**Spring 2022 Introduction to Artificial Intelligence**

**Report of Homework #4**

Student name: 劉子齊 Jonathan

Student ID: 0716304

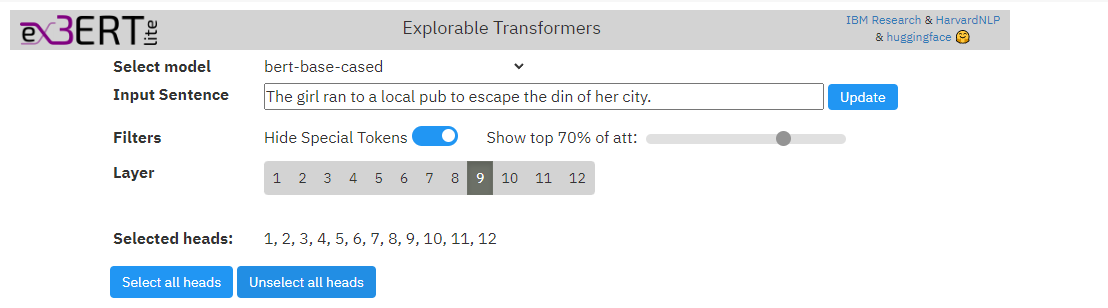
**Part 1: Attention Mechanism**

在實作作業二時，我自己碰到最大的問題就是不知道自己到底做出來的是甚麼東西，但透過Explainable AI，我的問題得以得到解決，而在眾多Explainable AI 的工具中，或者在眾多API中，我認為exBERT無疑是一時之選。

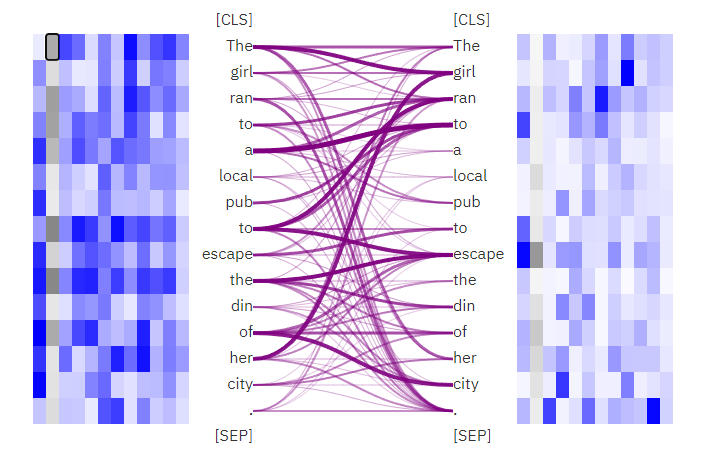
相較於其他解決方案，exBERT帶有互動式的介面，這介面不單單只是讓整體操作更為方便，更是讓我得以用更簡單的方式深入了解由 Transformers 模型形成的上下文表示的強大含意。又因為這些模型往往是由一系列人工智慧演算法來建構的，所以準確分析店倒在這個過程中究竟學到了什麼，好讓我們發現任何歸納偏差是非常重要的，而exBERT 的互動式介面更是強化了這一點。

另外，再深入研究時，我發現exBERT是以 Google 的語言模型 BERT命名，但需要注意的是，任何 Transformer 模型和語言資料庫都可以應用於 exBERT 上的任何領域或語言，因此，我們才能在這次作業中，將我們的模型套用於exBERT中。

對於我們提供的語句中的每個token，exBERT會根據我們所選擇的參數將Attention視覺化。在Attention View中，我們可以自由的更改圖層數量、選擇head並選擇我們想前幾%的attention，如下圖:



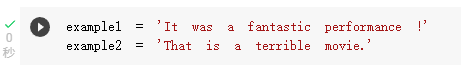
除此之外，我們可以根據我們所想要看到的資訊，將相對應的token做屏蔽，並且可以在整個語句中搜索特定token，以在顯示最高相似度匹配的視圖表中提供結果，如下圖，讓我們在研究時可以更加地深入和detail。



總結而言，我認為exBERT結合了兩大優勢，首先他將靜態分析的數據以更動態且互動式的方式展現出來，並且基於這個優勢，他讓使用者可以以一個更直覺的方式來觀察不管是在不同語句下的Attention或是不同模型所造成之差異。有著這兩大優勢，我認為在對於我一開始提到對於自己NLP模型有些困惑的問題提供了很大的改善。

