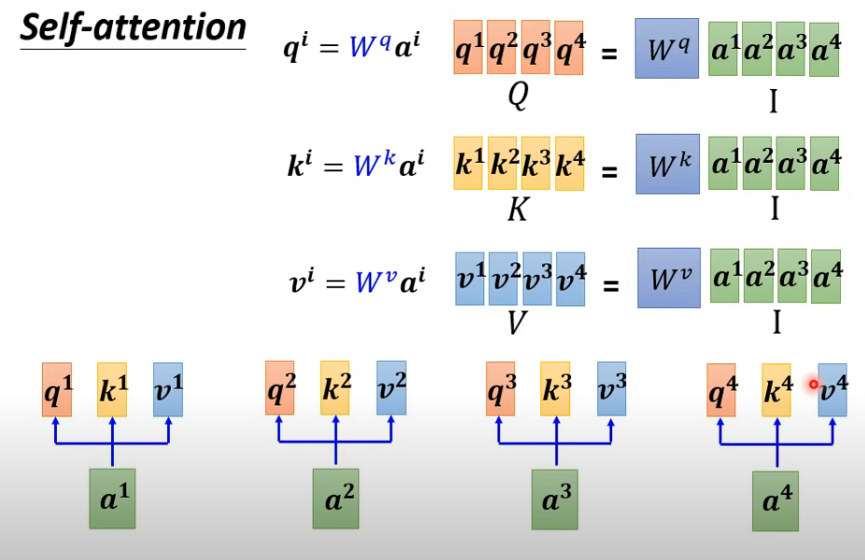
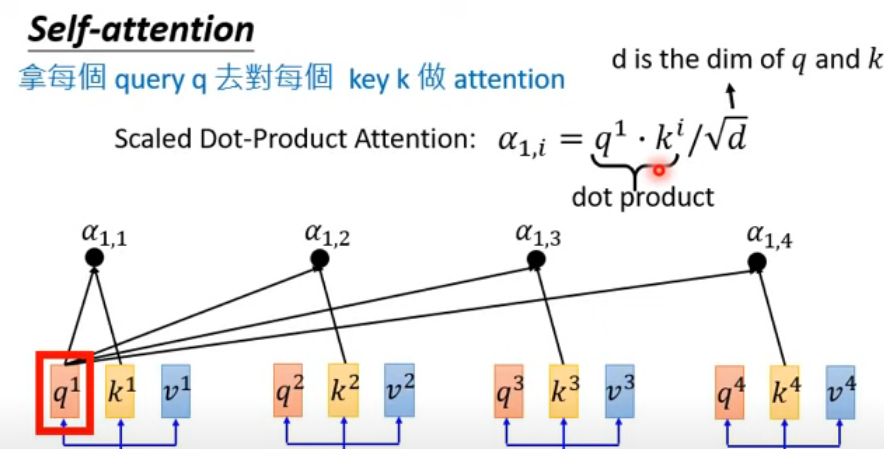
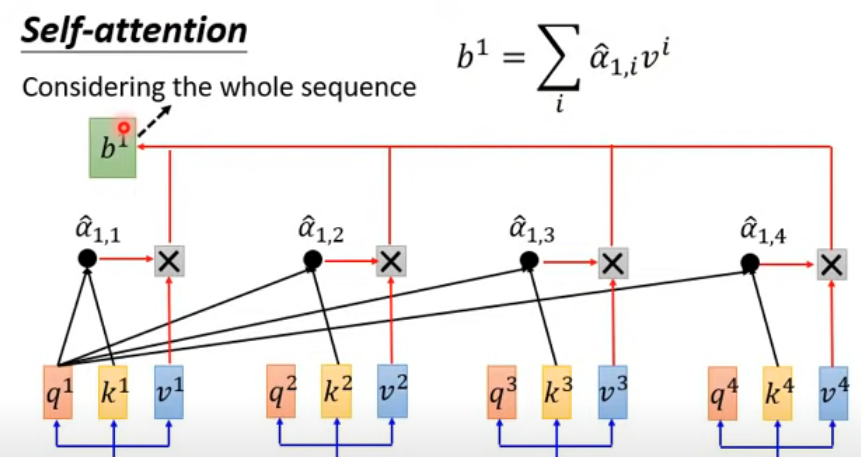
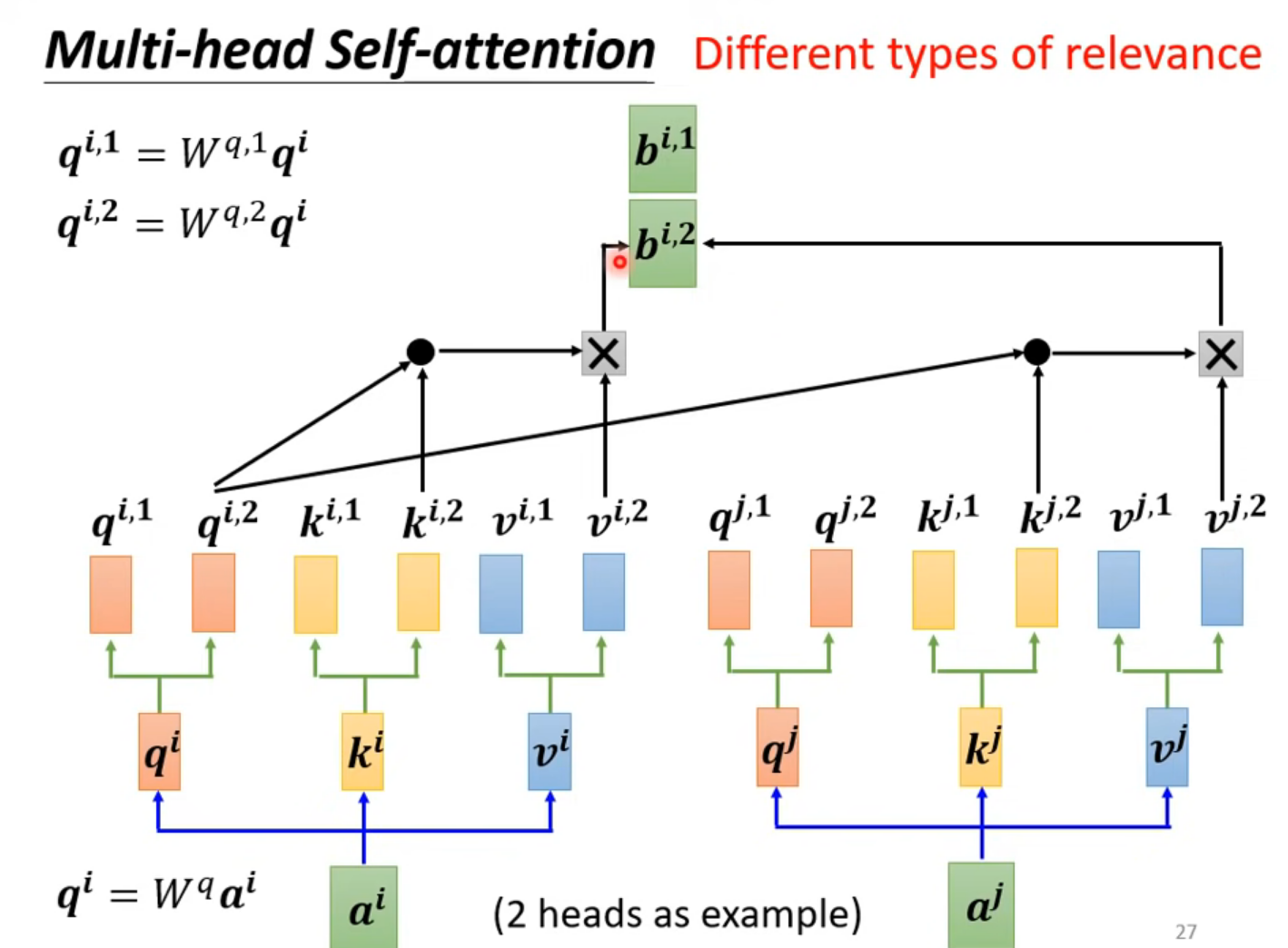
1. Attention mechanism:

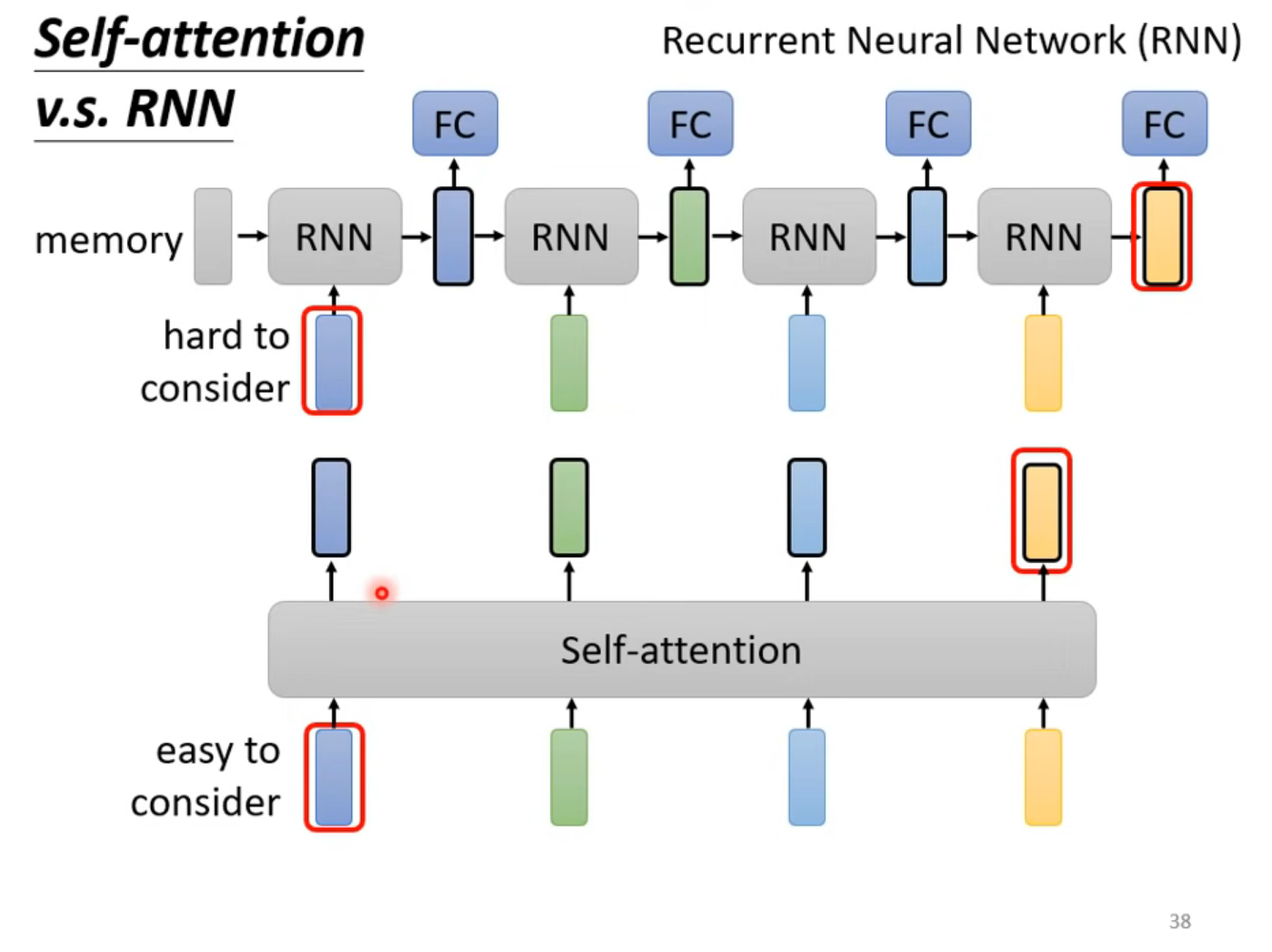
假如輸入是一段句子，在self attention中，將輸入進行input embedding後，分別乘上三個不同矩陣，成為q, k, v三種向量(如下圖)。

接著用q, k向量計算出alpha (如下圖)。

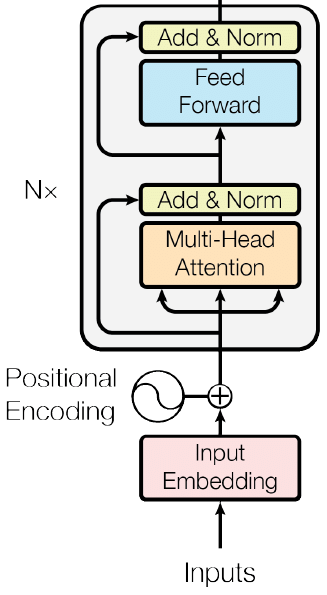
用softmax將alpha轉為alpha-hat後，利用alpha-hat與向量v計算出輸出(如下圖)。

但有時因為需要取得不只一種相關性，所以需要multihead(如下圖)。



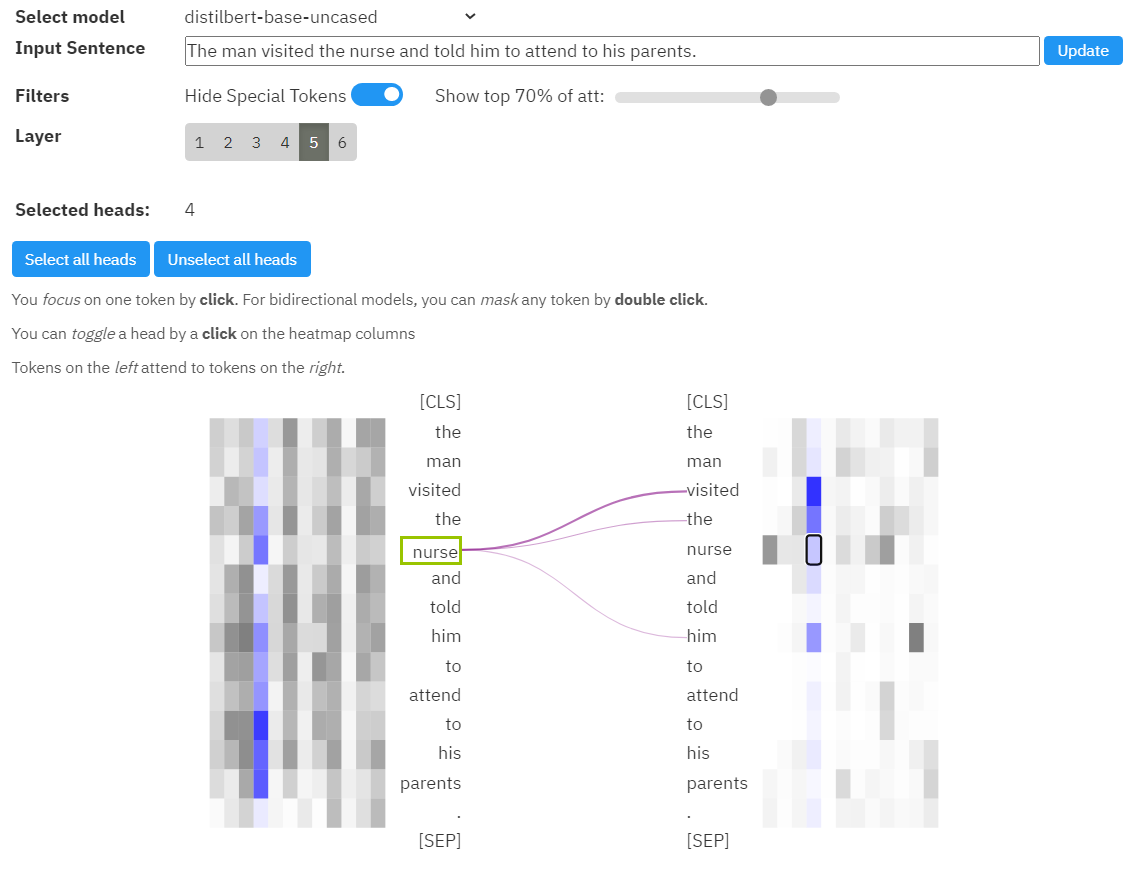
而跟RNN比較時，會發現self attention有不少優勢，包括它能輕鬆考慮到距離較遠的輸入，畢竟在self attention中，距離遠近不會有差別，但在RNN中，距離越遠的輸入就越難考慮到；此外，self attention中，可以運用矩陣進行資料平行運算，但在RNN中不行，因此耗時較久。

