國 立 中 央 大 學

資訊工程學系軟體工程碩士班

碩士論文

利用輔助語句與 BERT 模型偵測詞彙的上下位關係

Hypernym and Hyponym Detection Based on Auxiliary Sentences and the BERT Model

研 究 生：曾莊

指導教授：陳弘軒　博士

中 華 民 國 一百 一 十 年 六 月

國立中央大學圖書館學位論文授權書

填單日期：2021/08/05　　2019.9 版

授權人姓名	曾花	學 號	108525010
系所名稱	資訊工程學系	學位類別	☑碩士　□博士
論文名稱	利用中文日文語句與BERT模型偵測詞彙的上下位關係	指導教授	陳弘軒

學位論文網路公開授權

授權本人撰寫之學位論文全文電子檔：

・在「**國立中央大學圖書館博碩士論文系統**」.

(✓)同意立即網路公開

(　)同意 於西元＿＿＿年＿＿＿月＿＿＿日網路公開

(　)不同意網路公開，原因是：＿＿＿＿＿＿＿＿＿＿

・在**國家圖書館「臺灣博碩士論文知識加值系統」**

(✓)同意立即網路公開

(　)同意 於西元＿＿＿年＿＿＿月＿＿＿日網路公開

(　)不同意網路公開，原因是：＿＿＿＿＿＿＿＿＿＿

依著作權法規定，非專屬、無償授權國立中央大學、台灣聯合大學系統與國家圖書館，不限地域、時間與次數，以文件、錄影帶、錄音帶、光碟、微縮、數位化或其他方式將上列授權標的基於非營利目的進行重製。

學位論文紙本延後公開申請 (紙本學位論文**立即公開者此欄免填**)

本人撰寫之學位論文紙本因以下原因將延後公開

・**延後原因**

(　)已申請專利並檢附證明，專利申請案號：

(　)準備以上列論文投稿期刊

(　)涉國家機密

(　)依法不得提供，請說明：＿＿＿＿＿＿

・**公開日期**：西元＿＿＿＿年＿＿＿＿月＿＿＿＿日

※繳交教務處註冊組之紙本論文(送繳國家圖書館)若不立即公開，請加填「國家圖書館學位論文延後公開申請書」

研究生簽名：曾花　　指導教授簽名：陳弘軒

*本授權書請完整填寫並親筆簽名後，裝訂於論文**封面之次頁**。

國立中央大學碩士班研究生

論文指導教授推薦書

資訊工程學系軟體工程碩士班 學系/研究所 曾莊 研究生

所提之論文 利用輔助語句與 BERT 模型偵測詞彙的

上下位關係

係由本人指導撰述，同意提付審查。

指導教授 馮弘緒 (簽章)

110 年 07 月 09 日

國立中央大學碩士班研究生
論 文 口 試 委 員 審 定 書

資訊工程學系軟體工程碩士班 學系/研究所 曾莊 研究生

所提之論文 利用輔助語句與 BERT 模型偵測詞彙的上下位關係

經由委員會審議，認定符合碩士資格標準。

學位考試委員會召集人 黃瀚萱

委　　員 馮玉珍

張嘉惠

中　華　民　國 110 年 7 月 27 日

1100719

利用輔助語句與 BERT 模型偵測詞彙的上下位關係

摘要

詞向量模型是一種利用文本的上下文關係產生詞彙相應之向量的技術。通常，我們可利用詞向量間的餘弦相似度來計算兩個詞彙間的相關程度。然而，我們卻難以利用詞向量來偵測兩個詞彙是否具備上位詞-下位詞的關係。另外，由於上下關係是一種不對稱的語義關係，即使給定一對具備上位詞-下位詞關係詞彙，我們也難以採用一般對稱的距離量度來決定何者為上位詞、何者為下位詞。

本論文提出一個基於 BERT 預訓練語言模型搭配額外建構的輔助語句來判斷一對詞彙的上下關係，任務共分兩階段。階段一：判斷詞對是否具有上下關係。若階段一的結果為真，則進入階段二：判斷何者為上位詞，何者為下位詞。經過實驗，我們發現兩種建構輔助語句的方式: BERT+Q 和 BERT+Q+PosNeg 能有效地利用詞向量判斷階段一及階段二的任務。

關鍵字: 詞向量, BERT 語言模型, 微調, 上下關係

Hypernym and Hyponym Detection Based on Auxiliary Sentences and the BERT Model

Abstract

The word embedding model is a technique that utilizes contextual words to generate a vector for each word, which is called word embedding. Usually, we can use the cosine similarity between a pair of word embeddings to calculate the relevance score between the two words. However, it is difficult to use word embeddings to detect the hypernym-hyponym relationship between two words. In addition, being an asymmetric semantic relationship, even when given a pair of vocabularies with a hypernym-hyponym relationship, it is challenging to apply general distance measures, which are often symmetric, to determine which is the hypernym and which is the hyponym.

This thesis proposes a model based on a BERT pre-trained model with auxiliary sentences to determine the hypernym-hyponym relationship of a pair of words. The entire process is consisted of two tasks. First, when given a pair of words, the model determines whether the word pair has a hypernym-hyponym relationship. Then, if the result is true, the model proceeds to the second task: distinguishing the hypernym and the hyponym. Experimental results show that two approaches to construct auxiliary sentences, BERT+Q and BERT+Q+PosNeg, can effectively accomplish both tasks.

Keywords: Word embeddings, BERT, fine tune, Hypernym

目錄

圖目錄

頁次

表目錄

頁次

一、 緒論

1.1 研究動機

在自然語言處理中，如果想要以向量表示詞語，最簡單直接的方法為 1-of-N Encoding，向量的維度為字典大小，字典中的每一個詞對應到一個維度。但這種表示法的向量維度會隨著字典大小的增加越來越大，且向量本身無法表示詞與詞之間的關係。為了解決此問題，科學家根據分佈假說 (distributional hypothesis)[1] 利用大量文本上下文資訊訓練的詞向量例如：Word2Vec[2]、GloVe[3] 被提出。

但是利用分佈假說提出的詞向量有一個主要的缺點：詞與詞之間不同的關係像是同義詞、反義詞、上下關係等等不同的關係都被混合在同一個向量空間中無法分辨。為了使上下關係得以被分辨，常見的方法有三種：對以上下文資訊訓練過後的預訓練詞向量進行微調 [4][5]、在訓練詞向量時加入額外與關係有關的資訊 [6][7][8][9]、訓練分類器時加入上下關係 [10][11]。

受到 [12][13] 的啓發，在預訓練 BERT 模型中加入輔助句子能使分類任務的目標更加精確，既不需重新訓練如 BERT 一般巨大的語言模型，且取得了不錯的效果，本論文在建構輔助的語句微調 BERT 分類模型時，加入詞向量上下關係的資訊，希望透過此方式使調整後的 BERT 分類模型具有分辨上下關係的能力。

1.2 研究目標

本論文透過建構輔助的 (auxiliary) 句子將上下關係語料庫中的單詞對樣本轉為真實的語句，使輸入 BERT 分類模型的文字貼近預訓練時使用的文本，並藉由輔助的句子使模型學習到上下關係。