**Part 2: Comparison of the two models**

在這個部分中，我使用的是助教所提供的 TA\_model\_1和TA\_model\_2。在下方LIME和SHAP的結果中，我使用的都是同樣的兩筆測資，分別如下:

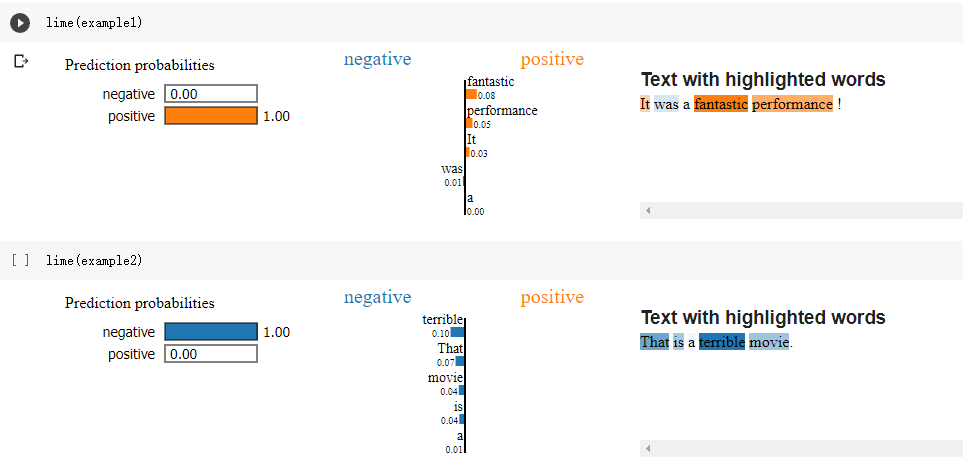


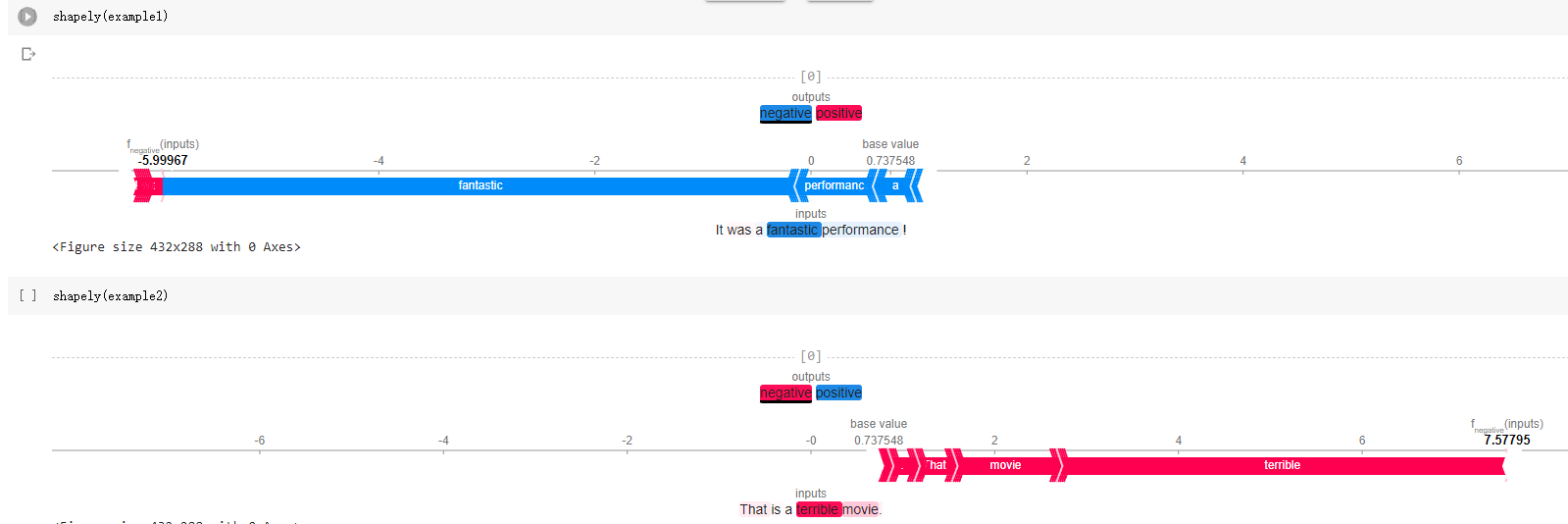
從下方的結果，可以發現在第一筆測資中TA\_model\_2在加權上似乎更重，不管是從 “was”或是其他較為中性或是正向的詞彙的便是結果中都可以發覺這件事。此外，藉由第二筆測資，我還發覺到一個現象，在TA\_model\_1的結果中，這個模型似乎一但偵測到一個負面的詞彙，就會把其餘的詞彙便認為負面的詞彙。

