Final Project Report - Video Caption

隊名：NTU_r07946015_DiuDiu

一、Members :

1. 學號：R07946015 系級：資料科學學位學程碩一 姓名: 許睿修
2. 學號：R07946007 系級：資料科學學位學程碩一 姓名: 陳庭安
3. 學號：R07942075 系級：電信工程學研究所丙組碩一 姓名: 陳郁棋

二、Introduction & Motivation :

這次期末專題的題目，是希望我們能從五個文字的選項中，挑選一個選項可以最正確地描述一支影片。我們有主要的三個想法來定義這問題，也分別使用三種模型來預測結果。

第一種是視為 ranking 的問題，也就是說，訓練模型的目的是要比較選項之間哪一個比較適合這支影片。常見的作法有 pointwise 、 pairwise 或 listwise ， 我們選用最簡單的 pointwise ， 對於每支影片的每個選項給予一個分數，代表選項正確描述影片的機率，然後從五個選項中挑選分數最高的作為預測結果。實作上使用 DNN 模型。

第二種是視為分類問題，將選項拆分成主詞、動詞、受詞的結構，個別訓練一種分類用的模型，預測一支影片最有可能出現的主詞、動詞、受詞分別為何，再與選項做比較。實作上使用三個 Random Forest 模型。

第三種是視為 sequence-to-sequence 的問題，概念上是訓練一個模型能夠將影片轉換成描述該內容的一句話。具體上比較有挑戰性的地方，在於把影片（連續的圖片）encode 成能代表這支影片資訊的向量，然後再透過 decoder 產生句子（連續的文字）。透過比較模型的輸出與五個文字選項相似性，選出相似性最高的選項作為預測結果。

接下來的幾個章節，會先介紹資料前處理的過程，再來描述我們使用過的模型，以及模型的測試結果和表現，最後則是總結。

三、Data Preprocessing / Feature Engineering :

1. Text Preprocessing

- Tokenization – 去除標點符號、全轉小寫字母
- Spelling Correction – 拼字校正
- Remove the stop words – 刪除常見，較無意義、無代表性的字，如 is
- Remove the words that not in the English dictionary – 刪除不出現在字典中的字
- Unify the tense of verbs and singular/plural nouns – 動詞時態、名詞單複數統一

2. Video Preprocessing

- Use Vgg16 extract features from a short video clip. (助教預先處理過的資料)

四、Model Description and Discussion:

1. DNN

- Input：

  模型的輸入為每支影片 80 個 frame 的 feature map，以及影片對應到的 caption。 Caption 最多 20 個字（不足的做 padding ），每個字經過 word embedding 轉成 300 維的向量。將兩者 concatenate 起來，變成一個 80*4096 + 20*300 的向量。

- Output：

  模型的輸出為介於 0～1 的浮點數，其數值代表的含義為該句 caption 正確描述影片的機率。

- Loss：

  Weighted binary cross entropy

- Model：

Layer (type)	Output Shape
dense_9 (Dense)	(None, 256)
leaky_re_lu_7 (LeakyReLU)	(None, 256)
batch_normalization_7 (Batch	(None, 256)
dense_10 (Dense)	(None, 128)
leaky_re_lu_8 (LeakyReLU)	(None, 128)
batch_normalization_8 (Batch	(None, 128)
dense_11 (Dense)	(None, 32)
leaky_re_lu_9 (LeakyReLU)	(None, 32)
batch_normalization_9 (Batch	(None, 32)
dense_12 (Dense)	(None, 1)
activation_1 (Activation)	(None, 1)

