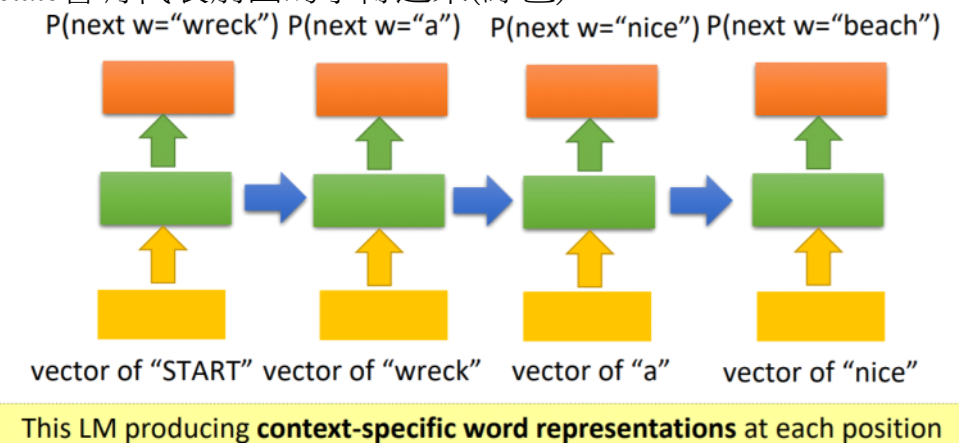
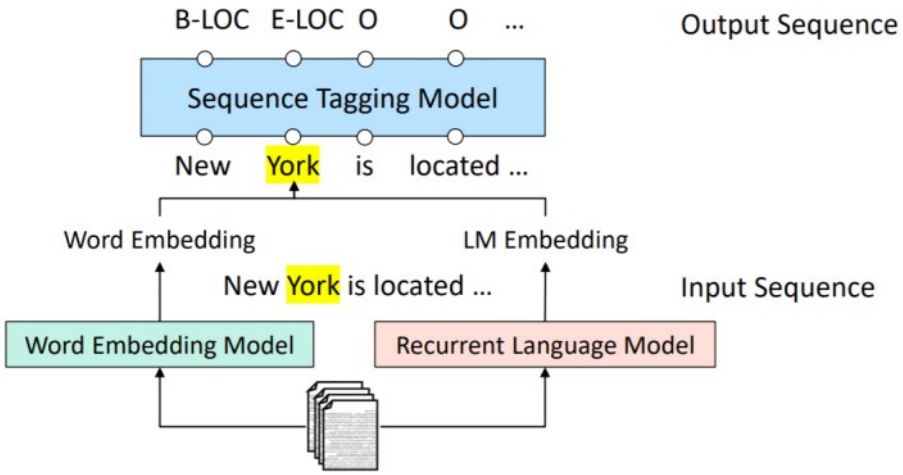
ADL—ELMo (contextual embedding)

2019年3月26日　　上午 09:27

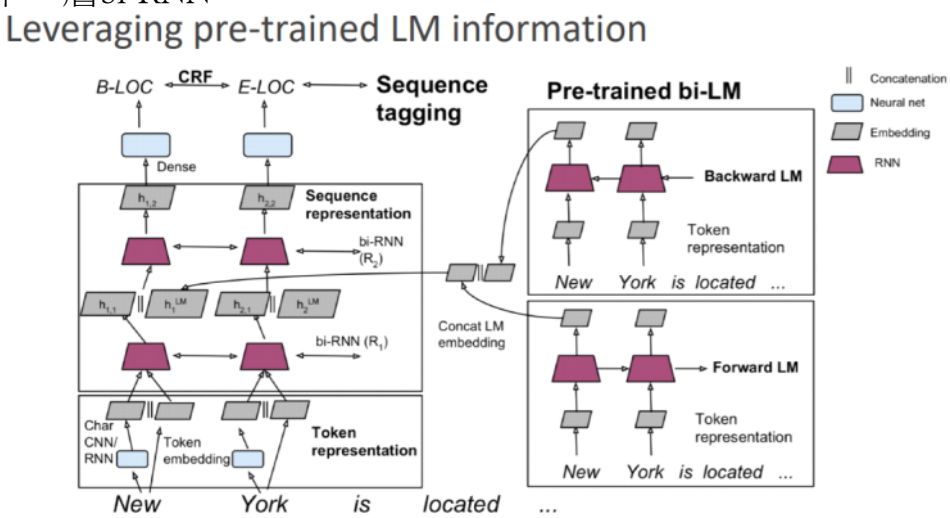
word embedding都是依據字去學習他的向量
但是其實是一字多義(multiple sense)的，所以可能這個固定向量恐怕未必可以encode
一字多義EX: rock
不同詞性會代表不同意思，單一向量無法理解，要根據前後文推估意思
希望word embedding可以根據context調整vector可能的樣子