1.3 研究貢獻

在 BERT 語言模型的預訓練詞向量形成的過程中，輸入的文本皆為真實存在、有文法結構的語句；而通常 BERT 為了下游分類任務而進行的微調大概流程如下：輸入一個或數個句子，最後將開頭的 token([CLS]) 進行分類，使調整後的模型具有分辨單詞對之間是否有上下關係的能力。本方法只要能取得上義詞—下義詞的語料庫，便可以透過微調 BERT 語言模型調整預訓練的 BERT 詞向量，使其得以分辨兩個單詞是否具有上下關係，以及若有上下關係何者為上義詞、何者為下義詞。

1.4 論文架構

本論文共分為五個章節, 架構如下:

第一章、說明本篇論文之研究動機、研究目標及研究貢獻。

第二章、介紹與本論文相關的研究。

第三章、說明本論文模型架構。

第四章、說明本論文實驗設計、使用資料與實驗結果探討。

第五章、本論文之結論與未來展望。

二、 相關研究

2.1 加強詞向量的同反義字詞辨別能力

2.1.1 Retrofitting

這篇論文 [4] 提出了使用同義詞語料庫調整預訓練詞向量的方法, $\hat{Q} = (\hat{q_1}, ..., \hat{q_n})$ 代表育訓練的詞向量，並以 $\hat{Q}$ 初始化一個新的向量空間 $Q = (q_1, ..., q_n)$，目標函式為最小化新的向量 Q 中同義詞對 (i, j) 之間的距離, 其詳細定義如下:

$$\psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{i,j \in E} \beta_{ij} \|q_i - q_j\|^2 \right] \tag{2.1}$$

式中 E 為同義詞語料集，α、β 為控制權重用的超參數，α 控制新向量 Q 與原向量 $\hat{Q}$ 的相近程度，β 控制新同義詞向量彼此接近的程度。

2.1.2 JointReps

這篇論文 [6] 提出了同時使用上下文資訊及知識庫 (knowledge base) 學習詞向量的方法。目標函數一一敘述如下:

$$J_C = \frac{1}{2} \sum_{i \in V} \sum_{j \in V} f(X_{ij}) (\boldsymbol{w}_i^T \widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j - log(X_{ij}))^2 \tag{2.2}$$

$$f(t) = \begin{cases} (t/t_{max})^{\alpha} & if\ t < t_{max} \\ 1 & otherwise \end{cases} \tag{2.3}$$

J_C 為 GloVe 原本的目標函數，V 為單字字典，X_{ij} 表示在目標的詞 w_i 的 contextual window 中，$\widetilde{w}_j$ 出現的次數；$\boldsymbol{w_i}, \widetilde{\boldsymbol{w_j}}$ 為詞向量，$b_i, \tilde{b}_j$ 為 bias，*f(t)* 為 weighting function。

$$J_S = \frac{1}{2}\sum_{r\in R}\sum_{i\in V}\sum_{j\in V} R_r(w_i, \widetilde{w_j})(\boldsymbol{w}_i - \widetilde{\boldsymbol{w}}_j)^2 \tag{2.4}$$

J_S 為與 knowledge base 中關係 (如同義詞、反義詞等) 有關的目標函數。R 代表在 knowledge base 中的語義關係，$R_r(w_i, w_j)$ 代表兩個字之間的語義關係在 knowledge base 中有多強。

$$J = J_C + \lambda J_S \tag{2.5}$$

將 J_C 與 J_S 相加，就是最後的目標函數。

2.2 加強詞向量的上下位字詞辨別能力

2.2.1 HyperVec

該論文 [7] 由同作者的前一篇 [14] 拉近同義詞、推遠反義詞的目標函式修改而來。同反義詞論文目標函式如下:

$$\begin{aligned} &\sum_{w\in V}\sum_{c\in V}\{(\#(w,c)log\sigma(sim(w,c)) + k\#(w)P_0(c)log\sigma(-sim(w,c))) \\ &+(\frac{1}{\#(w,u)}\sum_{u\in W(c)\cap S(w)} sim(w,u) - \frac{1}{\#(w,v)}\sum_{v\in W(c)\cap A(w)} sim(w,v))\} \end{aligned} \tag{2.6}$$

其中在2.6中，第一行為 skip gram 目標函式，w 為中心字，c 為 w 的上下文，$S(w)$ 為同義詞集合，$A(w)$ 為反義詞集合，$W(c)$ 為上下文 c 的 sliding window。本式藉由對每一個上下文 c 的 sliding window 中的同義詞拉越近、反義詞推越遠，使得與中心字 w 擁有越多相同上下文的同義詞拉近，反義詞推遠。

HyperVec 論文提出在 SGNS 的目標函數中同時加入上下關係的資訊一起訓練，其目標函數為:

$$J_{SGNS} = \sum_{w \in V_W} \sum_{c \in V_C} J_{(w,c)} \tag{2.7}$$

$$\mathbb{H}^+(w,c) = \{u \in \mathbb{W}(c) \cap \mathbb{H}(w) : cos(\vec{w},\vec{u}) - cos(\vec{u},\vec{c}) \geq \theta\} \tag{2.8}$$

$$\mathbb{H}^-(w,c) = \{v \in \mathbb{W}(c) \cap \mathbb{H}(w) : cos(\vec{w},\vec{u}) - cos(\vec{v},\vec{c}) < \theta\} \tag{2.9}$$

J_{SGNS} 與 2.6第一行一樣為 skip gram 目標函式。在式 2.8 2.9 中，$\mathbb{W}(c)$ 表示在上下文單字 c 的 sliding window 中的字，$\mathbb{H}(w)$ 表示與 w 有關的上位詞；$\mathbb{H}^+$、$\mathbb{H}^-$ 代表以 θ margin 將 $\mathbb{W}$ 分成兩組。

$$L_{(w,c)} = \frac{1}{\#(w,u)} \sum_{u \in \mathbb{H}^+(w,c)} \partial(\vec{w},\vec{u}) \tag{2.10}$$

$$L_{(v,w,c)} = \sum_{v \in \mathbb{H}^-(w,c)} \partial(\vec{v},\vec{w}) \tag{2.11}$$

式 2.10、2.11應為2.6第二行同義詞、反義詞目標函式修改而來；其中 $\partial(\vec{x},\vec{y})$ 為 cosine similarity 對 x 的微分。式 2.10 目標為利用上下文單字 c 將共同出現在 sliding window 中的上下關係詞對 (w, u) 拉近。式 2.11

則是相反，如果上下文單字 c 離上位詞較近，則把上位詞往下位詞拉近。

$$J_{(w,v,c)} = J_{(w,c)} + L_{(w,c)} + L_{(v,w,c)} \tag{2.12}$$

$$J = \sum_{w \in V_W} \sum_{c \in V_C} J_{(w,v,c)} \tag{2.13}$$

最後目標函式即為式2.7 2.10 2.11 三式相加。

2.2.2 Poincaré

該論文 [8] 用 Poincaré ball 學習雙曲空間中的具有上下關係詞語的詞向量，使其學習到的詞向量同時具有上下階層的關係和詞向量間的相似度。此論文僅僅使用了有上下關係的語料庫，並沒有使用其他文本來訓練詞向量上下文之間的關係，其目標函式如下:

$$\mathcal{L}(\Theta) = \sum_{(u,v) \in D} log(\frac{e^{-d(\boldsymbol{u},\boldsymbol{v})}}{\sum_{v' \in N(u)} e^{-d(\boldsymbol{u},\boldsymbol{v'})}}) \tag{2.14}$$

$$d(\boldsymbol{u},\boldsymbol{v}) = arcosh\left(1 + 2\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{(1-\|\boldsymbol{u}\|^2)(1-\|\boldsymbol{v}\|^2)}\right) \tag{2.15}$$

