MLDS 2017 Assignment 1

Problem Description

The Microsoft Sentence Completion Challenge

這次作業的目的是要解決語意填空的的選擇題問題，每一個問題的句子都會包含一個挖空的空格以及其對應的五個選項，並從中選擇一個最符合語意的選項為答案，並使用正確率(Accuracy)來當作衡量的標準。

Example:

The darkness was _____ , but much was still hidden by the shadows.
(a) rising, (b) healed, (c) ponderous, (d) neglected, (e) attractive

測試資料的來源是Sherlock Holmes novels，總共有1040句測試資料。訓練資料的來源也是來自於相同的資料集，總共有236MB，以便我們的模型去學習其中的語意資訊。

Environment

- OS: Linux
- CPU: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, Memory: 64GB
- GPU: GeForce GTX 1080, Memory: 8GB
- Libraries:
  - Python - 3.4
  - Tensorflow - 1.0
  - NumPy - 1.12.0
  - gensim - 1.0.1 (只用來拿出pre-trained word2vec embedding)

Model Description

Data Preprocessing

由於Sherlock Holmes novels的文章中，每一篇都包含著許多與內文不相關的部分(目錄、前言)，為了避免影響訓練的結果，我們將原始的文章去做了一些預處理，移除冗餘的部分，並且將文章根據結束符號做分段，將原始的文章切成介於10~40長度的句子，並且將不足40長度的文章做padding，且在每一句的頭尾的部分加上<START> <END>的字彙。

<START> But I guess we can solve the problem <END> <END> <END> . . . .

接著建構出整篇文章的所包含的字彙，將出現次數少於100次的的字彙移除，並代換成<UNK>，最後將已經切斷好的句子中的每一個字彙對應成其相對應的ID，生成訓練資料。

Model Structure

為了建構訓練資料中的語言模型，我們採用的是一層RNN的model，其中我們使用LSTM當作RNN中傳遞的cell。LSTM (Long Short Term Memory)，同時存在hidden、cell state， 透過一個線性傳遞的cell state能有有效的保存長句子中的資訊，幫助模型建構出文章的表示向量，最後再透過一個分類器將每一個位置的表示向量去做分類，方式是先做完softmax後，使用cross entropy作為損失函數(loss function)，並用Adam演算法做back propagation將模型中的參數更新。

其中每一筆訓練資料的輸入的預測目標，是透過將輸入資料做平移(delay)，將平移的結果當作每一筆要預測的目標，也就是說當平移一格時，每一個字在模型中所要預測的結果是，他在同一個句子中的下一個字。

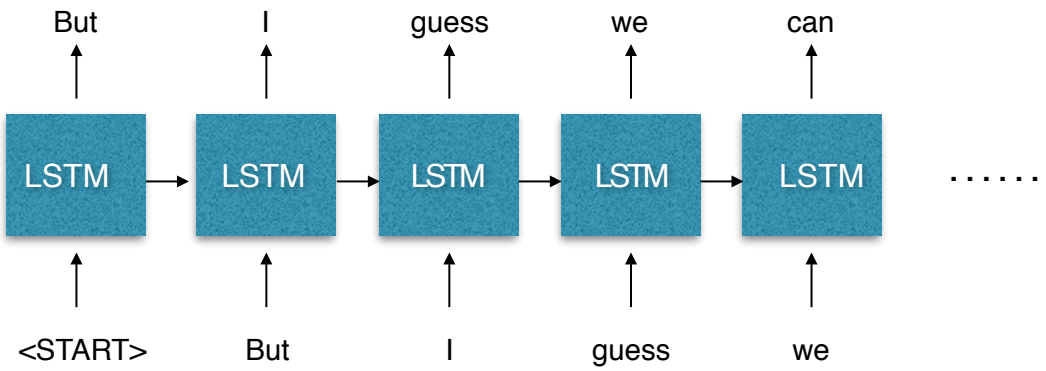

Figure 1: Model example (delay=1)

Inference

在做測試資料的預測的時候，我們會先將每一個選項填入空格之中，因此每一筆測試資料會產生出五個句子，接著利用已經訓練好的模型去計算這五句的loss，並將產生loss最小的句子的那個選項當成是這個句子的預測結果，將這個方法套用在所有的測試資料上產生出最後的結果。