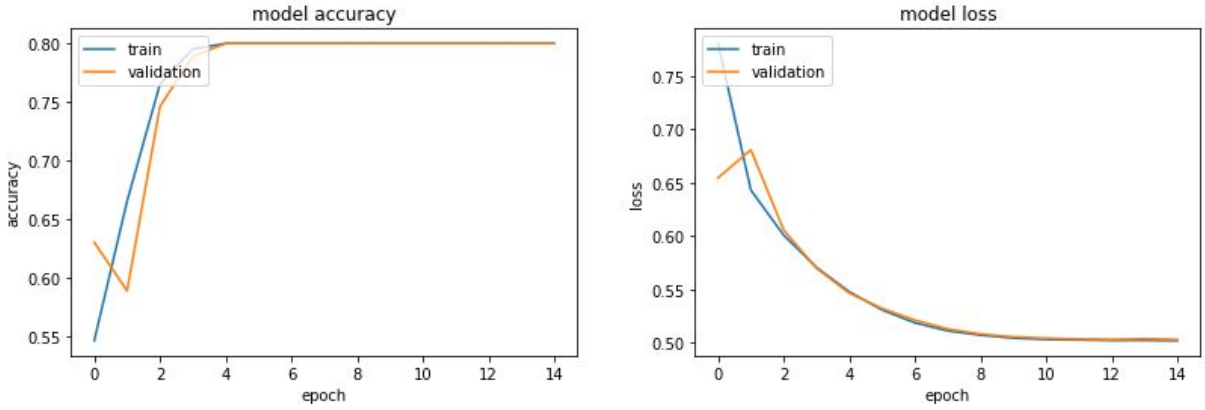
- Experiment：

  a. Positive and Negative Examples

  以每支影片 ground truth 的 caption 當作 positive example，label 設定為 1；從其他 4 支影片隨機抽出一句 caption 作為 negative examples，label 設定為0。

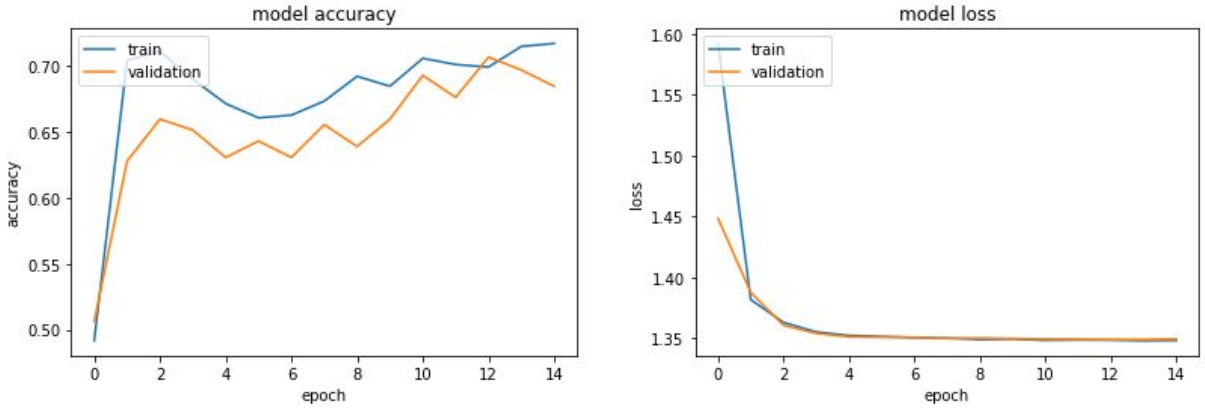
b. Training and Validation Accuracy

我們發現訓練模型的過程如下圖，在 training 和 validation 的準確率都會收斂到 0.8 這個值，剛好為 positive 和 negative examples 的比例，可以合理地推測模型會預測所有選項的機率接近 0 。



c. Weighted loss function

由於上述的情況發生，我們相信模型並沒有真正的學習到如何辨識 positive 和 negative examples，為了強調在 positive example 上的學習，同時避免預測全部為 0 的情況發生，我們選用了 weighted binary cross entropy。從下圖可以發現準確率不會直接收斂到 0.8 了。



d. Kaggle Result

Loss Function	Kaggle Public Score
Binary Cross Entropy	0.28799
Weighted Binary Cross Entropy	0.30200

e. Discussion

由 Kaggle Public Score 可得知，使用 weighted loss function 有較高的準確率，因此我們發現強調 positive example 可有效地讓模型學習到比較多的資訊。值得一提的是，從 testing data 預測的結果可以觀察到，同支影片不同選項預測的機率差異在 1E-5 的數量級，顯示模型大部分的資訊量被 4096 * 80 維的影片向量 dominate，選項的 300 * 20 維向量則對預測結果影響較小。

