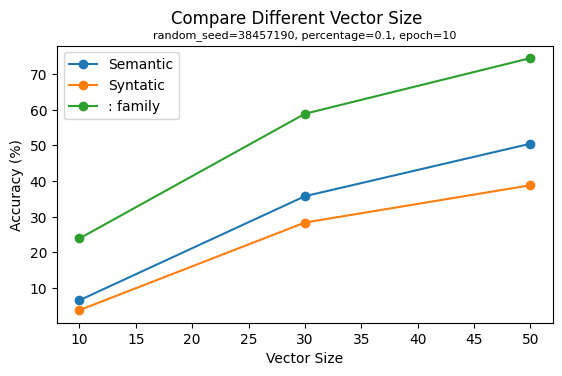
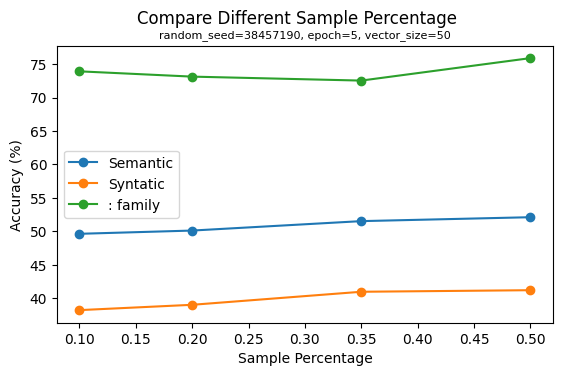
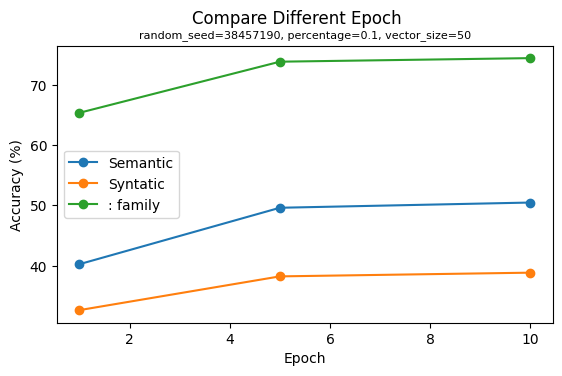
**11310CS563100 NLP Assignment 1: Word Analogy Report**