在Transformer Encoder中就是使用了self attention(如下圖)。

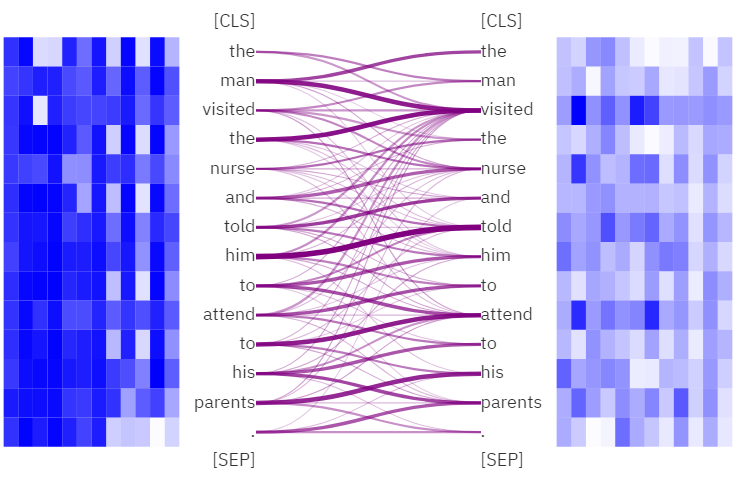


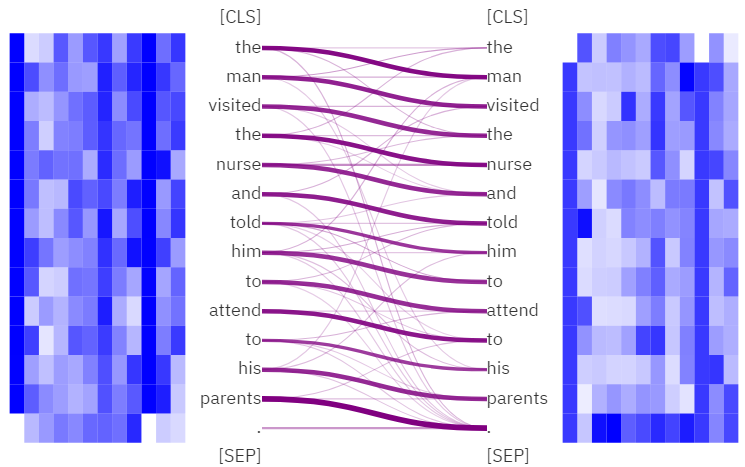
而DistilBERT是較為簡易的BERT，保留了97%的語言理解能力，但同時減少空間大小40%且加速60%。它是一種深度雙向、無監督、且僅使用純文字進行Pretrain的模型，它疊加多層Transformer Encoder，可以更好地理解語意，並在各NLP項目中被不斷使用。

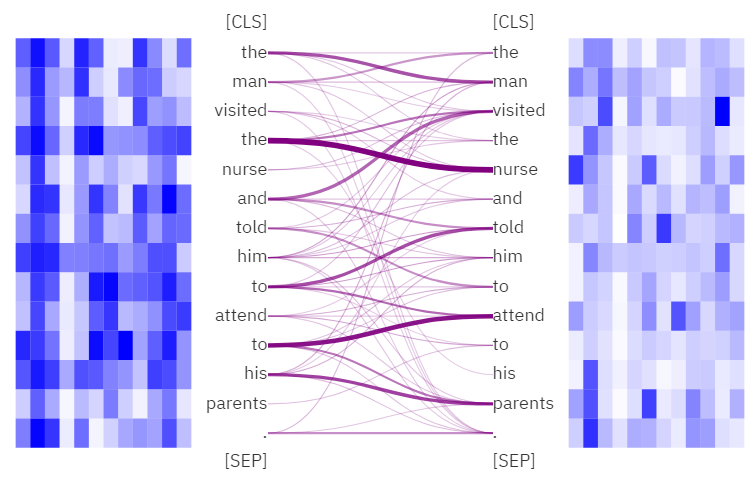
在exBERT中，就能很輕鬆地看到各個attention的關聯，此外，它能夠讓使用者自由選擇layer, heads, tokens等等(如下圖)。

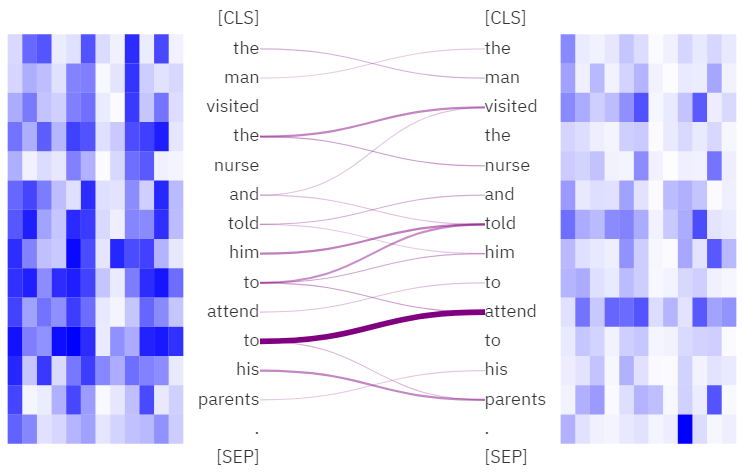


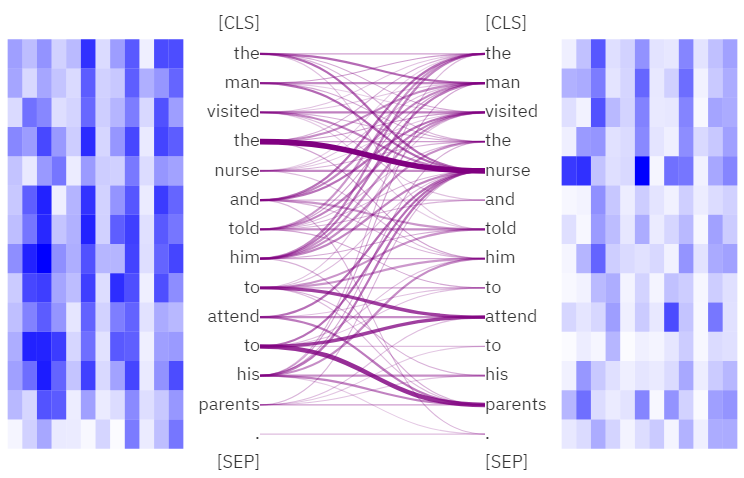
譬如選擇不同layer可以看到各layer間的差異(如下)。

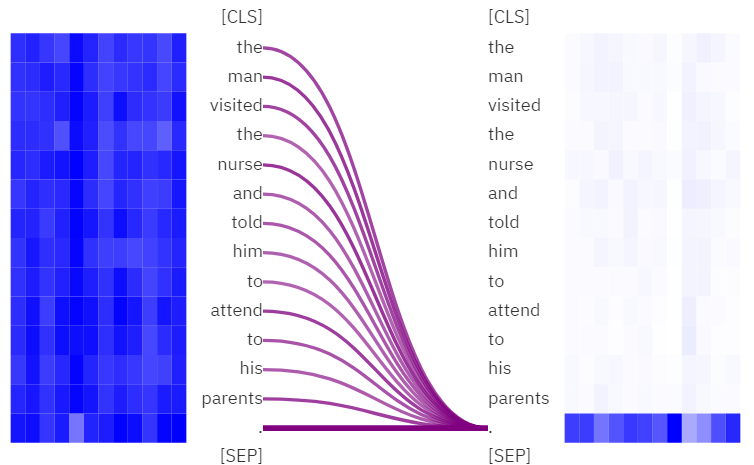
Layer1:

Layer2:

Layer3:

Layer4:

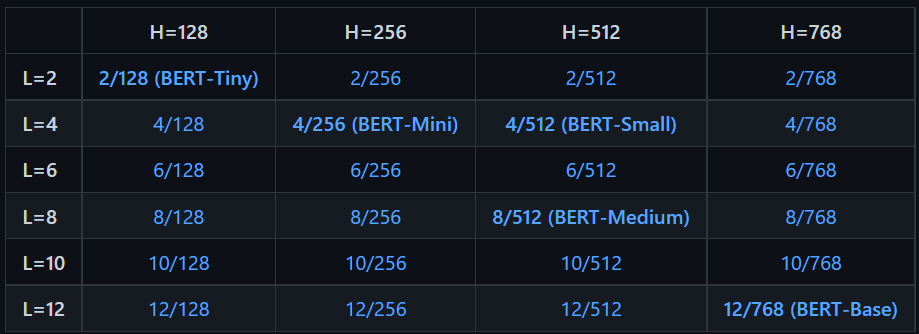
Layer5:

Layer6:

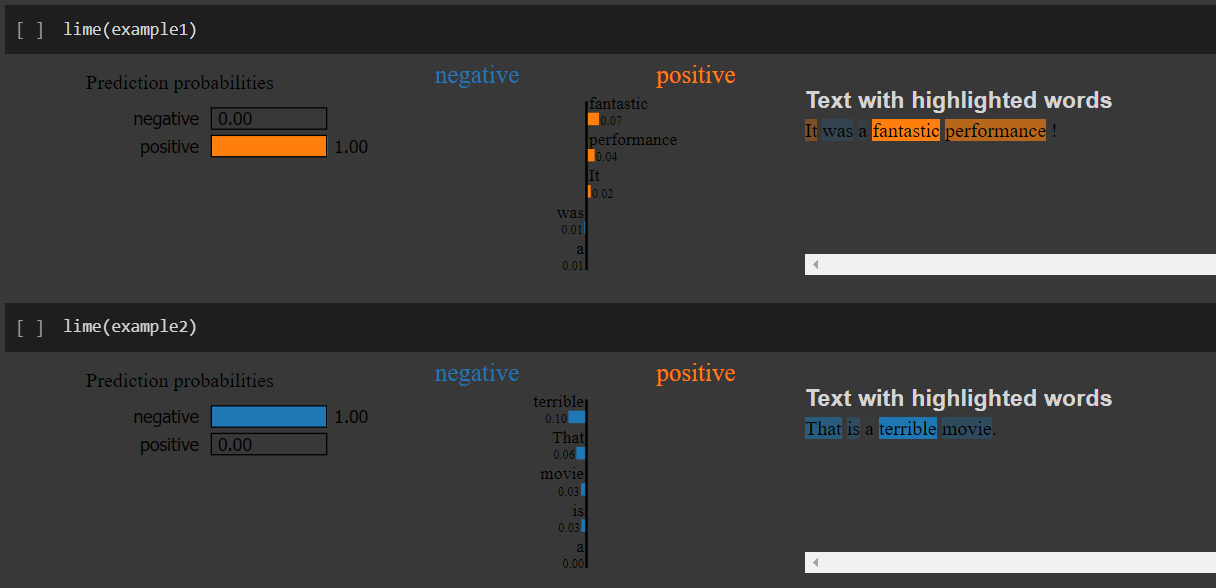
1. Compare 2 sentiment classification models
   1. TA\_model1(distilbert-base-uncased):

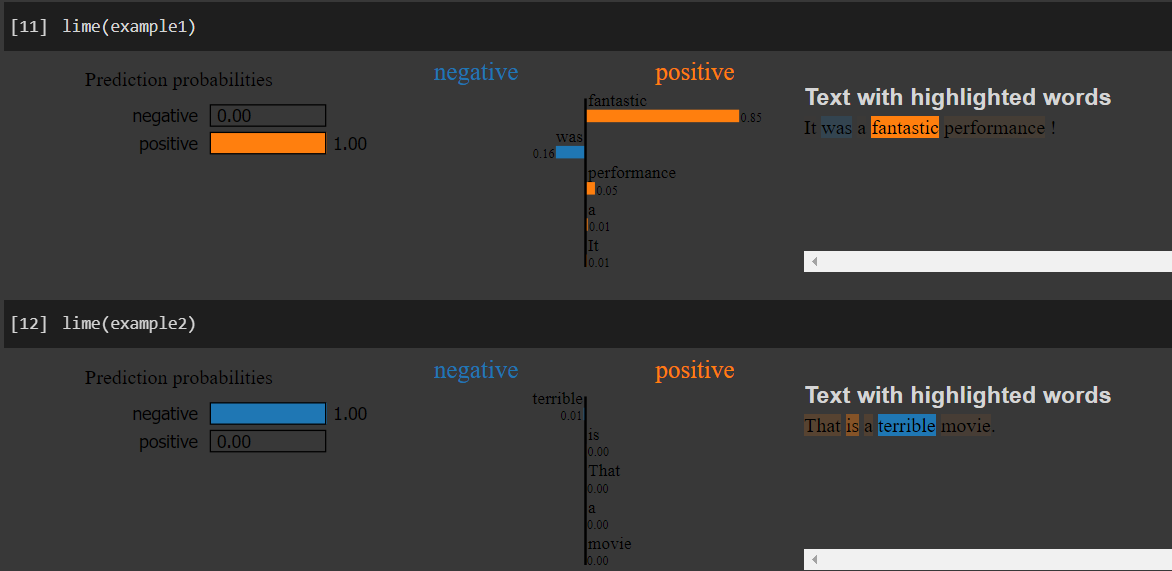
DistilBERT是較為簡易的BERT，保留了97%的語言理解能力，但同時減少空間大小40%且加速60%，使用distill technique，從原本的BERT中蒸餾成較小的模型。

* 1. TA\_model2(prajjwal1/bert-small):

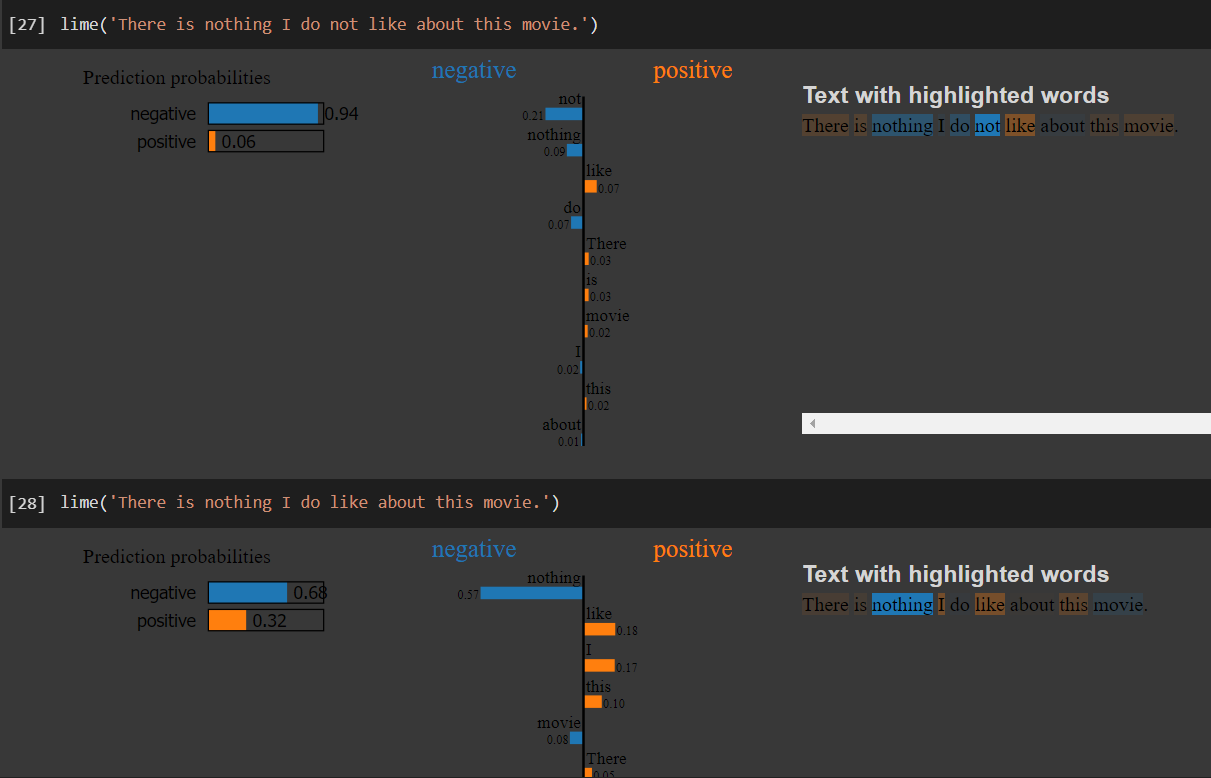
BERT-small與BERT-base的差距是他們的model大小差異， BERT-base有12個attention header、768個hidden layer，而BERT-small只有4個attention header、512個hidden layer(如下圖)；在GLUE score方面，BERT-small比BERT-base差了一點，不過這些較小的model被訓練出來的用意就是為了在較少運算資源時也能使用，而模型的表現只是一個tradeoff而已。