$D = (u, v)$ 為具有上下關係的詞對，$N(u) = \{v' \mid (u, v') \notin D\} \cup \{v\}$ 為與 u 沒有上下關係的詞。$d(\boldsymbol{u},\boldsymbol{v})$ 為詞對 (u, v) 在雙曲空間中的距離，$\|\cdot\|$ 為 Euclidean norm。

2.2.3 LEAR

這篇論文 [5] 提出一個 post-training 的方法，藉由一些上下文以外的上下關係資訊對以上下文預訓練的詞向量進行調整，使調整後的詞向量具有判斷上下關係的能力。而此篇論文藉由調整各個詞向量之間的

cosine similarity 和具有上下關係的詞向量彼此的 norm，使得有上下關係的詞彼此較近，且上位詞的長度較下位詞長。詳細目標函數共有四項，Attract 項為拉近同義詞，Repel 項為推遠反義詞，Lexical Entailment 為調整上下關係詞向量的長度。各項詳細定義如下:

$$\begin{aligned} Att(\beta_A, T_A) = \sum_{n=1}^{k_1} \big[& \tau\left(\delta_{att} + cos(\boldsymbol{x_l^i}, \boldsymbol{t_l^i}) - cos(\boldsymbol{x_l^i}, \boldsymbol{x_r^i})\right) + \\ & \tau\left(\delta_{att} + cos(\boldsymbol{x_r^i}, \boldsymbol{t_r^i}) - cos(\boldsymbol{x_l^i}, \boldsymbol{x_r^i})\right) \big] \end{aligned} \tag{2.16}$$

式2.16中 β_{Att}、T_A 分別為同義詞的一個 batch 及每個字的 psuedo negative set，$(\boldsymbol{x_l^i}, \boldsymbol{x_l^i})$ 為同義詞樣本。$\boldsymbol{t_l^i}$ 為 $\boldsymbol{x_l^i}$ 的 pseudo negative，指的是在一個 batch 中離 $\boldsymbol{t_l^i}$ cosine similarity 最小的字。δ_{att} 是 margin，$\tau(x) = max(0, x)$ 是 hinge loss，cos 是 cosine similarity。

$$\begin{aligned} Rep(\beta_R, T_R) = \sum_{n=1}^{k_2} \big[& \tau\left(\delta_{rep} + cos(\boldsymbol{x_l^i}, \boldsymbol{x_r^i}) - cos(\boldsymbol{x_l^i}, \boldsymbol{t_r^i})\right) + \\ & \tau\left(\delta_{rep} + cos(\boldsymbol{x_l^i}, \boldsymbol{x_r^i}) - cos(\boldsymbol{x_r^i}, \boldsymbol{t_r^i})\right) \big] \end{aligned} \tag{2.17}$$

式2.17中 β_{Att} 為反義詞的一個 batch，$(\boldsymbol{x_l^i}, \boldsymbol{x_l^i})$ 為反義詞樣本，其餘符號與式2.16 相同。

$$Reg(\beta_A, \beta_R) = \sum_{\boldsymbol{x_i} \in V(\beta_A \cup \beta_R)} \lambda_{reg} \|\widehat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x_i}\|_2 \tag{2.18}$$

式2.18的 $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}$ 代表預訓練的詞向量，$\boldsymbol{x_i}$ 表示調整過後的詞向量，λ_{Reg} 為常數。此項用來限制調整後的詞向量不合預訓練的詞向量差太多。

$$LE_j(B_L) = \sum_{i=1}^{k_3} D_j(\boldsymbol{x_i}, \boldsymbol{y_i}), where$$
$$\begin{cases} D_1 = |x| - |y| \\ D_2 = \frac{|x|-|y|}{|x|+|y|} \\ D_3 = \frac{|x|-|y|}{max(|x|,|y|)} \end{cases} \tag{2.19}$$

式2.19中 $(\boldsymbol{x_i}, \boldsymbol{y_i})$ 為上下關係的正樣本，$\boldsymbol{x_i}$ 為下位詞，$\boldsymbol{y_i}$ 為上位詞，藉此調整詞向量的長度，使得上位詞的 norm 大於下位詞。

總目標函式為 4 項相加：$Att(\beta_A, T_A) + Rep(\beta_R, T_R) + Reg(\beta_A, \beta_R) + LE_j(B_L)$。

2.2.4 HWE

這篇論文 [9] 提出若是有由下至上的一條上下關係的路徑 P(bird) = (bird → vertebrate → chordate → organism → animal)，就可以使用 GloVe[3] 的目標函數J_C 加上和上下關係路徑有關的目標函數 J_T，使該路徑的葉節點 (leaf node) 詞向量接近該路徑的其餘節點。例如，使 P(bird) 中 bird 的詞向量接近其餘 P(bird) 裡的詞向量。其目標函數入下:

$$J = J_T + J_C \tag{2.20}$$

$$J_T = \frac{1}{2} \sum_{i \in V} \left\| \boldsymbol{w_i} - \sum_{j \in P(w_i)} \lambda(\widetilde{w_j}) \widetilde{\boldsymbol{w_j}} \right\|_2^2, where$$
$$\lambda(\widetilde{w_j}) = exp(L_{w_i} - D_{\widetilde{w_j}}) \tag{2.21}$$

V 為單字字典；λ 為在路徑中上位詞的權重，L_{w_i} 代表整個 P(w_i) 路徑長，$D_{\widetilde{w_j}}$ 代表從 w_i 到其上位詞 $\widetilde{w_j}$ 路徑長。

$$J_C = \frac{1}{2}\sum_{i\in V}\sum_{j\in P(w_i)} f(X_{ij})\left(\boldsymbol{w_i^T}\widetilde{\boldsymbol{w_j}} + b_i + b_j - log(X_{ij})\right)^2, where$$
$$f(t) = \begin{cases} (t/t_{max})^\alpha & if\ t < t_{max} \\ 1 & otherwise \end{cases} \tag{2.22}$$

X 為 co-occurrence matrix，其 context window 大小為 10 ，X_{ij} 代表 w_i 與其上位詞 $\widetilde{w_j}$ 的共同出現次數，$f(t)$ 為權重函數，b_i 和 b_j 分別為 w_i 和 $\tilde{w}_j$ 的 bias，$\alpha = \frac{3}{4}$，$t_{max} = 100$。

2.2.5 Roller and Erk

該論文 [15] 提出像 Principal Component Analysis(PCA) 一樣的方法，對一組樣本 (H, w)(H代表上位詞，w代表單字) 串接在一起的向量 $< H, w >$，每一次都找到一個 "$Hypernym - feature\ detector(H - feature\ detector)$" $\hat{p}$ (例如：“such as”，“include”)，將每一個 $< H, w >$ 做以下處理：$H_{i+1} = \frac{H_i - proj_{\hat{p}}(H_i)}{\|H_i - proj_{\hat{p}}(H_i)\|}$ 、$w_{i+1} = \frac{w_i - proj_{\hat{p}}(w_i)}{\|w_i - proj_{\hat{p}}(w_i)\|}$，如同 PCA 一般，去除前一個 Hypernym-feature detector 的元素。最後用下式：

$$F_i(< H_i, w_i >, \hat{p}_i) =< H_i^T w_i, H_i^T \hat{p}_i, w_i^T \hat{p}_i, (H_i^T - w_i)^T \hat{p}_i > \tag{2.23}$$