以第二筆測資為例，當第一個模型看到＂terrible＂時，似乎就偏好將其他的字彙辨別為具有負面意思的詞彙，會做這樣的假設是因為在第一筆測資中，TA\_model\_1並不存在這樣的情形，而是在句子中，有出現負面詞彙時，才會有這樣的反應與結果，在我測試其他句子時，也會有同樣的反映。

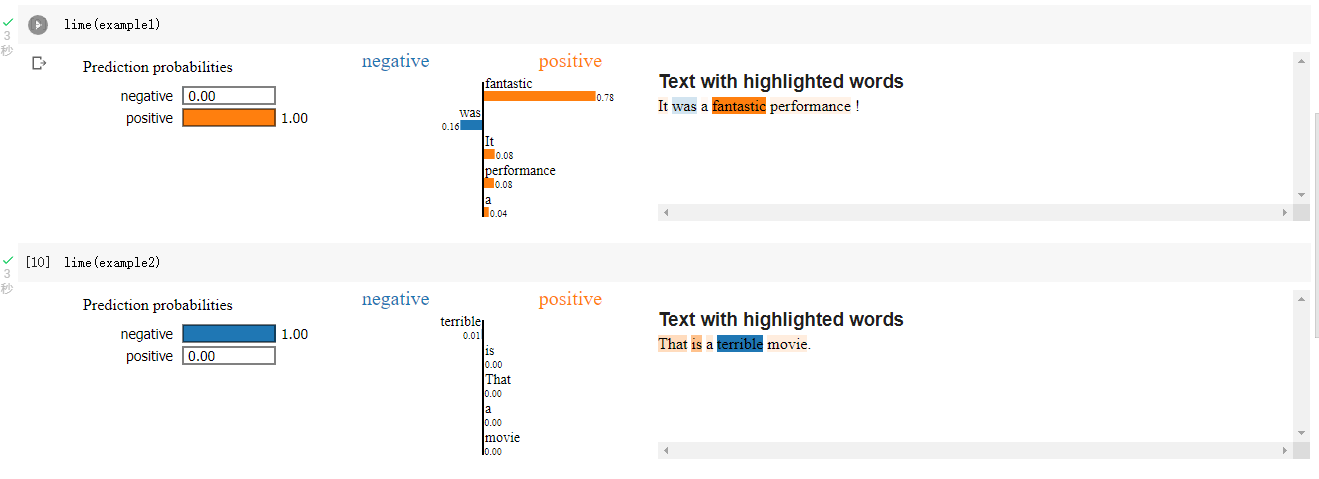
總結而言，我認為前面提到有關TA\_model\_1會被負面詞彙影響的狀況，在辨別語句意思上會有很大的影響，甚至可能導致結果有所偏差。因此，TA\_model\_1 和TA\_model\_2比較之下，我認為TA\_model\_2有較好的表現。