2. Random Forest

- Main idea：

  一般句子常常會由主詞、動詞、受詞所組成，因此從 captions 抽出其主詞、動詞以及受詞，並用三個 Random Forest 分別去預測每一個 frame 所生成的主詞、動詞、受詞，並與 groud truth 的選項做比對。

- Text Preprocessing：

  利用 nltk.pos_tag 來進行句子的詞性標注，並判斷一個詞的 pos = 'NN' or pos = 'NNS' 且出現在句子的前五個 (index <= 5) 則為該句的主詞；出現在後面且 pos = 'NN' or pos = 'NNS' 的詞則為受詞；而一個詞的 pos = 'VB' 則為該句的動詞。

  此外，利用 nltk.stem 中的 WordNetLemmatizer 進行詞形還原，例如：dogs -> dog, doing -> do。並設定主詞、動詞、受詞出現頻率的 threshold，分別是 3, 2, 3，藉此過濾掉頻率極低的詞。

- Random Forest：

  利用 sklearn.ensemble 中的 RandomForestClassifier 進行實作。分別針對主詞、動詞、受詞 train 三個 Ranfom Forest。其中每個 Forest 的 input 為一個個 4096 維的 frame，而 output 為該 video 所對應到的主、動、受詞。

- Experiment：

  a. Turing the parameter： n_estimators
     - 比較參數：n_estimators (The number of trees in the forest.)
     - 固定參數：max_depth = default
     - 衡量標準：oob_score_ (Score of the training dataset obtained using an out-of-bag estimate.)

n_estimators	Subject	Verb	Object
50	0.8692	0.8771	0.8524

75	0.8794	0.8964	0.8655
100	**0.8831**	**0.9062**	**0.8727**

b. Turing the parameter： max_depth
   - 比較參數：max_depth = default, 10
   - 固定參數：n_estimators = 100
   - 衡量標準：oob_score_

max_depth	Subject	Verb	Object
default	0.8831	**0.9062**	**0.8727**
10	0.5275	0.4229	0.4233
30	0.8706	0.8741	0.8306
40	**0.8837**	0.8981	0.8613

c. Discussion

由上述 a 部分的實驗可得知，當 n_estimators 數越大時，out-of-bag estimate 分數會越高，代表該 Random Forest 能夠越 fit 此 training data。一般來說 n_estimators 太小，會不太能夠 fit 原始 training data；n_estimators 越來越大時，計算量也會相對提升許多。例如在實驗中，n_estimators 分別為 50 以及 100 時的執行時間就差了將近快兩倍，因此我們決定將 n_estimators 參數停留在 100，以避免 n_estimators 升高所造成的龐大執行時間。

上述 b 部分為針對 Random Forest 的最大深度參數所進行實驗測試。由實驗數據可得，普遍來說 max_depth = default，也就是 Decision Tree 在建立 sub-tree 的時候不會限制 sub-tree 的深度，這個參數在 out-of-bag estimate 分數的表現上較好。

d. BLEU: a Method for Automatic Evaluation of Machine Translation

BLEU的目標為提出一個衡量方式，計算機器翻譯跟實際 ground truth 語句之間的相似程度。其主要概念為比較 candidate 的 n-gram 與 reference 的 n-gram 有對應到的個數。假設 match 到的 word 是 position-independent，因此對應到的數量越多，代表 candidate 翻譯得越好。然而為了避免 “the the the the the the the.” 這種不合理的語句出現，在計算 modified unigram precision 時，當 reference 中的字一旦被對應到就不會重複被考慮。

e. Kaggle Result

拿上述 out-of-bag estimate 分數表現最好的參數（n_estimators = 100, max_depth = default）來進行 Testing 。不過這裡拿 BLEU 來作為與 options 的比較衡量標準。在 Kaggle Public Score 的表現不如預期，推測可能的原因為最後不太適合拿 BLEU 當做比對 predicted output 與選項間的相似程度，因而可能會因為一些句法或是型態的不同而導致相似意思的兩個句子被衡量為相似度較低，因此容易選擇了錯誤的選項。