Model Details

基本和best的版本使用相同的模型。

訓練資料：13705個字彙，2433185個句子。

模型：一層Basic LSTM，hidden layer size: 400，delay: 1，batch size: 50，Adam optimizer, learning rate: 0.002, epoch: 8。

Improvement

Pre-trained word embedding

每個字彙會根據自己的ID去查對應的word embedding，從one-hot encoding變成一個向量的表示法，由於我們將word embedding設為一個可以訓練的變數，因此在模型訓練中會不斷被更新。

我們試了兩種初始word embedding的方式，第一種是隨機產生初始值，第二種使用Google的pre-trained 300維 word2vec embedding。根據下表，我們發現這兩種初始方式最後的結果差不多，沒有哪一種一定比較好。隨著模型訓練的迴圈數越多，初始值越來越不重要，因此有無使用pre-trained word embedding 差異沒有很大。在我們最好的模型中，我們沒有使用pre-trained word embedding

Pre-trained word embedding	Public Accuracy	Private Accuracy
not used	0.27885	0.32885
used	0.29615	0.31923

Delay

Delay K代表RNN在輸入K個input後才開始預測句子的第一個字，因此每個輸出的字根據前面至少K個字來決定。我們最後的模型選用 delay = 1。

Noise-contrastive estimation

不直接計算所有字可能出現的機率，loss只取target word和K個noise samples來做計算，計算量變少，因此可讓訓練時間變短，但僅能使用在training階段。Test時依然要使用一般的softmax計算loss。

Experiments

Basic Settings

在以下實驗中，我們預設的模型為上述介紹的RNN模型，其中字彙量為11562，batch大小為100，learning rate為1.0，layer數為1，hidden大小為256，delay為1，embedding使用Word2Vec pre-trained 300dim embedding作初始化，訓練的epoch總共為5次。

Delay

-performance

Delay	Public Accuracy	Private Accuracy
1	0.29615	0.31923
3	0.30962	0.35192
5	0.28654	0.31538
7	0.26293	0.28846

-analysis

在這個實驗中，我們討論的是delay長度對正確率的影響。我們嘗試了delay為1, 3, 5, 7長度。發現在delay為3的時候效果最好，更長的delay反而造成負面影響。較長的delay我們期待能讓模型在看過足夠多的輸入資料後再做預測，因此效果較好。而過長的delay可能超出要預測的字的影響範圍，而前面多出來的字與要預測的字的語意關聯不大，訓練時會成為noise，使得模型預測出錯。

Layer

-performance

Layer	Public Accuracy	Private Accuracy
1	0.29615	0.31923
2	0.38846	0.41154
3	0.36923	0.42500

-analysis

在這個實驗中，我們討論的是layer數對正確率的影響。其中layer為2時效果最好，更多的layer數並沒有顯著影響。越多層的模型越複雜，理論上在足夠多的資料下應該效果比較好。但是多層的模型可能需要較多的訓練時間，或者因為overfitting使得平均正確率些微下降。

Model

-**RNN**: 見「Model Description」裡的「Model Details」

-**LSA Similarity**: 對training data使用LSA (Latent semantic analysis)抽取每個字500維的embedding。Test時，每個選項對句子中每個單字計算cosine similarity加起來，總和最大的取為答案。

-**Word2Vec Similarity**: 同LSA similarity計算方法，但使用pre-trained Word2Vec當作字的embedding。

-performance

Model	Public Accuracy	Private Accuracy
RNN	0.43462	0.51538
LSA Similarity	0.42308	0.44231
Word2vec Similarity	0.33846	0.38077

-analysis

三個模型中，RNN有最好的performance，因為RNN不單單考慮句子中有哪些字，也會考慮字出現的順序、一個字和前面的字的關係。若算一個字和整個句子的cosine similary，那就只考慮字的意思，沒有字與字之間的前後關係。從實驗中也發現使用LSA抽取每個字的embedding比使用pre-trained Word2Vec好，一個可能的原因為LSA使用的是我們的training data，所以對文本中的字有比較好的表示法，但pre-trained Word2Vec沒有看過我們的training data，因此沒辦法精確代表這個文本中的字。

Team Division

組員	分工
江東峻 r05922027	data preprocessing, LSTM, delay和layer實驗, report
陳翰浩 r05922021	data preprocessing, Bidirectional LSTM, report
鄭嘉文 r05922036	data preprocessing, LSTM, LSA similarity, Word2vec Similarity, report