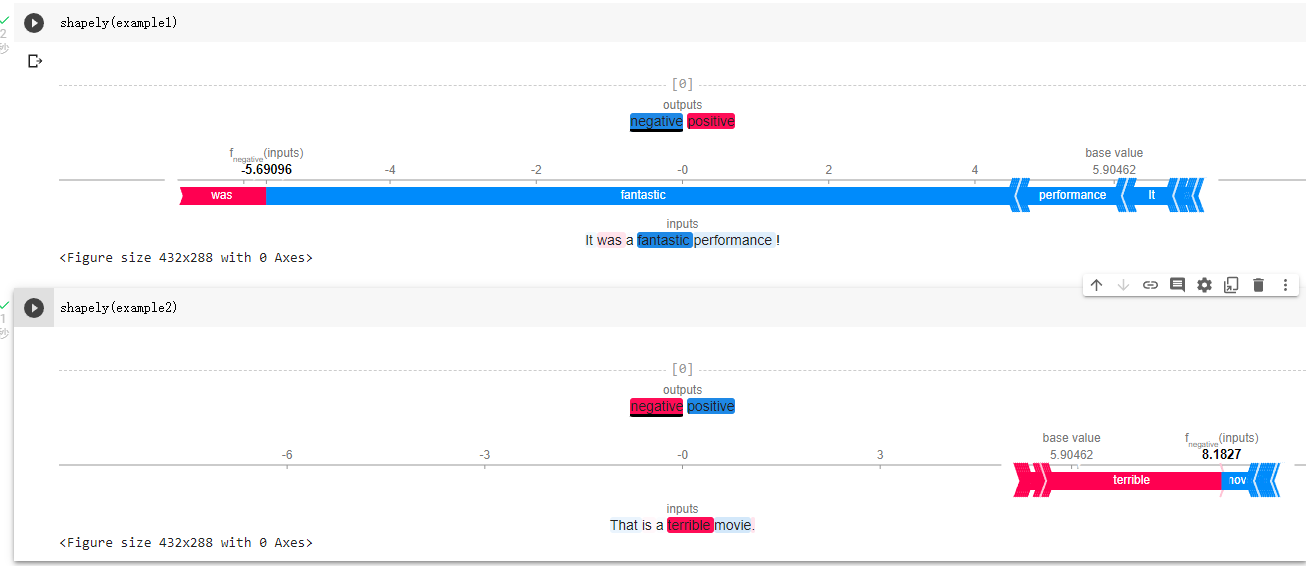
LIME & SHAP Result of TA\_model\_1





LIME & SHAP Result of TA\_model\_2





**Part 3: Comparison of LIME and SHAP**

在真正開始比較前，我想先談談兩者所具有的功用，兩者如下:

* LIME 會在本地創建一個有關我們希望了解的語句之代理模型，並且將整體之結果顯示出來供使用者作參考。
* Shapley value (SHAP)會將整體之預測結果根據不同字詞之屬性及其權重於結果中做標記。

Result of LIME Result of SHAP

但要真正獲得每個不同字彙之Shapley value，就必須針對省略字彙屬性之方式做出一些決定，這就是得出這些Shapley value的過程。而不免俗地，模型最後解讀的結果，可能會隨著這些決定有所不同。例如，今天如果我省略了一個字詞屬性，我要針對其餘的value做一些像是平均之類的處理嗎？或者在做這些決定時，我是否有一些標準呢? 這些都是有可能會是影響結果的潛在原因。

簡而言之，SHAP會以附加的方式告訴我們它是如何獲得最終結果分數的，但是對於我們省略字詞的策略，也帶有一些可能會影響結果之選擇。而LIME只是簡單地告訴我們，在這個句子中，我們模型感興趣的字詞周圍最重要的屬性是什麼。

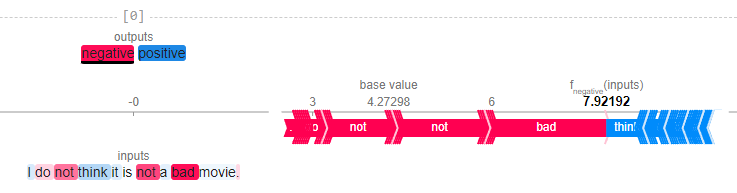
總歸而言，我認為兩者沒有好壞，但對我而言，我會更喜歡使用SHAP，因為相較於LIME，我認為SHAP報出的結果更加的直覺，我不需要眼睛動來動去，左右兩個圖表相互對照，我只需透過單一表格，就可以獲取所有我需要的資訊了。因此，在SHAP與LIME比較之下，我會更加偏好使用SHAP作為我的Explainer。

**Part 4: My Attacks**

在這個部分，我這邊用的是我前面有提到，我個人認為具有較好表現的TA\_model\_2來做實驗與比較。這邊我使用了多個不同的測資，試圖攻擊這個模型，想要看看他的能耐，我也得出了一些有趣的結果。

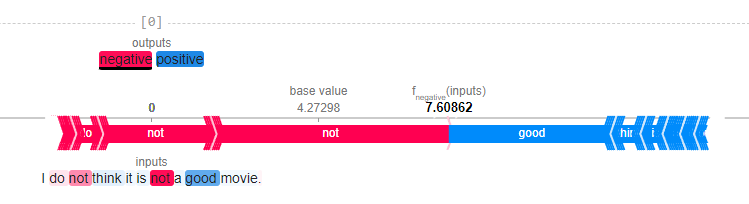
1. 'I do not think it is not a bad movie.'