將 $F_1 ... F_n$ 組成長度為 4n 的特徵向量丟入分類器，訓練得出最後的分類結果。

2.2.6 Shwartz

這篇論文 [10] 提出同時使用上下關係的路徑以及詞向量兩部分來訓練分辨上下關係的分類器。

路徑的部分：一句有上下關係的句子 “parrot is a bird” 會首先被轉換成 edge 的向量 $\vec{v_e} = [\vec{v_l}, \vec{v_{pos}}, \vec{v_{dep}}, \vec{v_{dir}}]$ 的形式，X/NOUN/nsubj/<, be/VERB/ROOT/-, Y/NOUN/attr/>。而一條 path 的向量為將與 path 有關的 edge 丟入 LSTM 模型後的輸出向量 $\vec{o_p}$。一個詞對 (x, y) 可能會有很多路徑，因此真正的路徑向量為所有與 (x, y) 有關的路徑向量平均：

$$\vec{v}_{paths(x,y)} = \frac{\sum_{p \in paths(x,y)} f_{p,(x,y)} \cdot \vec{o_p}}{\sum_{p \in paths(x,y)} f_{p,(x,y)}} \tag{2.24}$$

$f_{p,(x,y)}$ 代表路徑 p 在所有路徑中出現的頻率。

詞向量部分：將 (x, y) 詞對的詞向量 $(v_{\vec{w_x}}, v_{\vec{w_y}})$ 與 $\vec{v}_{paths(x,y)}$ 連接成為 $v_{xy} = [v_{\vec{w_x}}, \vec{v}_{paths(x,y)}, v_{\vec{w_y}}]$ 丟入分類器進行分類，如此一來 (x, y) 詞對就會擁有與其路徑有關、也與詞向量有關的向量：v_{xy}。

另外，本篇論文提出的資料集詳細介紹在 section 4.3.1。

2.2.7 BiRRE

這篇論文 [11] 提出不直接使用預訓練詞向量來判斷是否具上下關係的方法，而是將詞向量正交投影至新的向量空間，並在投影時加入額外的上下關係資訊，認為如此可以避免 “lexical memorization” 的問題。

所謂 “ lexical memorization” 指的是模型所學到的並非兩個詞之間的關係，而是學到某詞語是否為「典型」的上義詞，例如在訓練資料集中出現了 (dog, animal)、(cat, animal)、(cow, animal) 皆被標示為正樣本，模型可能學到了 animal 是一個「典型」的上義詞，進而將任何 (x, animal) 的樣本視為有上下關係。

該模型共分成兩個步驟訓練。第一個步驟為投影預訓練的詞向量至新的向量空間，並投影時加入上下關係資訊，共分為上位詞轉成下位詞、下位詞轉成上位詞兩部分：上位詞轉成下位詞部分限制新上位詞向量與原本的下位詞近；下位詞轉成上位詞部分限制新下位詞向量與原本的上位詞近。第二步驟為固定第一步驟的模型，並用第一步驟的轉換過後的上位詞與其下位詞原本向量的差，和另一部分轉換過後的下位詞與其上位詞原本向量的差串聯在一起，並以此進行是否有上下關係的分類。

2.3 Language Model(e.g., BERT) 加上輔助句子的研究

2.3.1 使用輔助句子幫助「面相情感分析」任務

在情感分析任務中，有一種任務是 “targeted aspect-based sentiment analysis (TABSA)”，首先要判斷出目標屬於哪一個方面的評論 (例如：餐廳的地點、價錢或是安全性)，接著判斷評論為正面、中立或負面。若是以 BERT 模型將輸入的語句直接分為 (方面類別數量 * 情緒類別數量) 類，效果並不好，因此 [12] 提出使用建構輔助的句子，使分類更明確簡單。假設有一個我們想知道的 target-aspect pair: (LOCATION1, safety)，建構的輔助語句如下：

表 2.1

Methods	Auxiliary Sentence
QA-M	what do you think of the safety of location - 1 ?
NLI-M	location - 1 - safety
QA-B	the polarity of the aspect safety of location - 1 is positive the polarity of the aspect safety of location - 1 is negative
NIL-B	“location - 1 - safety - positive location - 1 - safety - negative

將輸入要分類的樣本加上以上四種輔助句子，對 BERT 預訓練的模型進行分類任務的微調。

表 2.2

Methods	Output	Auxiliary Sentence
QA-M	**sentiment polarity**	Question without sentiment polarity
NLI-M	**sentiment polarity**	Pseudo without sentiment polarity
QA-B	yes, no	Question with sentiment polarity
NIL-B	yes, no	Pseudo with sentiment polarity

輔助語句的分類如表2.2，分類的類別由原本的 (方面類別數量 * 情緒類別數量) 變成 (yes, no) 兩類，或者情緒種類數量。

2.3.2 使用輔助句子幫助「文字分類」任務

這篇論文 [13] 提出了 BERT for text classification(BERT4TC)，嘗試想要藉由府建構輔助句子將 BERT 模型的分類問題轉換成 sentence pair classification。輸入範例如表2.3。

表 2.3

Model	Input Sequence	Label
BERT4TC-S	[CLS] I like this film. [SEP]	negative, **positive**
BERT4TC-AQ	[CLS] I like this film. [SEP] What is the result? [SEP]	negative, **positive**
BERT4TC-AA	[CLS] I like this film. [SEP] positive [SEP]	{0, **1**}
	[CLS] I like this film. [SEP] negative [SEP]	{**0**, 1}
BERT4TC-AWA	[CLS] I like this film. [SEP] The result is positive [SEP]	{0, **1**}
	[CLS] I like this film. [SEP] The result is negative [SEP]	{**0**, 1}

表2.3的 BERT4TC-S 是原本輸入的格式，其他三種是加上了不同輔助語句的樣子。可以發現，加上輔助語句後，要分類的問題更加精確，同時訓練模型的資料量也變大了。

另外，這篇論文也提到，BERT 預訓練模型的語料庫缺乏 domain knowledge，如果把與該分類任務的領域相關知識加入以訓練模型，可以地到更好的分類結果。

最後，我們整理了相關研究的特點，以及與本論文的差異，結果如表2.4。其中包含的特點有：是否使用上下關係訓練詞向量、是否屬於在訓練後加入知識 (post-training) 的方法、是否提供訓練好的詞向量、是否提供程式碼。如果有供訓練好的詞向量，則在本論文實驗中直接使用該詞向量，不重新訓練。

表 2.4: 相關研究性質

Model	使用上下關係訓練詞向量	Post-training	Pre-trained model	Code
CBOW[16](SVM)				√
SGNS[2](SVM)				√
GloVe[3](SVM)				√
Retrofit[4](CBOW)		√		√
Retrofit[4](SGNS)		√		√
JointReps[6][17](SVM)		√		
Roller and Erk[15]	√			
Schwatz[10]	√	√		√
Glavas and Ponzetto[18]	√	√		√
HyperVec[7](SVM)	√	√	√	√
Poincaré[8](SVM)	√			√
LEAR[5](SVM)	√	√	√	√
HWE[9]	√	√		
BiRRE[11]	√	√		
Our Work	√	√		√