- RNNLM
  - 希望讓NN可以condition on所有前面出現的字，去predict下一個字
  - 實際上每一個位子都匯給這個word相關的embedding，那個hidden state會有代表前面的字傳過來(綠色)
  - 

  - RNNLM每一個時間點都有把前面的context都encode起來了
- TagLM—PreELMo
  - 訓練LM不需要label只要大量data就好，contex-specific embedding，只要把input餵進去，就會在每一個時間點產生相對應的embdding，因為information都會從前面傳過來。再去做目標的task
    EX:去標註哪些位子是專有名詞，label太少，所以可以用相關的LM去預測哪裡具有專有名詞
  - EX: 希望可以detext 哪裡有人名地名
  - 傳統只有左半邊(綠)，現在有右半邊(粉)可以去encode前面字跟字之間的關聯

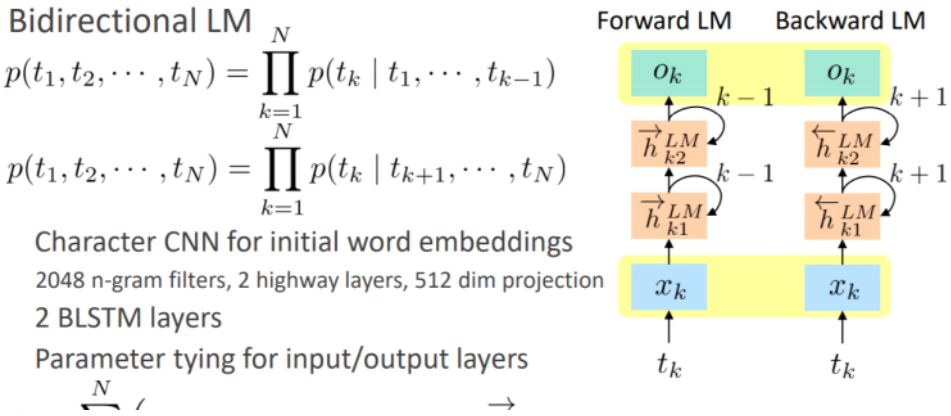


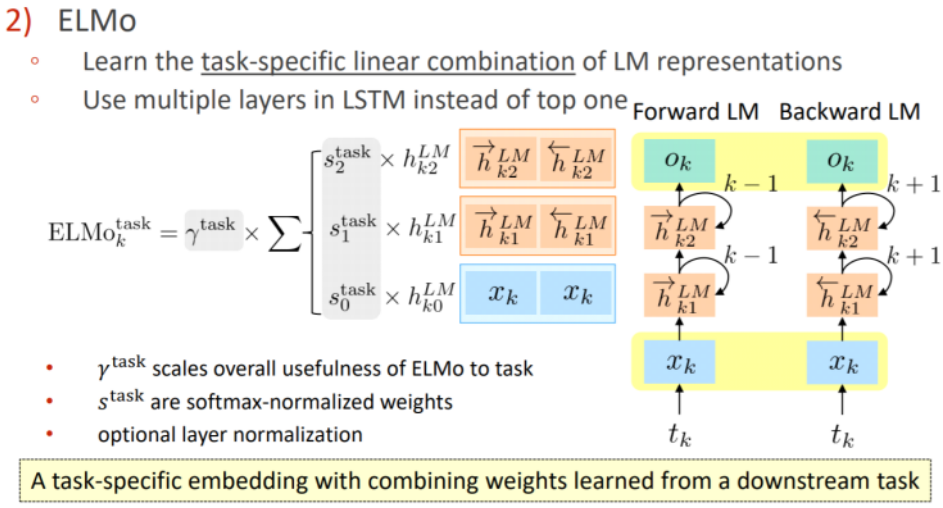
- character CNN/RNN防止OOV，人名不在字典當中 所以不會有tag，使有機會tag出人名
- 左邊兩層bi-RNN
- 經過第一層後會把兩邊的LM提供的vector concate在一起，在經過下一層bi-RNN