這邊我先試了一些基本的負負得正的語氣，我這邊想要表示的是 ” 我不認為這不是一部糟糕的電影。” 這句話所帶有的應該使屬於比較負面的意思，而這邊模型也成功的辨認出來了，但我這邊懷疑他是不是會針對 ”not”作一些既定的解釋，因此，我接下來換了另一個方式攻擊它。



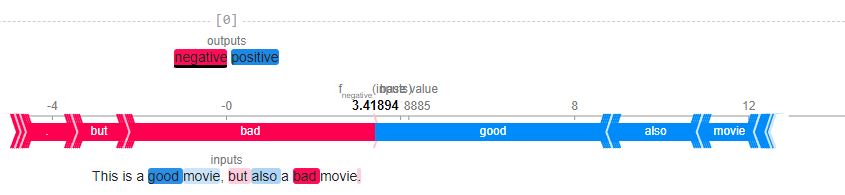
1. 'I do not think it is not a good movie.'

繼前一個攻擊，我繼續嘗試，這邊想要確定的是它是不是會針對”not”做一些既定的解釋。因此，這邊我的攻擊句為 ”我不認為這不是一部好電影。” 這句話本身應該是依據正向的話，但從下面的結果可以得知，模型將這個句子解讀為一個負面的句子。因此，這個模型會針對 ”not” 做一些既定解釋的潛在可能又更佳的確鑿了。



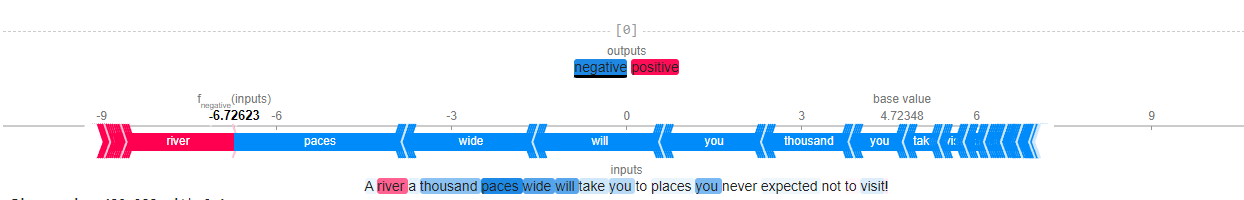
1. 'This is a good movie, but also a bad movie.'

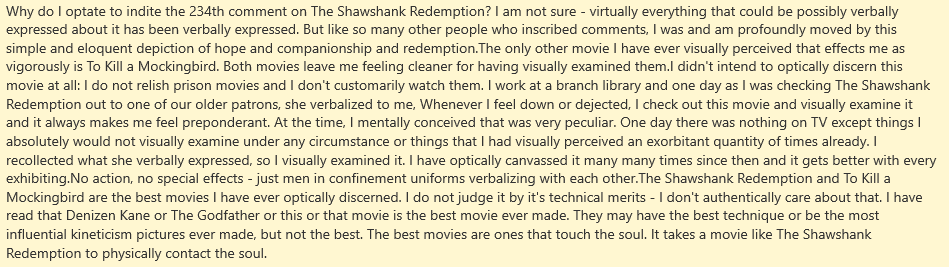
這邊我想要測試的是，矛盾句能否成功攻擊這個模型，我使用 “這是一部好電影，也是一部爛電影。” 來發動攻擊。這邊模型判斷出的結果是負面的句子。由於這是一句矛盾句，我也沒辦法確定這個句子到底是正面還是負面。但由SHAP本身會顯示之各字彙的value，我認為這個模型的判斷結果也是情有可原。因為從下方的結果可以看到，模型將”but”視為一個關鍵字彙，在現實生活中，我們如果在讚美後面加上一個反詰語氣，所帶來的往往會是一個負面的評論。因此，我認為這部分模型是判斷正確的。



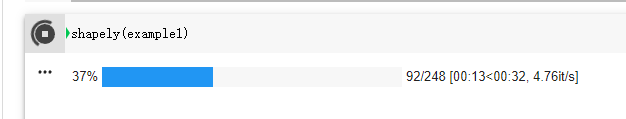
1. 'A river a thousand paces wide will take you to places you never expected not to visit!'