三、 模型及方法

3.1 模型架構

本論文以 WordNet[19] 的動詞、名詞上下關係作為語料庫，用以微調 BERT 語言模型 [20]，模型概觀如圖3.1所示。將語料庫中的文字格式整理為 $[CLS]$ w_1 w_2 w_3 w_4 $[SEP]$ (w_i 為經過 BERT tokenizer 處理後的 token，其作用將在 section 3.2 中詳細說明) 後，輸入 BERT 語言模型 [20] 進行微調。Task1 任務為分辨是否具有上下關係；Task2 任務以已有上下關係為前提，判斷何者為上位詞，何者為下位詞。而將 Task1 及 Task2 結果結合，可以得到最後的結果。

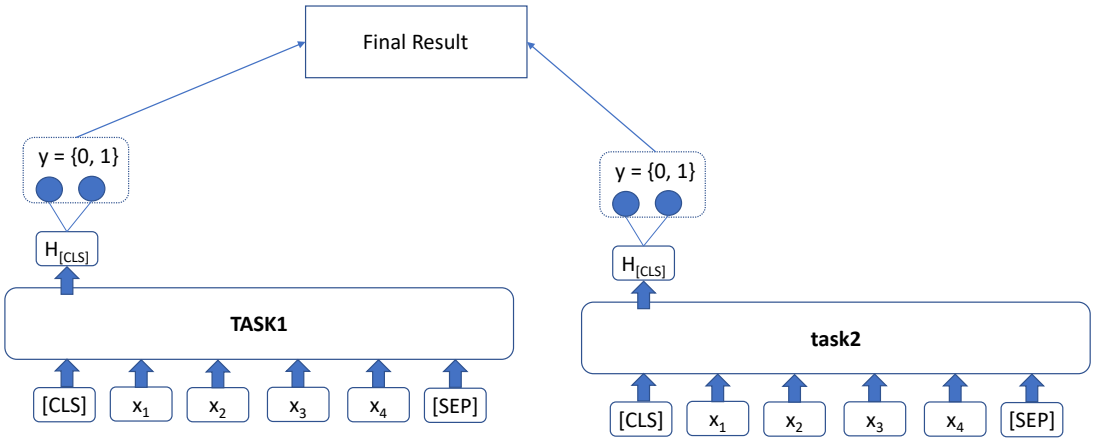

圖 3.1: 模型概觀

3.2 Task1 模型

Task1 模型的目標為當輸入樣本 (A, B) 經過 BERT tokenizer 處理，變成 $[CLS]$ w_1 w_2 w_3 w_4 $[SEP]$ 後，分辨樣本是否具有上下關係。BERT tokenizer 是一個將純文字轉換為分詞的標記解析器。作法為若是一個英文單字不存在單字字典中，則需要將其拆分成為單字字典中存在的字。例如 “hierarchically” 會被分解為 hierarchical ##ly，## 代表接續前一個字。[CLS] 為 classifier token，是輸入序列的第一個符號，當進行序列分類 (sequence classification) 時丟入分類器進行分類。[SEP] 為 separator token，當一個序列有一個以上的句子時，用 [SEP] 分句。

輸入分為兩類：正樣本及負樣本。正樣本為自 WordNet[19] 中蒐集、有上下關係的 word pairs，並有 50% 的機會 word pairs 的字詞會前後對調，如一對 word pair(bundle, object) 有 50% 的機會對調為 (object, bundle)；負樣本為從語料庫中隨機挑選的 word pairs，負樣本皆不在原本正樣本中。

將正負樣本輸入 BERT 語言模型 [20] 的格式分為三種：BERT、BERT+Q、BERT+Q+PosNeg，詳細輸入範例於表3.1。

BERT 模型為基線 (baseline)，做為 BERT+Q 模型及 BERT+Q+PosNeg 模型的比較基準。若樣本為 (A, B)，A 為第一個詞，B 為第二個詞，A 和 B 可能包含不只一個單字；輸入的字句為 *[CLS] A B [SEP]*。亦取 BERT 語言模型輸出的第一個 token(CLS token) 丟入線性分類器分為兩類，$y = 0$ 代表 AB 兩個字無上下關係，$y = 1$ 則代表有上下關係。

BERT+Q 模型就像在詢問 BERT 語言模型 [20] 一般，輸入字句為 *[CLS] A and B are hierarchically related [SEP]*，並取 BERT 語言模型輸出的第一個 token([CLS] token) 丟入線性分類器分為兩類，$y = 0$ 代表無上下關係，$y = 1$ 則代表有上下關係。

BERT+Q+PosNeg 模型則是在模型 BERT+Q 的句子之後加上第二

句: positive 或 negative, 輸入字句為: *[CLS] A and B are hierarchically related [SEP] positive [SEP]*、*[CLS] A and B are hierarchically related [SEP] negative [SEP]* 兩句，同樣取 BERT 語言模型輸出的第一個 token(CLS token) 丟入線性分類器分為兩類，$y = 0$ 代表 positive 或 negative 並不是正確的標籤，$y = 1$ 則代表是正確的標籤。

表 3.1: Task1 training input examples

Model	Examples	y
BERT	[CLS] fox animal[SEP]	{0, **1**}
	[CLS] hat bird [SEP]	{**0**, 1}
BERT + Q	[CLS]fox and animal are hierarchically related[SEP]	{0, **1**}
	[CLS]hat and bird are hierarchically related[SEP]	{**0**, 1}
BERT + Q + PosNeg	[CLS] fox and animal are hierarchically related [SEP] positive [SEP]	{0, **1**}
	[CLS] fox and animal are hierarchically related [SEP] negative [SEP]	{**0**, 1}
	[CLS] hat and bird are hierarchically related [SEP] positive [SEP]	{**0**, 1}
	[CLS] hat and bird are hierarchically related [SEP] negative [SEP]	{0, **1**}

當一個詞對輸入 BERT+Q+PosNeg 後，會產生兩句輔助句子，詳細說明於表 3.2。

表 3.2: BERT+Q+PosNeg 預測結果方式說明：當一筆資料輸入至 BERT+Q+PosNeg 後，會產生兩個句子並分別預測結果。最後預測結果決定的方式為：取兩句中 $y = 1$(問句 +positive 為正確、問句 +negative 為正確) 的機率來比較，機率較大高的當成預測結果。如果 positive 的句子機率較大，則判定該詞對有上下關係，反之則沒有上下關係。

examples	y	fibnal predict
[CLS] A and B are hierarchically related [SEP]positive [SEP]	{0, 1}	取句子 1 與句子 2 中
[CLS] A and B are hierarchically related [SEP]negative [SEP]	{0, 1}	y = 1 機率高者為結果

3.3 Task2 模型

Task2 模型的目標為當輸入樣本 (A, B) 經過 BERT tokenizer 處理，變成 $[CLS]$ w_1 w_2 w_3 w_4 $[SEP]$ 後，分辨樣本的上下關係為正向或是反向。Task2 輸入分為兩類：正向 word pair 及反向 word pair，兩類皆由上下關係語料庫產生。正向 word pairs 為下位詞接著上位詞，如 (gin, alcohol)，而反向 word pairs 則為相反 (alcohol, gin)。將正向、反向 word pairs 輸入 BERT 語言模型格式同樣分為三種：BERT、BERT+Q、BERT+Q+PosNeg。