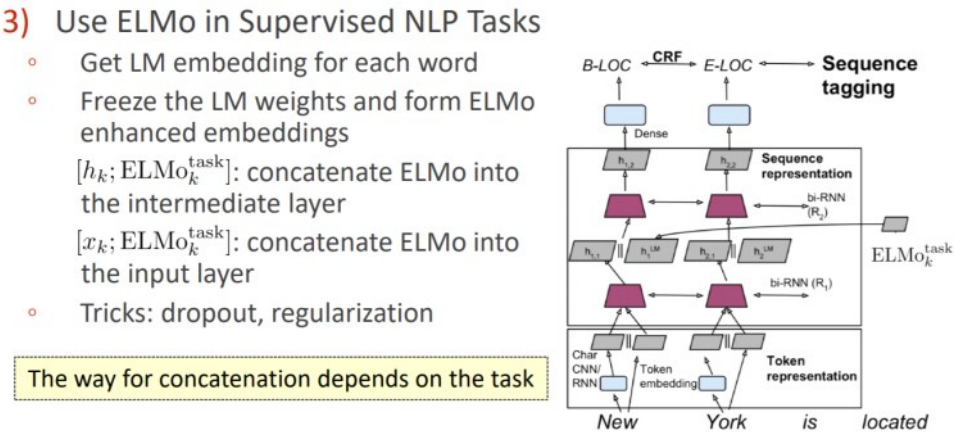
- CoVe
  - 利用sequence model資訊給target task資訊，當成context
  - machine translation(MT)可以捕捉到sequence上面的meaning
  - 把MT所學到的資訊傳給task，希望可以embed整句的context意思
  - 比GloVe好，但結果沒有Neural LM還好
    - 用比較困難的task的資訊給比較簡單的task資訊會壞掉
    - 字跟字之間的關係從LM來學，供應給困難task的資訊比較有效。因為MT所encode的資訊並不會比LM的資訊還要多
  - 沒人用了

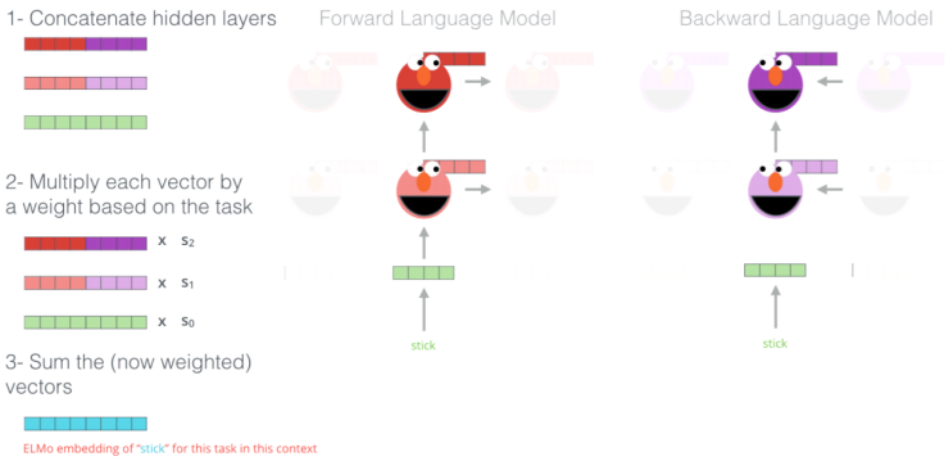
- ELMo
  - 藉由很長的context去encode previous 資訊
  - RNN-based LM
  - 每一層的contribution可能會不太一樣，中間是deep LM (LSTM)
  - 要做bidirectional LM
    - 一個forward (左邊到右邊)