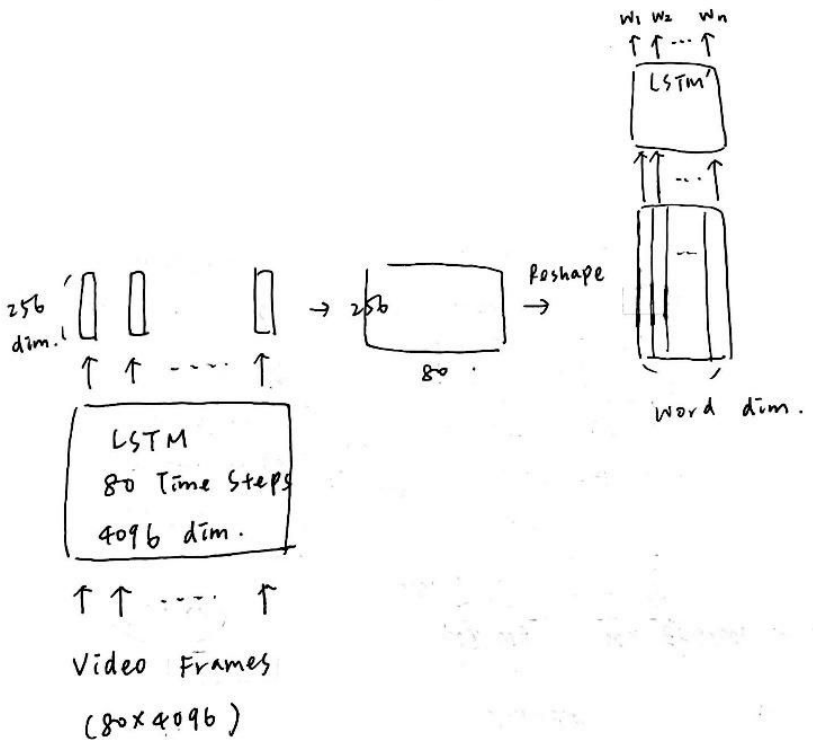
Model	Kaggle Public Score
Random Forest + BLEU	0.23200

3. Sequence to sequence

I. without Captions Information as Inputs

- Notion:

將 4096 維 Video frames 透過 LSTM 保留 frames 間的連續性特徵，每一 Time step 輸出 vector stack 成 matrix 作為該 Video representation 。將 video representation reshape 後作為另一 LSTM 的輸入，期望每一個 Time step 輸出 caption 的字。最後 predict 輸出的一句 caption 再與選項一一作 cosine similarity ，選擇最相似的一個選項。

