**Report：NLP\_HW1 成功大學\_N96134645\_許宸華**

**Running environment：Colab Python version：Colab**

**程式撰寫使用Chat gpt 4o進行協助**

**一、Which embedding model do you use? What are the pre-processing steps? What are the hyperparameter settings?**

在這次實驗中，我們使用了 **GloVe-wiki-gigaword-100** 作為預訓練的詞嵌入模型。此模型是基於維基百科和其他語料庫訓練而成，向量維度為100。

前處理步驟包括：

1. **資料清理**：將資料集中的問題按類別和子類別分類，首先清理每一行的換行符號，然後遍歷每一行，分類不同的SubCategory，再將前5個分配到"Semantic"Category，後9個分配到"Syntatic"Category。並儲存為 DataFrame 形式，以便後續操作。

**顯示結果：一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述**

1. **去除不符合格式的詞彙**：將詞依空格分開後，將模型無法預測的字詞刪除，這樣模型才能順利進行預測，但也導致有些類別的準確率不佳。

**二、What is the performance for different categories or sub-categories?**

在評估模型的準確率時，根據不同的類別進行了詳細分析。實驗結果顯示，不同類別和子類別的模型表現差異顯著，以下是部分結果：

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

* **語義類別（Semantic）**: 類似於「國家與首都」的問題，準確率非常低，我認為是因為這些詞在wiki上的出現頻率非常低，導致模型無法預測，所以常出現的family子類別表現的就很好。
* **句法類別（Syntactic）**: 類似於「動詞時態變化」的問題，準確率相對較低，評估是因為詞的變化較複雜且依賴上下文。

具體結果顯示，語義類別的成功預測的準確率約為 **80%**，而句法類別的準確率則下降至 **40%-50%**。

**三、What do you believe is the primary factor causing the accuracy differences for your approach?**

我認為準確率差異的主要因素有以下幾點：

1. **詞彙的語義關聯性**：在語義類別中，詞彙之間的關係較為明確，模型可以較準確地進行向量運算。而在句法類別中，詞彙的變化更依賴上下文，導致模型表現不佳。
2. **詞嵌入的訓練資料**：預訓練模型中的語料庫對於某些詞彙的涵蓋度較高，但有些可能完全沒有，像是國家和首都的關聯詞，在對於動詞變化等句法關係的學習也較為有限。

**四、What’s your discovery from your t-SNE visualization plots?**

通過 t-SNE 將詞嵌入向量降維至 2D，並對「family」子類別的詞彙進行可視化，我們觀察到：

1. **詞群聚集現象**：類似的詞彙會聚集在一起，例如「father」和「mother」、「brother」和「sister」，說明詞嵌入模型能夠有效地捕捉詞之間的語義關聯。
2. **類比詞之間的關係**：一些具有類比關係的詞組，如「husband」和「wife」，在 t-SNE 圖中也顯示出相對緊密的關係。

這些結果展示了模型在語義理解方面的能力，但也揭示出對於一些更細微的語義區分，模型仍存在局限。一張含有 文字, 螢幕擷取畫面, 圖表 的圖片

自動產生的描述

**五、What’s the difference in word representations if you increase the amount of training data?**

若增加訓練資料量，詞彙表示會有以下變化：

1. **表示精確度提升**：詞向量能更準確地表達詞彙的語義和句法特徵，特別是在類似詞之間的區分上，如不同詞性的同義詞等。
2. **覆蓋範圍擴大**：更多的訓練資料使得模型能涵蓋更多的專業詞彙和少見詞彙，進而提升對長尾詞的表示能力。

總體而言，增加資料量可以顯著改善模型在某些具挑戰性的子類別上的準確率。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

六、關於自訓練模型

Preprocessing：

* 將文本轉為小寫，統一格式。
* 去除標點符號，保持乾淨的詞彙集合。
* 分詞（tokenization），將每行文章劃分為單詞列表。
* 移除停用詞（如 "the", "is", "and" 等不具有重要語義的詞），減少無效資訊對模型的影響。

使用 Gensim 套件中的 Word2Vec 模型來訓練詞嵌入。

設置的參數包括：

* vector\_size=100：每個詞的嵌入向量大小為 100 維度。
* window=5：上下文窗口大小為 5，表示模型將考慮每個詞前後最多 5 個詞的上下文。
* min\_count=5：僅對出現次數大於或等於 5 次的詞進行訓練，過低頻率的詞將被忽略。
* workers=4：使用 4 個 CPU 核心進行並行訓練，以提高訓練效率。

由於要處理的檔案過大，使用\_\_iter\_\_ 方法使得 WikiCorpus 類別可以像一個迭代器一樣使用。

結果：各項精確度都略遜於預訓練的模型，而且結果十分相似，我判斷是兩個模型都是使用wiki的資料來做訓練，因此在資料集十分相似的情況下得到的結果也會近乎一樣；而關於精確度都遜於預訓練模型，我認為是因為1. 資料量比較小，只使用20%資料進行訓練2. 模型可再優化，由於google colab的限制，我在訓練模型的時候必須分段訓練才不會RAM不足，我相信RAM足夠的情況下結果會更好。

對於資料集的改變：因為有發現在有些子類別的準確度為0，我判斷是因為資料量不足以讓模型了解字意義，因此我在Kaggle上找了一個關於國家的資料集(https://www.kaggle.com/datasets/nelgiriyewithana/countries-of-the-world-2023?resource=download) **Global Country Information Dataset 2023** 作為data加入訓練，然而得到的結果沒有變化，我認為是我提供的資料偏向國家資訊，而非國家關係，若能找到更好的資料集，我相信準確度表現會更好。