BERT 模型為基線 (baseline),做為 BERT+Q 模型及 BERT+Q+PosNeg 模型的比較基準。輸入字句為 *[CLS] A B [SEP]*，$y = 0$ 代表 word pair 為反向，$y = 1$ 代表 word pair 為正向。

BERT+Q 模型輸入字句為 *[CLS] A is a type of B [SEP]*，參考 HyperLex [21] 測試資料集問句 *"To what degree X is a type of Y?"*，同樣取模型輸出的第一個 token([CLS] token) 丟入線性分類器中分為兩類，$y = 0$ 為反向 word pairs，$y = 1$ 為正向 word pairs。

BERT+Q+PosNeg 模型輸入字句為模型 Q 的句子之後加上第二句：positive 或是 negative ，如同 Task1 BERT+Q+PosNeg 模型一般，取兩句中為正向 word pair 且第二句為 positive 以及為反向 word pair 且第二句為 negative 者標為 $y = 1$，其餘兩句 $y = 0$。

表 3.3: Task2 training input examples

Model	Examples	y
BERT	[CLS] fox animal[SEP]	{0, **1**}
	[CLS] animal fox[SEP]	{**0**, 1}
BERT + Q	[CLS] fox is a type of animal[SEP]	{0, **1**}
	[CLS] animal is a type of fox[SEP]	{**0**, 1}
BERT + Q + PosNeg	[CLS] fox is a type of animal [SEP] positive [SEP]	{0, **1**}
	[CLS] fox is a type of animal [SEP] negative [SEP]	{**0**, 1}
	[CLS] animal is a type of fox [SEP] positive [SEP]	{**0**, 1}
	[CLS] animal is a type of fox [SEP] negative [SEP]	{0, **1**}

當一個詞對輸入 BERT+Q+PosNeg 後，會產生兩句輔助句子，詳細說明於表 3.4

表 3.4: BERT+Q+PosNeg 預測結果方式說明：當一筆資料輸入至 BERT+Q+PosNeg 後，會產生兩個句子並分別預測結果。最後預測結果決定的方式為：取兩句中 $y = 1$(問句 +positive 為正確、問句 +negative 為正確) 的機率來比較，機率較大高的當成預測結果。如果 positive 的句子機率較大，則判定該詞對為正向，反之則為反向。

examples	y	fibnal predict
[CLS] A is a kind of B [SEP]positive [SEP]	{0, 1}	取句子 1 與句子 2 中
[CLS] A is a kind of B [SEP]negative [SEP]	{0, 1}	y = 1 機率高者為結果

由表3.1、表3.3可以得知，BERT+Q+PosNeg 模型輸入的資料筆數為 BERT 與 BERT+Q 的兩倍，因此在微調模型時將 BERT 及 BERT+Q 模型的資料重複一次，以達到相同訓練資料量。

3.4 損失函數

Task1、Task2 皆為二分類問題，Loss function 皆為 cross entropy:

$$L = \frac{1}{n}\sum_{i}^{n} -[y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i})] \tag{3.1}$$

$\hat{y_i}$ 為預測的分類結果，y_i 為實際分類，n 為一個訓練資料筆數。

四、 實驗結果

4.1 實驗參數細節

本論文使用 BERT 語言模型 [20] 之英文 uncased pre-trained model，輸入為 BERT 語言模型的預訓練詞向量，維度為 768，task1 共訓練 19 epochs，task2 訓練 7 epochs；學習率 (learning rate) 為 $1 * 10^{-5}$，使用 adam optimizer，batch size 設為 64，超參數經 valiadtion set 決定。

使用 scikit-learn[22] 實作的支持向量機 (SVM) 二元分類器，使用 RBF kernel $\gamma = 0.03125$ 和 cost parameter C = 8.0，SVM 分類器為各個方法被比較時使用。本論文三種模型各重複實驗 3 次，並附上平均及標準差。自 WordNet[19] 蒐集有上下關係的訓練資料正樣本約 70 萬筆，並已移除與測試資料集重複的樣本。

實驗程式公開於 (URL:)。

4.2 訓練用資料集

4.2.1 訓練用資料集 Shartz 介紹

此資料集來自 [10]，訓練資料來源有 WordNet[19]、DBPedia[23]、Wikidata[24] 及 Yago[25]。上下關係詞對標為正樣本，其他語義關係詞對標為負樣本，上下關係與其他語義關係的比例為 1:4。資料分割方式有 random split 與 lexical split 兩種，詳細資料如表 4.1。random split 為隨

機分割，而 lexical split 特別將訓練資料集與測試資料集的單字加以分隔，以避免 “lexical memorization”。所謂 “ lexical memorization” 指的是模型所學到的並非兩個詞之間的關係，而是學到某詞語是否為「典型」的上義詞，例如在訓練資料集中出現了 (dog, animal)、(cat, animal)、(cow, animal) 皆被標示為正樣本，模型可能學到了 animal 是一個「典型」的上義詞，進而將任何 (x, animal) 的樣本視為有上下關係 [10]。

表 4.1: Shwartz 訓練資料集上下關係詞對數量

分割方式	**train**	**validation**	**all**
random split	49,475	3,534	53,009
lexical split	20,335	1,350	21,685

所有在 Shwartz random、Shwartz lexical 評估資料集上測試的模型都使用此訓練資料集訓練。

4.2.2 自 WordNet 蒐集的上下關係資料集

這個資料集從 WordNet 上蒐集有上下關係的正樣本共 710,041 筆，除去實驗一 Shwartz random、Shwartz lexical 兩個實驗，本論文其餘模型都使用此資料集訓練。本論文 task1 負樣本從正樣本中隨機抽樣，task2 負樣本為反轉的正樣本，兩個 task 負樣本數量皆與正樣本相同。

4.2.3 用於 SVM 訓練資料集介紹

此資料集自 WordNet 蒐集的 70 萬筆資料中隨機抽 25000 筆正樣本、隨機生成 25000 筆負樣本，共 50000 筆訓練資料，使用在實驗一的四個測試資料集 (Kotlerman、BLESS、Baroni、Levy) 的 SVM 模型訓練上。

4.3 實驗一：task1 模型評量

4.3.1 實驗一評估用資料集 Shwartz、Kotlerman、BLESS、Baroni、Levy 介紹

本實驗共使用 5 個測試資料集, 分別為 Shwartz[10] 、Kotlerman[26]、BLESS[27]、Baroni[28]、Levy[29]。測試資料集的正樣本數量及比例於表4.2。資料格式皆為 $(Word1, Word2, True\ or\ False)$，代表此 word pair $(Word1, Word2)$ 是否具有上下關係。Shwartz 資料集與其他四個資料集的差別在於：Shwartz 資料集有 random split 和 lexical split 兩種分割方式；其餘四個資料集皆為 random split。若訓練資料集中與測試資料集重複之樣本，皆已經從訓練資料集中移除。

表 4.2: 實驗一測試資料集正樣本比例

dataset	Kotlerman	BLESS	Baroni	Levy	Shwartz lex	Shwartz rnd
# of data	2940	14547	2770	12602	6610	17670
positive %	29.93	9.19	50	7.49	19.98	19.87