這邊我發動的攻擊是來自網路上的 Nonsense Generator生成的句子，我特別挑選了一個不帶有任何負面詞彙，完全是由好幾的正向詞彙所組成的句子來做攻擊。但由下方的結果可以看到，這個模型似乎認為 ”wide”, “you”, “thousand” 這些常見的詞彙是屬於負面的詞彙 XD 這些詞彙對於模型而言應該不陌生，但也許是這句話的廢話程度，讓模型直接火大，把這個句子便認為一個負面的評論。因此，我將這個攻擊是為一個成功的攻擊成功。

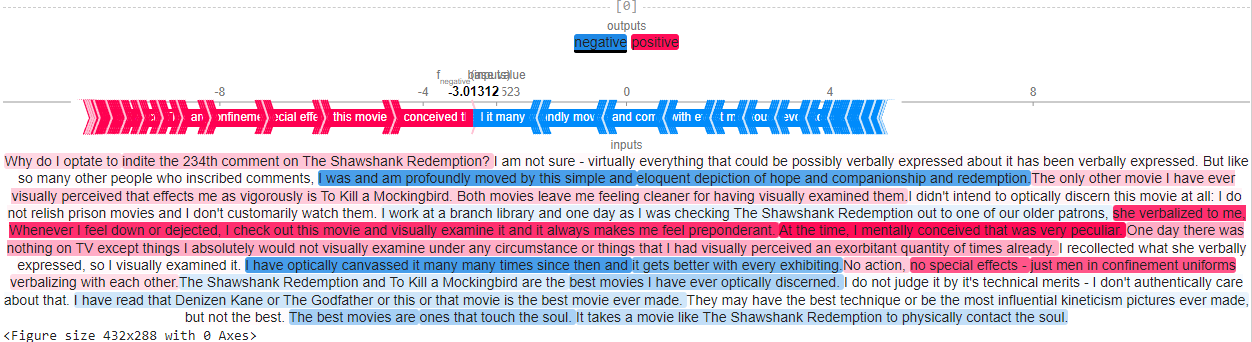


1. 

在這個攻擊中，我先是找了一則有關在 IMDb 上擁有最高評分的電影 - “刺激1995” 的評論，接著我把這個評論中的字詞作抽換，我相一些淺顯易懂的字，換成了一些較為晦澀難懂的同義詞，在把這個抽換過的評論未盡模型中，因為句子比較長的關係，它還花了一些時間做辨識。



下方是這次的模型解釋，它將這則評論解讀為負面的意思，但這則評論的作者的原意應該是正面的，作者先是闡明了自己不喜歡看監獄片，接著拿另外一部電影相比較過後，一語道破了刺激1995的過人之處，以及作者自己對這部電影的喜愛。因此，這則評論是一則正面的評論，但我想也許這個模型在解釋具有轉折的一些評論時，會因為對一些詞彙的既定解釋，如 ”not”，而使這次的結果有所偏差，我想，這就是使這次攻擊成功之主因。



針對上面的攻擊，我發現TA\_model\_2在解釋評論上的最大漏洞就是轉折語氣，或是負負得正的情況(not not)。我認為要改善這個情況的方式有兩種:

第一，因為這類型的評論、或是攻擊，是我們刻意製造的，多數時候，社會大眾在留下評論時，並不會用這麼「奇特」的語法，由此可以推知在訓練的資料中，這種類型的資料也比較少。因此，如果想要解決這個問題，也許可以在測資中加入一些比較刁或是比較「奇特」的測資。

第二，可以針對轉折詞，像是 “not”, “but”等，做一些特殊的處理，也許在不同情況下，或是不同出現次數 (因兩個連續的not可以相互抵銷)做不同的加權計算，這也許也是縮小這樣的攻擊所造成之影響的方式之一。