    - 一個backward
- token進去會先經過embedding EX: word2vec / char CNN (initialize embedding)
    - 2048 n-gram filter、2*hiway layers、512 dim，可以跟後面 jointly train
- 2*BiLSTM
- 把input跟output layer的weight tie在一起，所以最後只會有一組輸入跟輸出的參數，encode of input跟 output要share，中間的forward backward會不一樣(LM)
- 

- 先train好LM
- 在藉由這個LM，去跟最終task 弄在一起
- 會有不同weight，根據不同task不同的layer會不一樣重要
- gamma是要對整個ELMo資訊進行scale，因為這個context的information沒有provide太多資訊。三組vector做weighted sum的weight相加=1，最後在經過gamma調整
- 可以去做layer normalization
- 

- 學了一個ELMo，task specific
- 會有不同的concatenate方法，看要接在哪一層，可以多試試看(input/hidden)
    - 依照task
- 要做dropout跟regularization

- 

  - 把不同layer concate 在一起
  - 把前後意思concate起來，乘上個字的weight，放到target task layer裡面，在去訓練，獲得額外資訊
  - Transfer Learning
    - pre-train可以獲得一些資訊了，LM embedding可以在怎麼樣context下可以包含一些語意，讓embedding有基本知識，可以幫助各種不同的NLP task
  - 可以解決一字多義，可以依據鄰居知道他的意思在哪裡，不同context的embedding會不太一樣，所以不好viz哪一個字在哪個位子
  - 分析
    - 比較接近input的layer會encode syntax資訊 EX:PoS，第一層的資訊比較重要
    - 離output比較進會比較偏向語意抽象的資訊，可不可以區分開
  - 提供其他clue從別的地方來的資訊
- 

  - Pre-trained ELMo: https://allennlp.org/elmo
    - PyTorch allennlp
    - 有處理character的資訊，所以task也是有考慮補充資訊

=助教補充=

- OOV、UNK=>當作不知道是甚麼字的解法
- sub-word embedding EX:apple+care / un-fortunate-ly
  - 這些字可以出現在其他字的地方
  - vocabulary裝的是sub-word，把UNK轉換，

maxpooling/minpooling/avgpooling等等
    - 把樹狀資訊組成起來
        - morphological RNN
    - EX: 台灣大學生喜歡深度學習，中研院斷詞系統可能不夠好
        - 可以拆成深度、學習
    - EX: So gooooooooooood ，中間的o不一定，冗於部分可以改掉
    - 拆解幾個字?
        - EX: apple=[app ppl ple]
        - byte pair encoding
            - 把長間的n-gram sub pair取出來當成vocabulary
- character-level
    - 過一個model得到一個contextual model
    - 一個字字的，只要26個字母就可以解決，儲存英文字母
    - 取word vector那一層，接成end2end去學習該怎樣learn
    - 動態生成，拿pretrain word embedding，希望model跟pretrain model越像越好
    - MIMICK word embedding
    - pretrain
- FastText
    - embedding toolkit by FB
    - word2vec的skip-gram加進n-gram
    - supervised objective
- Sentence/Document Embedding
    - 把一串文字丟進去其實就是sentence embedding
    - 把character換成word就是snetence、sentence丟進去就變成document
    - skip-thought
        - 給定目前句子，預測前後兩個句子
        - 產生句子: generation
        - sentence 前後產生
    - Quick-thought
        - 給定中間句子，變成前後classification句子
        - 多收集冗於句子，冠詞定冠詞stop words不用學
    - InferSent
        - NLI訓練一個model去學 (給定假設句、前提句，判斷兩句的關係)

            把sent部分的句子拿出來
        - 對hidden state做over dimension做max pooling
        - downstream task，希望train好可以learn generalize
        - 理解了兩句的關係代表已經learn好了解了句子]
        - 不要train model可能比較好?!
- SIF : smooth inverse frequency
    - sentence embedding不用train
    - 把word embedding做weighted sum

- 字很常見就給少一點weight，因為包含的資訊比較不多
- 把weighted sum好的作PCA
- 取出vector當中取出重要的維度，把重要的維度learn出來，把常見共有的減掉
- 把pretrain embedding做weighted sum在去減掉共同東西
- 會贏過NN-based方法