本實驗使用 Shwartz[10] 之資料集檢視 task1 model 的有效性，及是否有 “lexical memorization”，該資料集之樣本自 WordNet、DBPedia 及 YAGO 蒐集。

Shwartz 測試資料集分為 random split 及 lexical split 兩種分割方式。random split 為隨機分割，而 lexical split 特別將訓練資料集與測試資料集的單字加以分隔，以避免 “lexical memorization”。我們希望避免 “lexical memorization” 的情況發生，也就是希望在這個資料集兩種分割方式的 F1 score 差越少越好，這樣代表模型越沒有”lexical memorization ”。

Kotlerman[26] 資料集為基於 [30] 提出的資料集，在一個詞對中 $(Word1, Word2)$ 中 $Word1$ 為概念較小的詞 (下位詞)，$Word2$ 為概念較

大的詞 (上位詞)。(下位詞，上位詞) 為正樣本，其餘關係，包含反向上下關係 (上位詞，下位詞) 標為負樣本。

本實驗的 BLESS 資料集為 [31] 提出，以 Bless[27] 資料集為基礎蒐集而成的子集合。BLESS 資料集包括 200 不同的英語名詞概念 (做為 Word2)，並與數個相關的 Word1 組成不同關係的 word pairs，資料集共有 14547 筆。關係共有五種：co-hyponym (coordinate)：Word1, Word2 擁有相同的上位詞，例如 (alligator, coord, lizard)；HYPER: Word2 為 Word1 的上位詞，例如 (alligator, hyper, animal) ；MERO: Word2 為 Word1 為 Word2 的一部分，例如 (alligator, mero, mouth); RANDOM-N: Word1 為隨機的名詞，與 Word2 無關，例如 (alligator, random-n, message)。

Baroni 資料集正負樣本各半，負樣本由反轉正樣本及隨機配對 Word1, Word2 產生。正樣本由 WordNet[19] 中產生 (extract)，並由 [28] 作者親自驗證，並移除具有大量下位詞的抽象概念 (如 entity, object) 而成。

Levy 資料集以人工標註的 entailment graph 為基礎構成，此資料集為醫學領域相關用字，且對上下關係認定較為寬鬆。(Hyperlex 論文 [21] 說明)

4.3.2 Kotlerman、BLESS、Baroni、Levy、Shwartz 資料集實驗結果

為了調整 F1 score 判斷正負樣本閾值，在模型預測前隨機從測試資料集抽 2% 資料，用以尋找最好的 F1 score 閾值，並將其自測試資料集中移除。ROC 曲線下面積的樣本排序：BERT 模型及 BERT+Q 模型使用「模型判斷為正樣本」的機率遞減排序，BERT+Q+PosNeg 模型使用輔助句子接上 positive，且被分類為 1 的機率遞減排序，使三個模型將其認為有上下關係的樣本排序靠前，並計算三個模型的 ROC 曲線下面積。

表4.3 中除了本論文以外的結果，為 [9] 把其他方法 (包括 CBOW、

SGNS、GloVe、Retrofit、JointRep、HyperVec、Poincaré、LEAR 、HWE) 預訓練的詞向量當作 SVM 分類器輸入，分為有上下關係、沒有上下關係兩類，並使用獨立的 validation set 調整分類器參數後的結果; 其中 CBOW、SGNS、GloVe 為沒有加入上下關係資訊訓練的詞向量, Retrofit、JointRep 為以同義詞調整後的詞向量。

結果於表4.3，Shwartz 在五個資料集上的表現無論是 F1 score 還是 ROC 曲線下面積都滿好的，而本論文三種模型的 F1 score 表現並不理想，和其他以 SVM 調整後的詞向量差不多。而 ROC 曲線下面積除了 Kotlerman 資料集外，表現皆與 Shwartz 差不多好。若是以本論文的 BERT、BERT+Q、BERT+Q+PosNeg 三種模型相比, BERT+Q+PosNeg 模型表現較好。

另一個值得注意的地方是：SGNS 詞向量在五個資料集的表現都十分不錯，而 SGNS 在訓練詞向量時並沒有上下關係的資訊。

表 4.3: task1 random split evaluation result

分類	Model	Kotlerman		BLESS		Baroni		Levy		Shwartz rnd 測試資料集	
		F1	**AUC**	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**
訓練詞向量過程未使用上下關係	CBOW(SVM)[16]	0.461	0.501	0.172	0.512	0.669	0.505	0.139	0.498	0.172	0.547
	SGNS(SVM)[2]	0.486	0.594	0.472	0.862	**0.845**	0.836	0.204	0.642	0.903	0.932
	GloVe(SVM)[3]	0.494	0.618	0.620	0.858	0.830	0.829	0.311	0.675	0.887	0.909
	Retrofit[4](CBOW)	0.458	0.591	0.553	0.862	0.702	0.708	0.271	0.628	0.868	0.905
	Retrofit[4](SGNS)	0.454	0.582	0.493	0.832	0.730	0.727	0.271	0.649	0.865	0.904
	JointReps*[6][17]	0.563	0.564	**0.907**	**0.896**	0.697	0.701	0.677	0.646	-	-
訓練詞向量過程使用上下關係	Shwartz[10]	**0.656**	**0.835**	0.640	0.875	0.656	0.835	**0.688**	**0.772**	**0.961**	**0.989**
	Glavas and Ponzetto[18]	0.465	0.607	0.490	0.873	0.746	0.765	0.273	0.682	0.877	0.968
	HyperVec[7]	0.425	0.578	0.479	0.844	0.823	0.818	0.267	0.642	0.831	0.858
	Poincaré[8]	0.199	0.514	0.246	0.590	0.207	0.540	0.096	0.513	0.666	0.753
	LEAR[5]	0.491	0.632	0.677	0.893	0.808	0.797	0.209	0.659	0.888	0.922
	HWE*[9]	0.573	0.576	0.860	0.819	0.773	0.777	0.685	0.648	-	-
	BERT	0.505 ±0.007	0.698 ±0.003	0.674 ±0.032	0.867 ±0.009	0.822 ±0.006	0.878 ±0.014	0.388 ±0.003	0.766 ±0.007	0.836 ±0.005	0.979 ±0.001
	BERT +Q	0.493 ±0.014	0.692 ±0.002	0.681 ±0.029	0.866 ±0.003	0.797 ±0.009	0.882 ±0.015	0.365 ±0.021	0.762 ±0.007	0.841 ±0.009	0.982 ±0.004
	BERT +Q +PosNeg	0.518 ±0.005	0.699 ±0.003	0.696 ±0.004	0.873 ±0.0081	0.786 ±0.019	**0.883 ±0.001**	0.367 ±0.025	0.770 ±0.005	0.841 ±0.005	0.972 ±0.010

* = 該論文回報結果

4.3.3 Shwartz 資料集實驗結果

因為 Shwartz 資料集特別將訓練資料的分割方式分為 lexical split 和 random split，我們在訓練模型時不使用 WordNet 的上下關係資料集，而是使用 Shwartz 資料集提供的訓練資料集。輸入同樣是 (Word1, Word2)，label 為沒有上下關係 ($y = 0$) 及有上下關係 ($y = 1$) 兩種；並使用 Shwartz 資料集的 validation set 來調整模型的判斷為樣本是正是負的閾值，ROC 曲線下面積的樣本排序：BERT 模型及 BERT+Q 模型使用「模型判斷為正樣本」的機率遞減排序，BERT+Q+PosNeg 模型使用輔助句子接上 positive，且被分類為 1 的機率遞減排序，使三個模型將其認為有上下關係的樣本排序靠前。