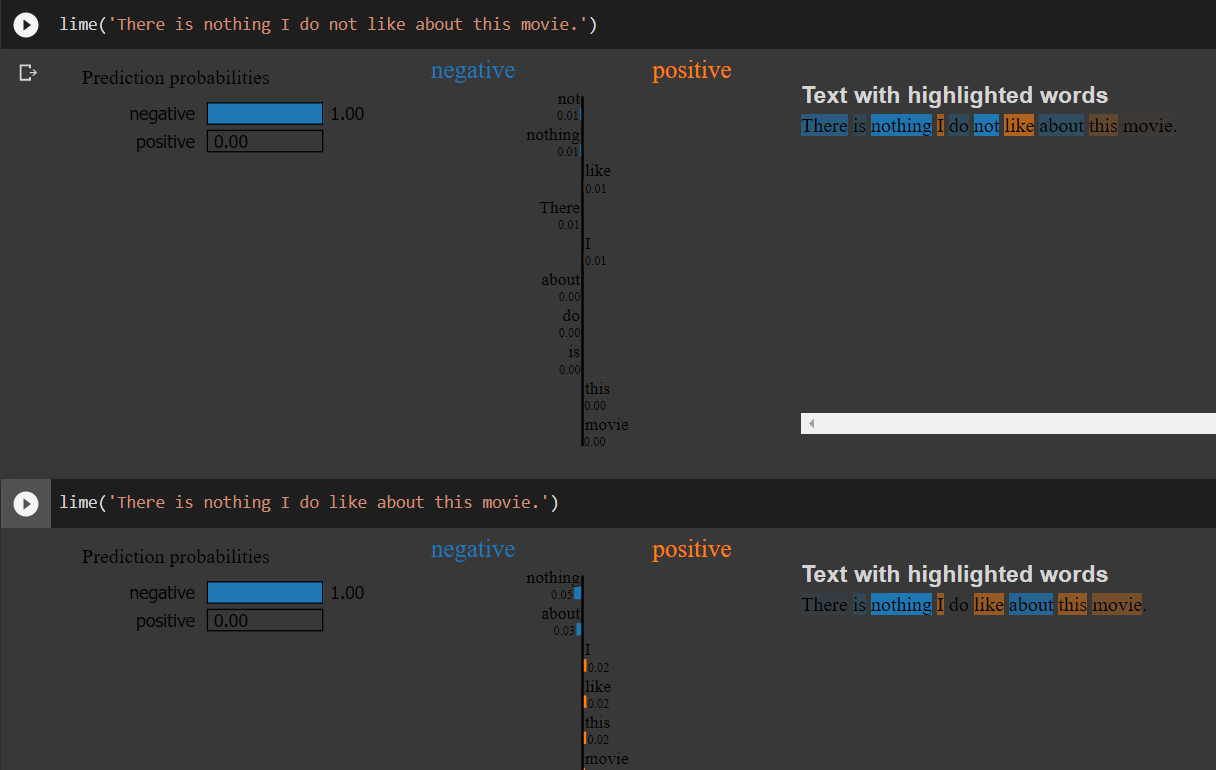
* 1. 使用LIME比較兩個model表現差異
     1. 使用例句('It was a fantastic performance !', 'That is a terrible movie.')

distilbert-base-uncased:

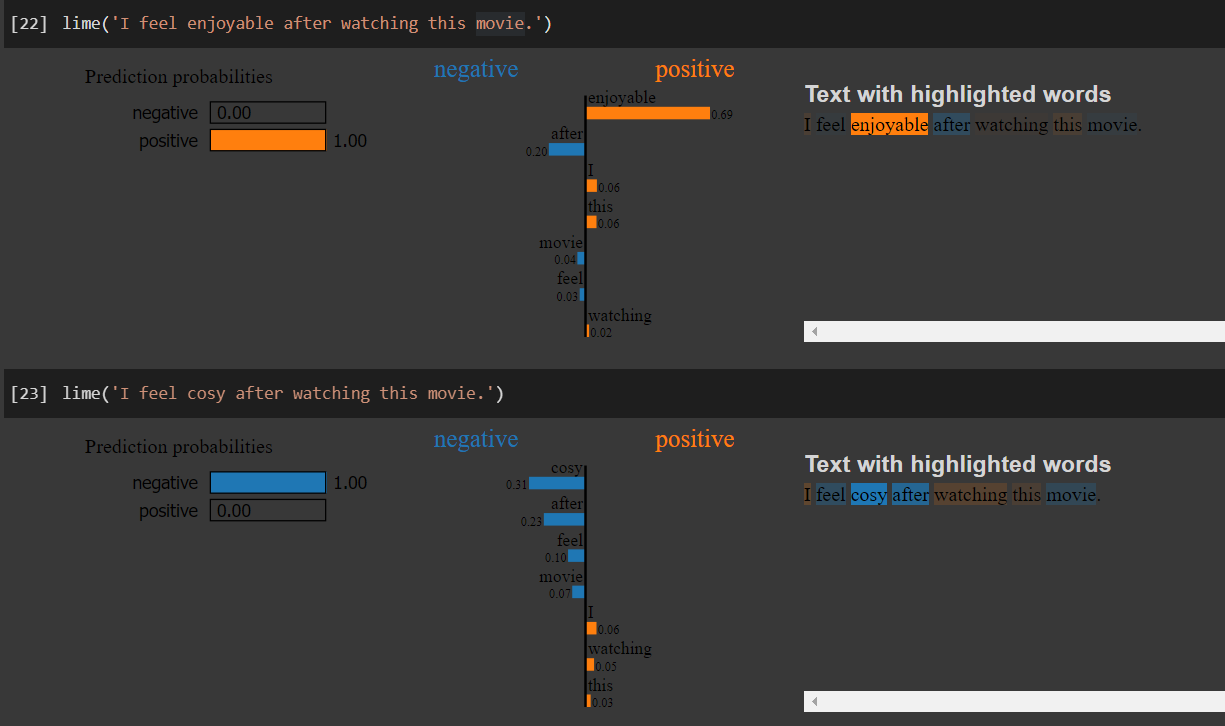
prajjwal1/bert-small:

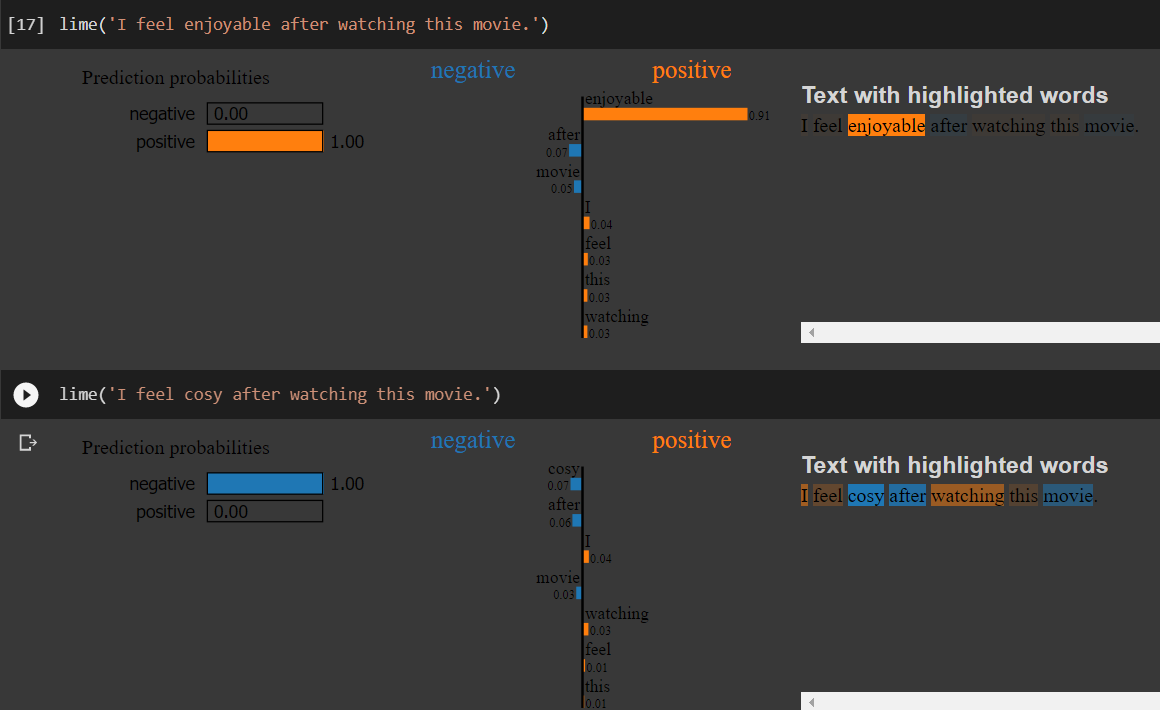
* + 1. 使用句子('There is nothing I do not like about this movie.', 'There is nothing I do like about this movie.')

