1. Describing your methods in detail.

使用 SpaCy 套件將句子之每個詞的 dep, pos 先標註出來後，針對 Subject, Object, Verb 的尋找分別進行以下處理。

Subject

若單詞之 dep 為 nsubj, nsubjpass, csubj, csubjpass, agent, conj，且其 pos 為 NOUN, PRON, PROPN，則將其加入 subj 列表中，視為可能的 Subject。

Object

若單詞之 dep 為 dobj, attr, pobj，則將其加入 obj 列表中，視為可能的 Object。(dependency 的選擇會在後續進行說明)

Verb

若單詞之 pos 為 VERB 或 AUX，則將其加入 verb 列表中，視為可能的 Verb。

之後，若 Dataset 的 S, V, O 欄位中與上述 subj, obj, verb 皆有重疊則預測 1，而若任一列表中與對應 S, V, O 沒有重疊則預設 0。

2. Is there any difference between your expectations and the results? Why?

預測效果與我所想的差不多。

由於我使用的方法會將任何可能的詞都加進列表，並且只要列表中的辭彙有重疊到就可以，因此在 Ground Truth 為 1 時預測正確的機率非常高，幾乎不太會出錯，而缺點則是 False Positive 的機率比較高。

在 example dataset 的測試中，雖然 false positive 有 15 個案例，但 false negative 只有三個案例，用傾向預測 Positive 的方式來提升準確率，與預期的狀況是一致的。

3. What difficulties did you encounter in this assignment? How did you solve it?

套件安裝

安裝 SpaCy 時 conda 一直出現各種奇怪的錯誤，最後直接在 conda 環境中用 pip 來安裝。

Subject, Object 找尋策略

原本在尋找 subject, object 時使用的是 token 的 pos，以 NOUN 出發去尋找可能的 subject 以及 object，結果策略可能有誤，會傾向預測為 0，預測效果不太好，因此之後改為由 dep 作為尋找 subject, object 的根據，效果才有所提升。

Dependency 選擇

改為由 dep 尋找的初期，預測效果也不是太好，因為當時要視哪些 dep 為候選詞時缺漏不少(比如 obj 中的 pobj, attr 一開始都沒有放入列表)，後來去了解各個 dep 的涵義後做進一步的篩選，新增了許多 dep 進入列表。

Dependency 進一步篩選

從 example dataset 的結果來看還是有一些 subject 容易被遺漏掉，或有些其詞不是 subject, object 卻容易造成誤判的，因此印出每個出錯的句子並觀察各個詞被預測出的 dep，以及哪個 dep 其實很常是 subject 卻沒被列入，而新增了 conj，並根據實驗結果刪除了 obj 列表中的 dative。

Position 額外篩選

由於使用前述方法後 false positive 居高不下，因此將 position 重新納入考量，單詞必須是 NOUN、PRON、PROPN 才可以列為可能的 subject (原本 object 也想用此限制，但實測後效果變差)，使預測效果再獲得些微的上升。