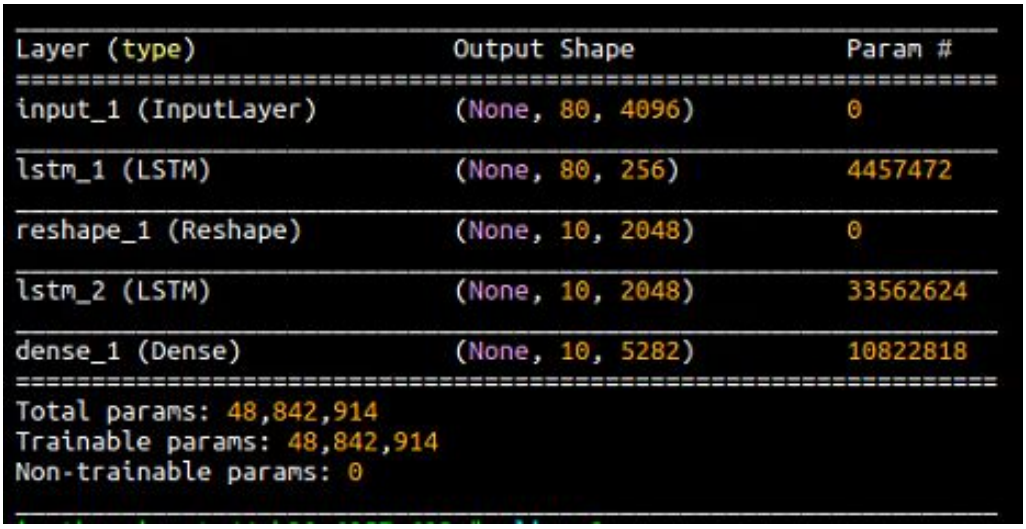


- Data Preprocessing and Feature Engineering:

  經過「三、Text Preprocessing 」後的 Captions ，取 Training data Captions 其中所有文字以及 <PAD>, <SOS>, <EOS> 與 <UNK> 建 vocabulary 與 Indices，並透過建好的 Vocabulary 與 Indices 將 Captions、Options 轉成 Lists of indices 的形式。其中，不在 Vocabulary中的文字對應到 <UNK> 的 Index。此外，由於 99% 的 Captions 長度均不超過 10 字，故所有 Lists of indices，即 Captions 與 Options 超出 10 字截斷，不足 10 字補上 < PAD > 的 Index。故 Decode 的 LSTM 的 Time steps 為 10，又 words 以 indices 表示，因此 LSTM 輸入的 dimension 為 1 。

- Model:
  - Input 1: Video frames (80 x 4096) [Encoder inputs]
  - LSTM 1: Encoding the videos as 256-dimensional vectors [Decoder inputs]
  - Reshape 1: Reshaping to cater to the specific length of the output

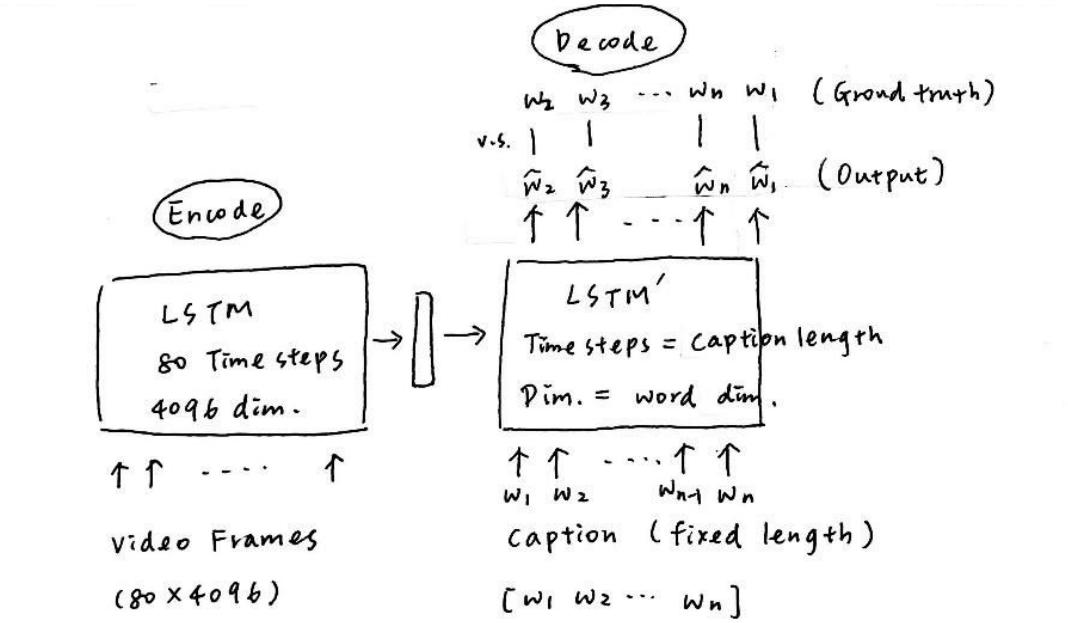
sentences
- LSTM 2: Predicting the words of a caption
- Dense 1: Rearranging the dimension to 5282, the size of the vocabulary built
- Y_PRED: The outputs of Dense 1
- Y_TRUE: One-hot codes of the words in the captions
- Loss: Cross_entropy_error(Y_PRED, Y_TRUE)