NTHU 110020007 施淙綸

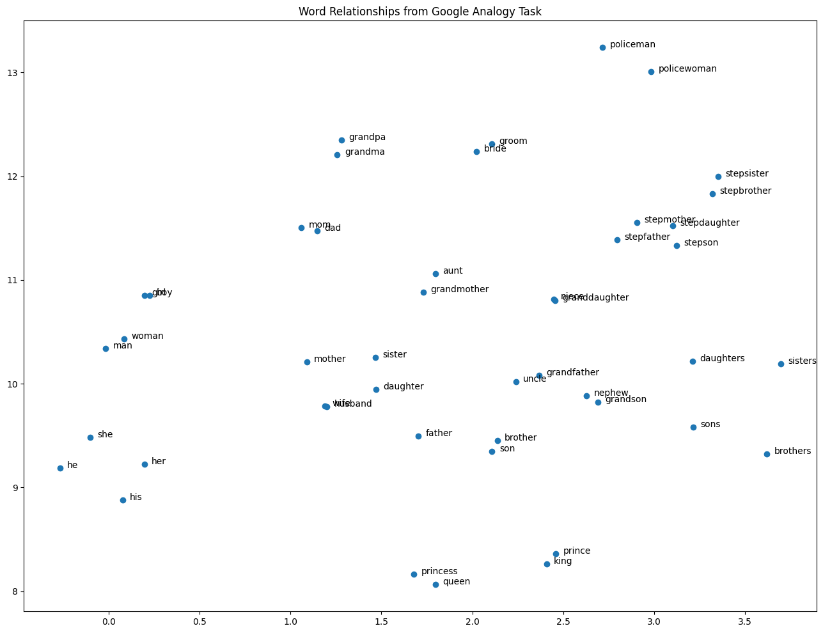
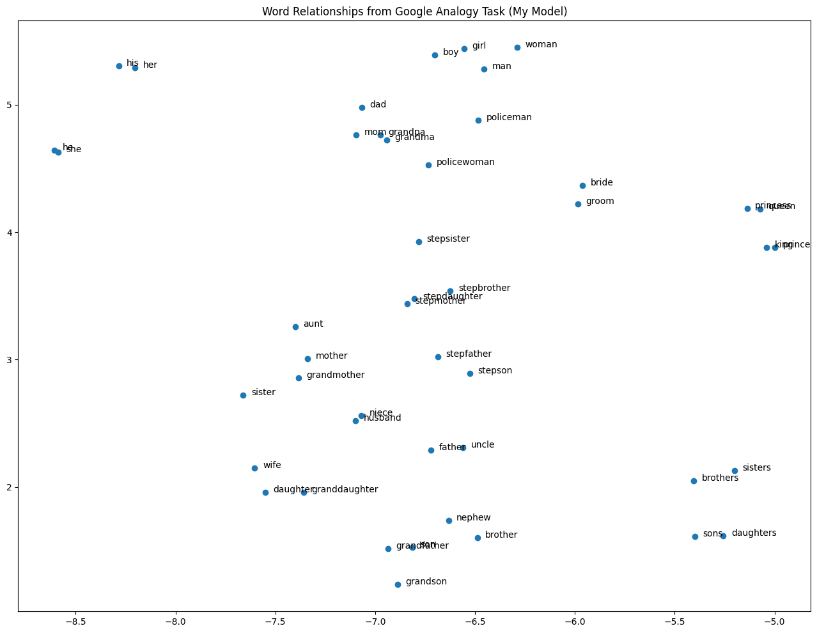
* Which embedding model do you use?
  + 我使用Word2Vec作為embedding 的模型。
* What are the pre-processing steps?
  + 在資料前處理時，助教給予的檔案已經將所有分隔字符（如逗號、句號和分號等）去除，並且分割為一列一句的形式。
  + 我在觀察檔案內容後，額外去除包含非ASCII 字符的 Word，如阿方索十三世的西班牙文原名Alfonso León Fernando María Jaime Isidro Pascual Antonio de Borbón y Habsburgo-Lorena會被預處理成Alfonso Fernando Jaime Isidro Pascual Antonio de y Habsburgo-Lorena。
  + 考慮真正的完整去除非英文單字實作上有困難，所以保留dataset中，單字字母皆為ASCII的其他語種字詞。
* What are the hyperparameter settings? What do you believe is the primary factor causing the accuracy differences for your approach? What’s the difference in word representations if you increase the amount of training data?
  + 由於時間有限，且訓練模型效能與耗時需求很高，故僅考慮：
    - RANDOM\_SEED: 隨機分割訓練集的種子碼。（Dummy variable, 討論影響模型表現沒有意義。）
    - RANDOM\_PERCENT: 隨機抽取用來訓練的句子，佔整個dataset的比例。
    - EPOCH: 模型訓練迭代數。
    - VECTOR\_SIZE: 訓練Word Vector維度大小。
  + 我總共訓練10多個模型來觀察各項變數對模型準確率的影響，數據圖表和分析說明如下：



* + 為了觀察各項參數是如何影響模型的準確率，我固定其餘三項參數，比較最後一個參數變動是如何影響模型在Semantic和Syntatic兩大分類及: family小分類中的表現。
  + 從左圖可以看到，模型在第一代和第五代間有明顯的準確率提升，但在第五代和第十代卻沒有明顯的差異，可以推測模型大約在第五代就已經收斂。訓練到模型收斂所需代數會受到抽樣比例、向量大小和抽樣均勻度影響，在左圖的測試當中第五代就收斂有可能是因為僅抽取百分之十的樣本進行訓練。在模型收斂前，越多代的訓練會帶來更佳的準確率，但仍須注意是否有過擬合的問題。
  + 從中間圖可以看到，模型在大分類中，隨著抽樣比例提高，模型準確率呈現上升趨勢。但在: family子分類中卻先下降後上升，推測是抽樣出來的訓練集不同導致的。抽樣比例越高、準確率上升是符合直覺的趨勢，若是效能允許，可以將完整訓練集投入訓練。我也有嘗試更小的抽樣比例，發現越小的抽樣比例雖然可以顯著減少模型訓練的時長，但在抽樣不一定平均的情況下，常常會有模型訓練出的word vector沒有完全包含測試題目集有的字詞，可能改進的方式是額外篩選並保留具有題目字詞的句子，針對題目提升準確率。
  + 從右圖可以看到，模型在embedded 的向量數量提升時，準確率有所上升。但在數量為30和50之間提升的幅度沒有數量為10和30來的大，推測是因為越多的向量漸漸能充分表示單字的抽象意涵，合理預期隨著越來越多向量數量，帶來的準確率提升有限，而越多的向量數量會導致訓練的時長大幅度增加，這是需要考慮的。
* What is the performance for different categories or sub-categories?
  + Hyperparameter settings of my model: RANDOM\_SEED=38457190, RANDOM\_PERCENT=0.5, EPOCH=5, VECTOR\_SIZE=50