distilbert-base-uncased:

prajjwal1/bert-small:

* + 1. 使用句子('I feel enjoyable after watching this movie.', 'I feel cosy after watching this movie.')

distilbert-base-uncased:

prajjwal1/bert-small:

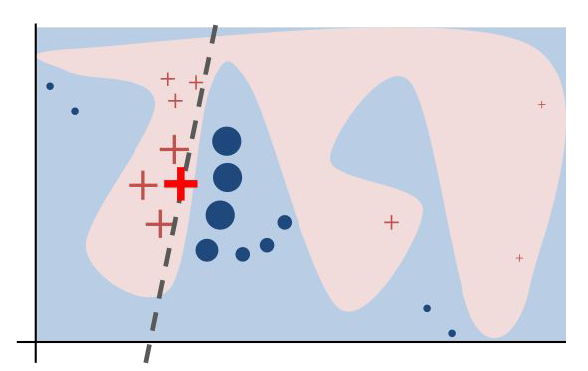
* + 1. 使用句子("I recommend this movie because of its plot.", "I do not recommend this movie because of its plot.")

distilbert-base-uncased:

prajjwal1/bert-small:

可以看到，這兩個model在判斷句子是negative或positive方面上差異不大，不過在句子中的單一token上，兩個model顯示出的prediction probability不太相同，平均上來說，BERT-small的probability較DistilBERT的小，不過也不排除是觀察例子過少所造成。

1. LIME & SHAP
   1. LIME（Local Interpretable Model-agnostic Explanations）

LIME是在"Why Should I Trust You?"論文中第一次被提出，他有著Model agnosticism的特性，代表著能夠被使用於任何視為black box的supervised learning model中；也有著Local explanations的特性，代表他局部的忠實性。假如利用兩個特徵x1, x2建立了一個複雜但準確的模型，要解釋個體為何屬於哪一區域是很困難的，因此LIME就在該個體的附近建立一個線性或是簡單的預測模型，在觀察某個體時，在該個體附近的預測準確度與原來的模型一樣，但在離該個體較遠的區域預測準確度就會大幅下降，就是所謂的局部忠實性(如下圖)。

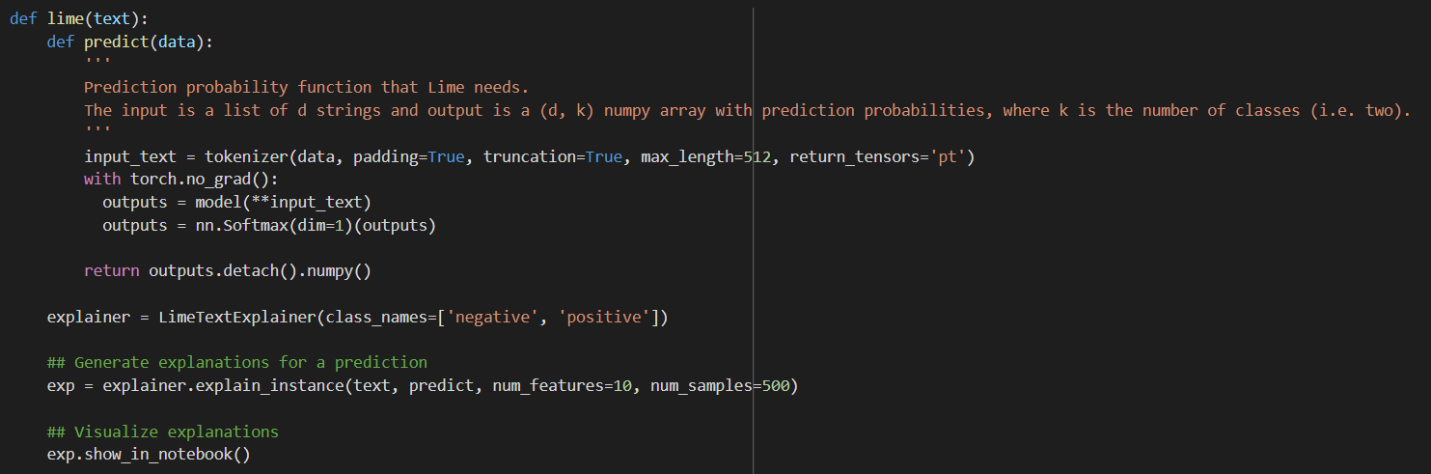
Step1. 在個體附近隨機產生pertubations，當作一組新的input data(x\_lime)

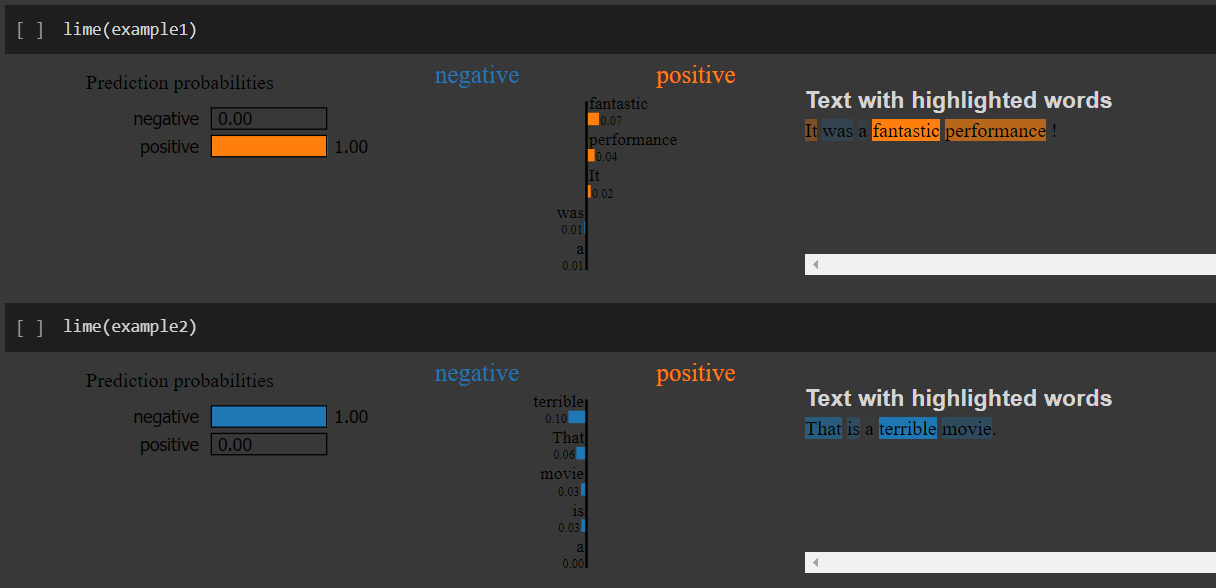
Step2. 用原本的預測模型預估上一步中的x\_lime，產生新的預測資料(y\_lime)

Step3. 計算個體與各個pertubation的距離(Euclidean distance or cosine distance)，並將距離用kernel function轉為0~1之間的weight