II. with Captions Information as Inputs

- Notion:

  將 4096 維 Video frames 透過 LSTM 保留 frames 間的連續性特徵，輸出 vector 作為該 Video representation 。再將 video representation 與 word2idx 後的 Caption 資訊作為另一 LSTM 的輸入，期望每一個 Time step 輸入 caption 一個字後，會輸出下一個字。最後 predict 輸出的一句 caption 再與選項一一作 cosine similarity，選擇最相似的一個選項。



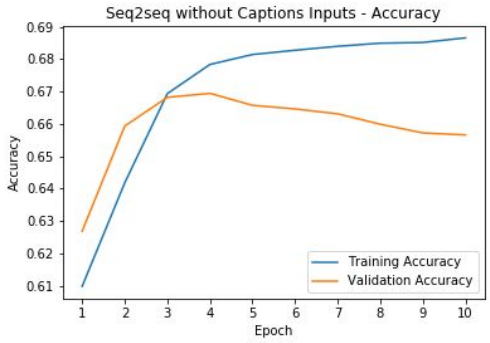
- Data Preprocessing and Feature Engineering:

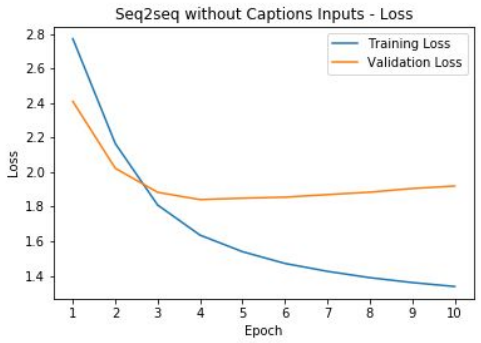
  同 **Sequence to sequence I. without Captions Information as Inputs**

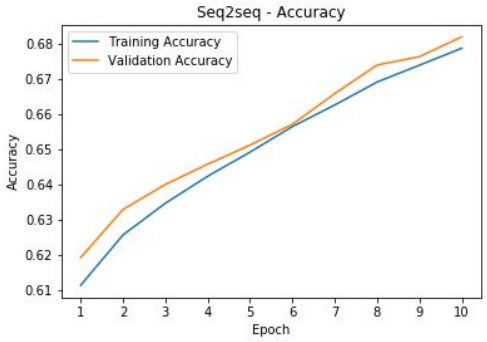
- Model:
  - Input 1: Video frames (80 x 4096) [Encoder inputs]
  - Input 2: Captions – lists of indices (10 x 1) [Decoder inputs]
  - LSTM 1: Encoding the videos as 256-dimensional vectors
  - LSTM 2: Predicting the next word of a caption in each time step
  - Dense 1: Rearranging the dimension to 5282, the size of the vocabulary built
  - Y_PRED: The outputs of Dense 1
  - Y_TRUE: One-hot codes of the words in the captions
  - Loss: Cross_entropy_error(Y_PRED, Y_TRUE)