|  |  |  |
| --- | --- | --- |
|  | glove-wiki-gigaword-100 | My Model |
| Semantic | **65.340** | **52.103** |
| : capital-common-countries | 93.874 | 71.937 |
| : capital-world | 88.948 | 67.838 |
| : currency | 14.203 | 11.085 |
| : city-in-state | 30.807 | 28.699 |
| : family | 81.621 | 75.889 |
| Syntatic | **61.255** | **41.190** |
| : gram1-adjective-to-adverb | 24.395 | 13.710 |
| : gram2-opposite | 20.074 | 12.685 |
| : gram3-comparative | 79.129 | 58.408 |
| : gram4-superlative | 54.278 | 29.234 |
| : gram5-present-participle | 69.508 | 29.830 |
| : gram6-nationality-adjective | 87.867 | 77.611 |
| : gram7-past-tense | 55.449 | 37.115 |
| : gram8-plural | 71.997 | 43.018 |
| : gram9-plural-verbs | 58.391 | 39.540 |

* + 由以上表格可知，我訓練的模型在每一個大、小分類的準確率皆低於預訓練的模型。其中，: family, : currency, : city-in-state子分類的表現和預訓練模型相近，其餘皆有較大的差距。我認為是因為隨機抽樣導致的訓練內容不平均，使得部分分類相關的字詞出現較多，訓練較為充分，反之。
  + 另外，可以觀察到兩種模型在不同分類上皆具有相似的準確率趨勢，比如在: capital-common-countries及: family兩個模型皆表現較好，: gram1-adjective-to-adverb和: gram2-opposite皆表現較差。
* What’s your discovery from your t-SNE visualization plots?
  + t-SNE在映射時好像會有隨機過程，這會導致每次視覺化後都無法得到相同的樣子，但還是可以從圖片中了解字詞的向量間的關係。



* + 從左圖左上可以觀察到在英文當中，he vs. her的關係和his vs. her是相似的，他們是平行對應的關係，從右圖左下也可以觀察到相同的關係，這是符合Word2Vec的直覺的。而右圖訓練出來的結果看起來比較均勻，四個角大約呈現平行四邊形。
  + 從兩張圖片可以發現特別有趣的事情，brother, sister, son and daughter四種關係和father的距離，複數型態的字詞都會比單數型態的字詞來的近，這也符合日常的直覺，很多個會比一個來的多，這是從Word2Vec模型相當吸引人的結果。
  + 兩張圖的boy vs. girl和men vs. women都可以發現，前者關係的距離都比後者來的近，這好像意味著隨著人類年齡增長，男性和女性詞語的距離會越來越遠，不同的特徵都會漸漸的越來越大，這可以是好玩的猜測。
  + 另外，step-稱謂系列和原生稱謂系列會各自形成一個聚落湊在一起，代表是否為「繼任」在英語的語境裡面是有影響的。
* 遭遇到的困難
  + 資料集過於龐大，內存不足：因為整個檔案大約就有20 GB，在抽樣的時候就無法用傳統方法全部讀進來直接random，目前是使用句子總數生成Boolean List作為挑選的mask，然後用迭代的方法把句子拉進來預處理再存出去。接下來，就算用小比例抽樣還是很容易無法一次就完整讀取sentences，後來翻閱Word2Vec的說明發現可以用檔案輸入，迭代的方法他已經幫忙寫好了，另一種解法是將句子包成Generator讓Word2Vec Class迭代呼叫。
  + 訓練集大小顯著的影響到不同的硬體設備的訓練時長，為了避免等待時間過久，我使用自己的電腦進行訓練，而非Google Colab，訓練環境如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | System | CPU | Python version |
| Running environment | Windows 11 Version 23H2 (KB5043145) | 11th Gen Intel® Core™ i7-11370H @ 3.30GHz | CPython 3.11.1 |

* + 前幾天Python 3.13正式釋出，根據官方的說法，兩個版本都有對效能改進，說不定把Python版本升上去可以減少一點訓練時間。