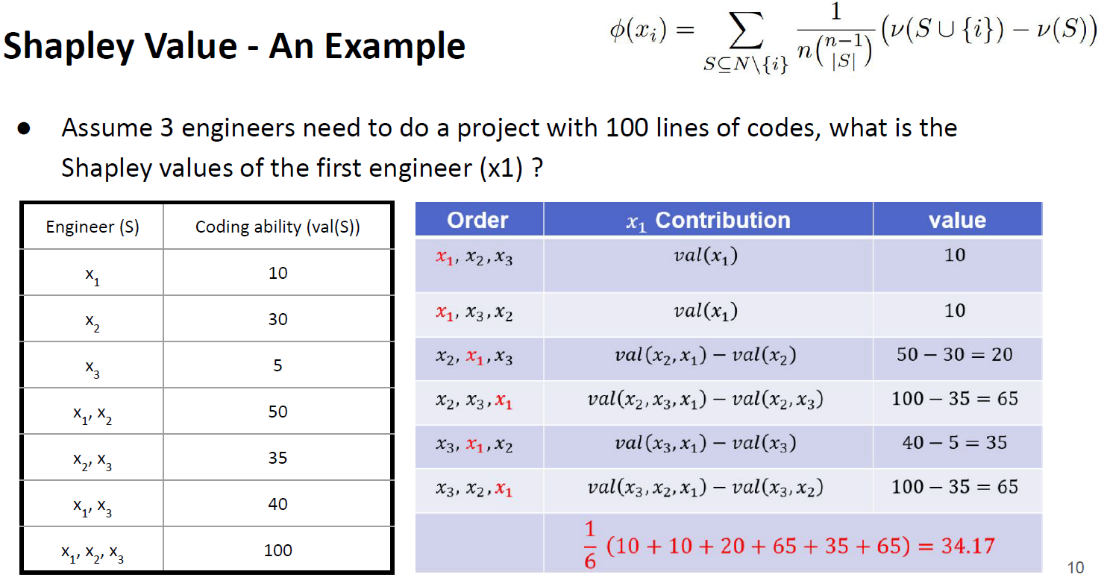
Step4. 用x\_lime, y\_lime, weight訓練一個新的簡易模型，它的預估準確度只會在個體附近與原本的模型一樣，距離過遠的準確度會逐漸下降。

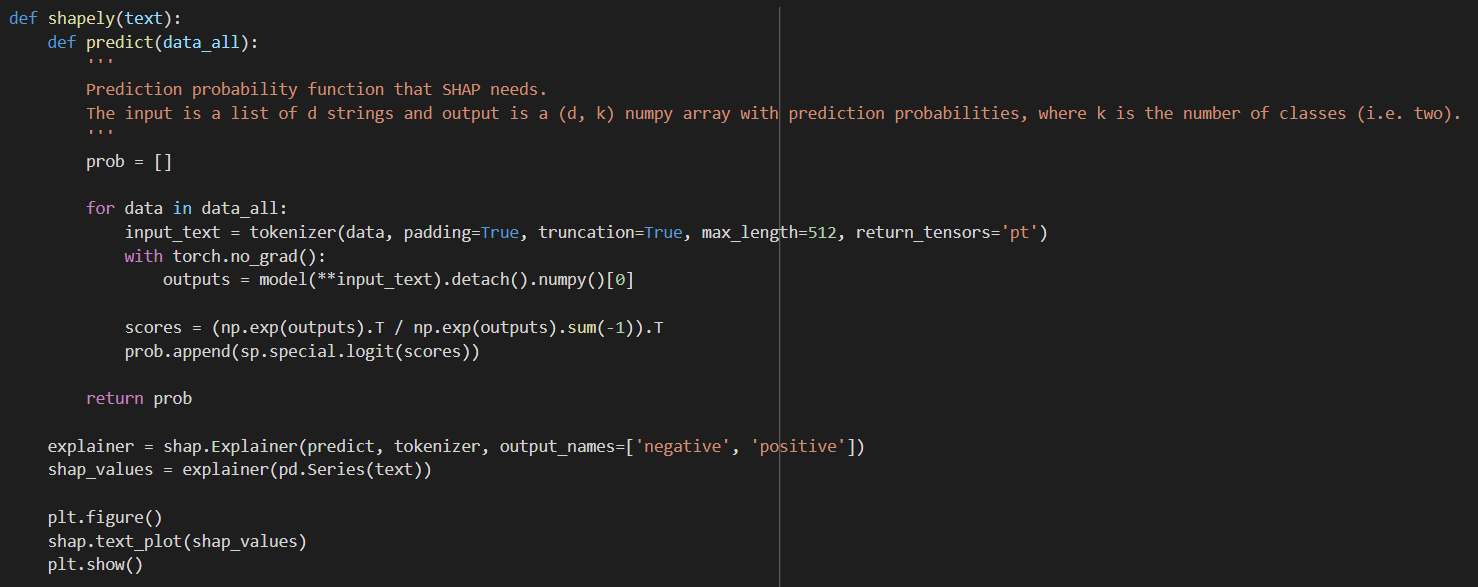
而我們也可以使用lime package輕易使用lime來解釋model，這次作業中就用lime解釋了sentiment analysis model(程式碼如下圖)。



使用lime後，可以看到每個token所佔的影響比例大小，下圖中就可看到fantastic在positive中佔了很大部分的影響，而terrible在negative中佔了很大部分的影響。

* 1. SHAP ( Shapley Additive Explanations )

在SHAP中會用到Shapley value，它是基於合作賽局理論提出來的解決方案，求出每人一個value來代表每個人所做的貢獻。本次作業中的講義就很清楚地以例子求出Shapley value(如下圖)。

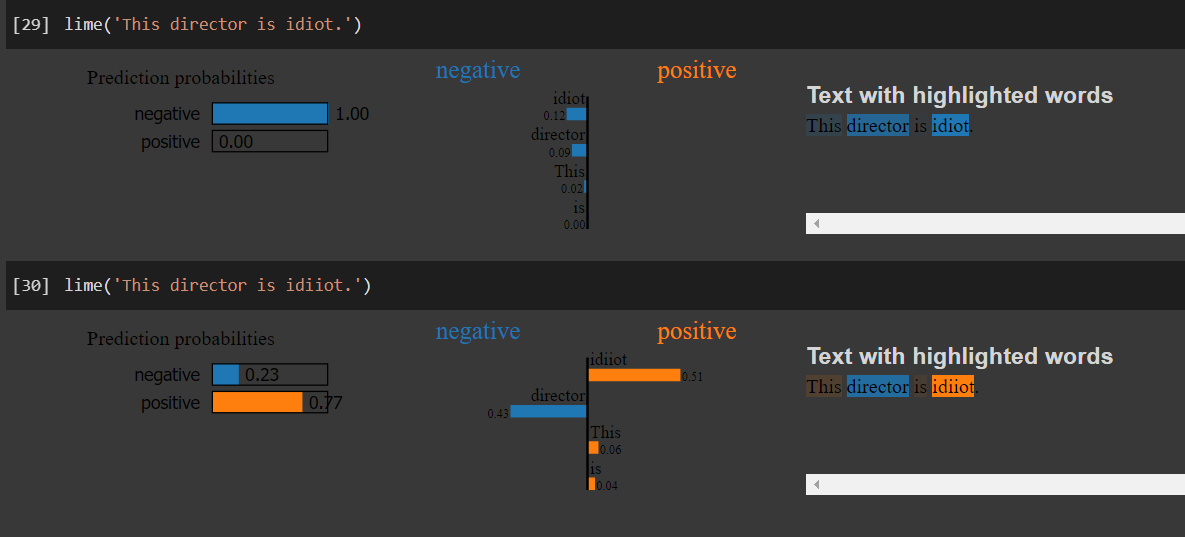
從Shapley value可以看出每個特徵的相對貢獻，因此也能用來解釋sentiment analysis model(程式碼如下圖)。