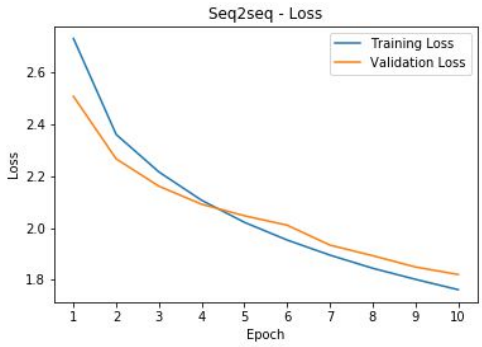
```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
input_1 (InputLayer)         (None, 80, 4096)          0
__________________________________________________________________________________________________
input_2 (InputLayer)         (None, 10, 1)             0
__________________________________________________________________________________________________
lstm_1 (LSTM)                [(None, 256), (None, 4457472     input_1[0][0]
__________________________________________________________________________________________________
lstm_2 (LSTM)                [(None, 10, 256), (N 264192      input_2[0][0]
                                                              lstm_1[0][1]
                                                              lstm_1[0][2]
__________________________________________________________________________________________________
dense_1 (Dense)              (None, 10, 5282)          1357474     lstm_2[0][0]
==================================================================================================
Total params: 6,079,138
Trainable params: 6,079,138
Non-trainable params: 0
__________________________________________________________________________________________________
```

- Experiment and Discussion:









Models / Methods	Loss Function	Kaggle Public Score
Seq2seq (without Captions Information) [Epoch = 3]	Cross Entropy Error	0.21200
Seq2seq (with Captions Information) [Epoch = 10]	Cross Entropy Error	**0.45400**

由 Accuracy, Loss 在不同 Training epoch 的趨勢圖可知，沒有加入 Captions 資訊當 inputs 的 model ，雖然一開始表現很好，但很快就 overfitting，而且 validation accuracy 開始下滑。最好的預測情況出現在第三個 epoch ，但在 testing 的結果卻只有 0.212 。

第二個模型我們使用類似作 Language translation 的 Sequence to sequence model ，每輸入 Caption 的一個字就輸出一個預測結果，期望能預測 Caption 下一個字。模型準確度就逐步增加，即使截在 epoch 10 ，準確度還是穩定成長。在 Testing data 的準確度也到 45.4% 。

第一個模型會失敗的原因除了少了 Captions 的資訊外，還有可能是在模型設計上，video representation 進入下個 LSTM 時作了 Reshape 的步驟，讓原本的 frames 的順序性部分被打亂。會 overfitting 的原因可能為，Video frames 為唯一 Inputs ，而 Training data 與 Testing data 的 Video差異大，可能導致預測會失真。

五、Overall Discussion :

本次題目是五選一的多選題，如果用隨機猜選項的方式來預測，準確率的期望值為 0.2 ，在各個實驗過的模型中，Sequence to sequence without caption information 和隨機猜選項的準確率最接近，也可以說是學到最少的資訊。

而 Random Forest 預測準確率又更高一些，我們相信透過 BLEU 以外的準則（例如 cosine similarity）來衡量 predicted output 與 options 之間的相似程度，可能會有更高的準確率。

DNN 模型的準確率能達到 0.3 ，然而 DNN 並沒有考慮影片 frame 或選項文字間的順序關係，也許使用如 LSTM 的模型來預測會有更好的結果。

Sequence to sequence with caption information 有最高的準確率，顯示模型在加入選項的資訊之後，有較多的能力學習到用一句話精確地描述影片，進一步比較 predicted output 和 options 之間的相似性能幫我們選到較為正確的答案。

六、Conclusion :

本次報告共使用了三種模型，分別對應到三種定義問題的方式： (1) DNN 用來實作選項的 ranking ； (2) Random Forest 做選項主詞、動詞、受詞的分類； (3) Sequence to Sequence 產生描述影片的一句話。我們發現在我們所提出的模型中，Sequence to Sequence 為最能夠精確解決此問題的作法，而該 model 也在 Kaggle 分數上有最高的預測準確率。

七、Reference：

[1] Rafael A. Rivera-Soto, Juanita Ordo´nez. Sequence to Sequence Models for Generating Video Captions. 2017. Stanford.

[2] Y Guo, B Yao, Y Liu. Sequence to Sequence Model for Video Captioning. 2017. Stanford.

[3] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3156–3164, 2015.

[4] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In Thirtieth AAAI Conference on Artificial Intelligence.

[5] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image Captioning with Semantic Attention. 2016 CVPR.

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation.

[7] A ten-minute introduction to sequence-to-sequence learning in Keras https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html