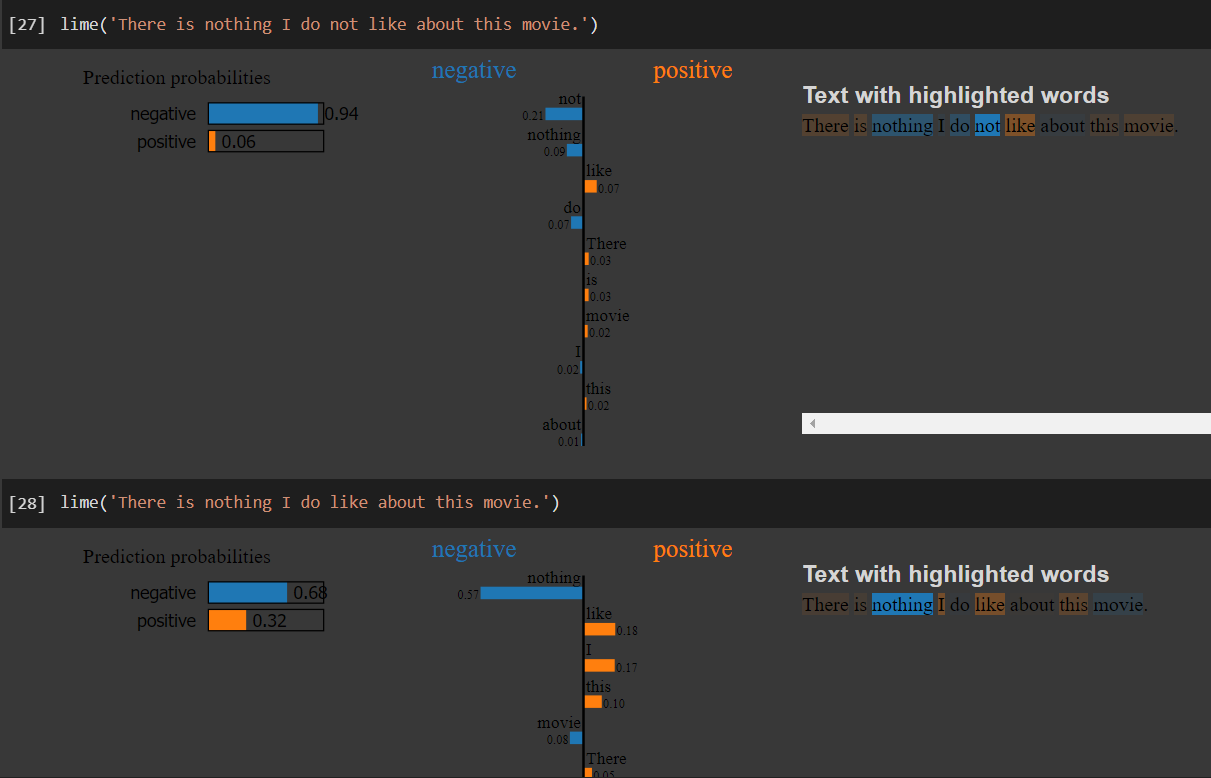
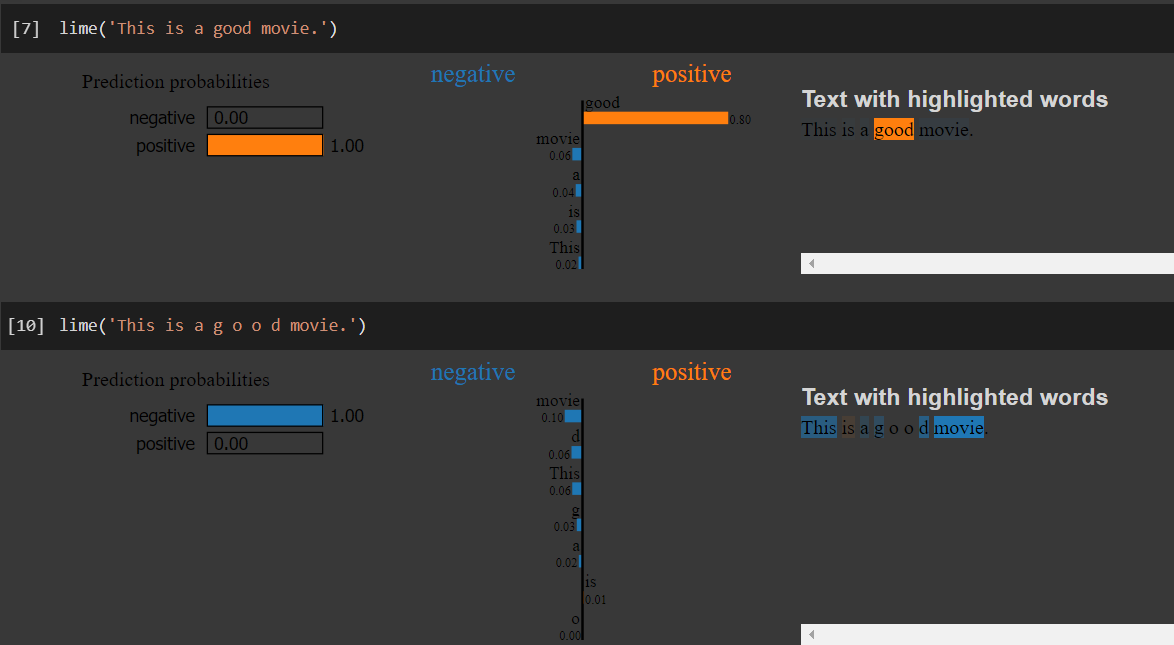
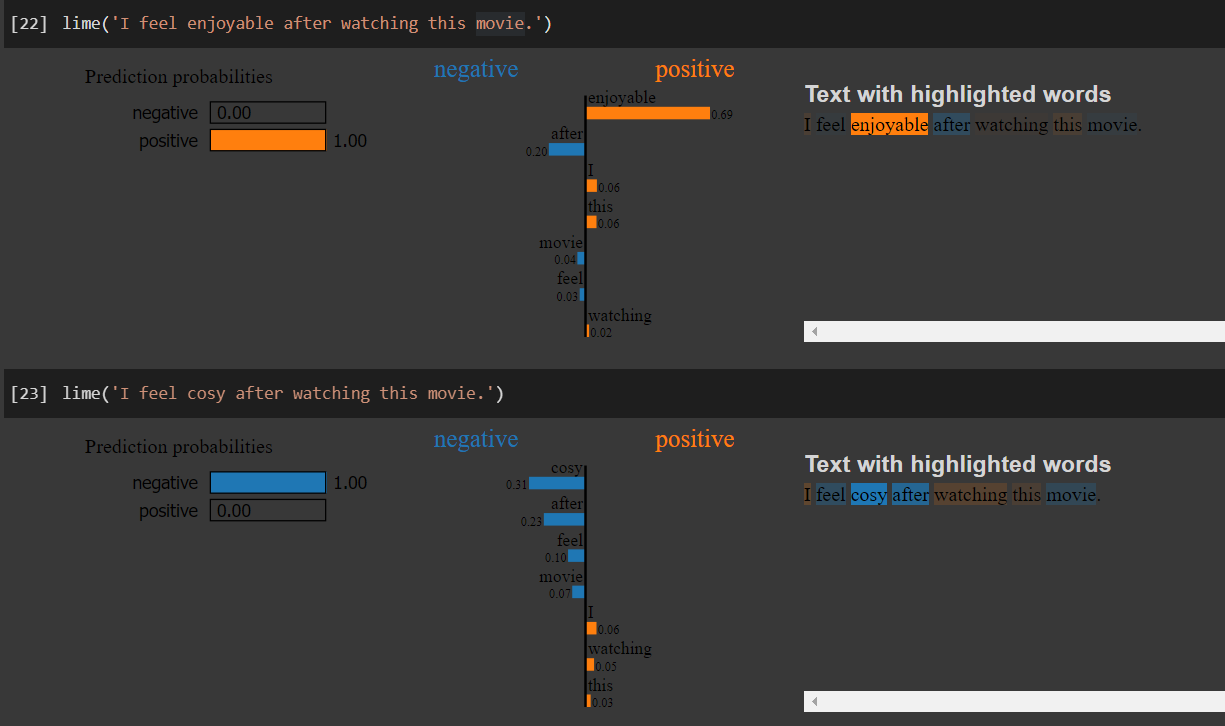
使用SHAP後，可以看到每個token所佔的影響比例大小，下圖中就可看到fantastic在positive中佔了很大部分的影響，而terrible在negative中佔了很大部分的影響。

1. Attacks in NLP

Adversarial attack在AI許多方面都有出現及研究，無論是Speech processing、Image classification或NLP中都有，攻擊方與防守方的技巧也都在不斷精進。在Speech processing中，加入一些雜訊就能讓model認為一段合成語音其實事人說話的聲音；Image classification中，加入肉眼不可見的雜訊就能讓model辨識錯誤，甚至是100%確定那個image就是他所錯誤辨認的label；在Question answering中，讓model無論是甚麼input，都只輸出”to kill American people”；在這次作業的sentiment analysis中也能看到這些model正常運作時performance都不錯，但面對adversarial attack都太脆弱了。

在使用助教提供的distilbert-base-uncased配合LIME解釋後可以看到，下面四個例子中都顯示了model的判斷錯誤。

* 1. idiot的拼音錯誤變成idiiot，導致句子判斷成positive。

* 1. model不太能判斷英文雙重否定句，第一句”There is nothing I do not like about movie”，其實是我喜歡這個電影的每個部份，但model判斷卻是negative。
  2. 將good拆開成g o o d後判斷成negative。
  3. Enjoyable與cosy算是同義詞，兩者皆是正面詞彙，但model確判斷cosy為negative。