從表4.4中可以看到, 本論文的三種模型和其他模型大致相同: random split 結果比 lexical split 高，但是與其他方法較為不同的是本論文的 precision 比 recall 還低。在 lexical split 和 Random Split 的 F1 score 上，本論文三個模型大約都相差到了 14%。BiRRE[11] 是在這個資料集上表現最好的模型，兩種分割方式的 F1 score 僅相差 4%，與其他的方法相比，突出許多。

CBOW、SGNS、GloVe 這一些沒有經過上下關係調整的方法在 Random Split 的表現上與其他經上下關係調整的詞向量相差不多，但是在 Lexical Split 上的表現並不好，而經過 Retrofit 同義詞的調整之後，CBOW 和 SGNS 在 Lexical Split 的表現有稍微變好。Poincare 為專門為有上下關係的詞語所訓練的詞向量，經過 SVM 的後的表現在 Lexical Split 上的表現比在 Random Split 的表現好。

如果單看本論文的三種模型, BERT+Q 、BERT+Q+PosNeg 模型比 BERT 模型在 random split 上表現好一點; 而在 lexical split 上, BERT+Q 較好，BERT+Q+PosNeg 次之，BERT 最差。觀察 random split、lexical split 兩種分割方式的 F1 score 差距，則以 BERT+Q+PosNeg 模型差 0.115 為最小，但仍比最好的 BiRRE 差許多。

觀察表4.3、4.4發現雖然在 random split 的資料分割方式下，本論文模型表現並不凸出，但是在 Shwartz lexical split 的 ROC 曲線下面積的表現十分的好，推測其他方法可能有 “lexical memorization” 的問題，可能僅僅學到了所謂 “典型” 的上位詞，而非詞對間是否具有上下關係。

為了驗證其他方法是否真的受到 “lexical memorization” 影響，我們將 kotlerman、bless、baroni、levy 四個測試資料集出現的字從 wordnet 上下關係訓練資料集移除，使得四個測試資料集與 wordnet 上下關係訓練資料集之間為 lexical split，並與表現較好的 Shwartz 模型、Glavas and Ponzetto 模型比較。Shwartz 模型因為硬體限制 (RAM 記憶體不足)，訓練資料僅 10000 筆；其他方法皆使用完整 lexical spilt WordNet 訓練資料集：正樣本 118264 筆，負樣本為正樣本隨機抽樣，無上下關係的詞對共 118264 筆。由表4.5的結果中可以發現：若資料切分方式為 lexical split，本論文的三種模型在 ROC 曲線下面積的結果確實比其他兩種方法好上許多。在 Kotlerman、BLESS、Baroni、Levy 四個測試資料集上，本論文模型也比其他方法較沒有 lexical memorization 的情況。

表 4.4: task1 random split vs random split evaluation result

分類	Method	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC
		Random Split				Lexical Split			
訓練詞向量過程未使用上下關係	CBOW(SVM)[16]	**0.976**	0.094	0.172	0.547	0.157	0.716	0.258	0.380
	SGNS(SVM)[2]	0.923	0.883	0.903	0.932	0.684	0.380	0.489	0.668
	GloVe(SVM)[3]	0.958	0.826	0.887	0.909	0.760	0.284	0.414	0.631
	Retrofit(SVM, CBOW)[4]	0.907	0.831	0.868	0.905	0.760	0.359	0.487	0.665
	Retrofit(SVM, SGNS)[4]	0.902	0.831	0.865	0.904	0.733	0.390	0.509	0.677
訓練詞向量過程使用上下關係	Roller and Erk*[15]	0.926	0.850	0.886	-	0.700	**0.964**	0.811	-
	Shwartz[10]	0.961	**0.961**	**0.961**	**0.989**	0.741	0.795	0.743	0.730
	Glavas and Ponzetto[18]	0.944	0.819	0.877	0.968	0.511	0.172	0.257	0.846
	HyperVec[7]	0.945	0.887	0.831	0.858	0.781	0.379	0.510	0.676
	Poincaré[8]	0.953	0.511	0.666	0.753	**0.980**	0.870	**0.922**	0.933
	LEAR[5]	0.914	0.863	0.888	0.922	0.848	0.414	0.556	0.698
	BiRRE*[11]	0.945	0.932	0.938	-	0.880	0.918	0.898	-
	BERT	0.891 ±0.019	0.788 ±0.023	0.836 ±0.005	0.979 ±0.001	0.570 ±0.019	0.889 ±0.013	0.695 ±0.015	0.924 ±0.006
	BERT +Q	0.915 ±0.011	0.777 ±0.018	0.841 ±0.009	0.982 ±0.004	0.570 ±0.015	0.936 ±0.013	0.709 ±0.013	**0.942** ±0.008
	BERT +Q +PosNeg	0.906 ±0.023	0.786 ±0.016	0.841 ±0.005	0.972 0.010	0.584 ±0.023	0.918 ±0.028	0.713 ±0.011	0.930 ±0.020

* = 該論文回報結果

表 4.5: task1 在四個測試資料集上 lexical split 結果

Model	Kotlerman				BLESS				Baroni				Levy			
	Random Split		Lexical Split		Random Split		Lexical Split		Random Split		Lexical Split		Random Split		Lexical Split	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
Shwartz[10]	**0.656**	**0.835**	0.338	0.534	0.640	**0.875**	**0.558**	0.694	0.656	0.835	0.622	0.708	**0.688**	**0.772**	**0.684**	0.717
Glavas and Ponzetto[18]	0.465	0.607	0.462	0.518	0.490	0.873	0.165	0.547	0.746	0.765	0.667	0.533	0.273	0.682	0.152	0.553
BERT	0.505	0.698	0.510	0.702	0.674	0.867	0.268	0.756	**0.822**	0.878	**0.802**	0.810	0.388	0.766	0.299	0.735
BERT + Q	0.493	0.692	0.525	**0.713**	0.681	0.866	0.364	**0.788**	0.797	0.882	0.765	**0.816**	0.365	0.762	0.330	**0.743**
BERT + Q + PosNeg	0.518	0.699	**0.535**	0.705	**0.696**	0.873	0.341	0.775	0.786	**0.883**	0.791	**0.816**	0.367	0.770	0.252	0.732

4.4 實驗二：task2 評量結果

4.4.1 評估用資料集 BLESS$_{hyper}$ 介紹

BLESS$_{hyper}$ 資料集為實驗一的 BLESS 資料集的子集，只包含實驗一 BLESS 資料集中被標示為 Hyper 的樣本共 1337 筆，BLESS$_{hyper}$ 資料集上的任務為：判斷已知有上下關係的詞對中何者為上位詞。BLESS$_{hyper}$ 資料集任務前提是已經知道有上下關係，所以不需要 task1 判斷，直接利用 task2 判斷何者為上位詞即可。

4.4.2 評估用資料集 BIBLESS 介紹

BIBLESS 資料集由 [32] 提出，包含了正向例如 (gin, alcohol)、反向如 (gin, alcohol) 及無關如 (sparrow, football) 三類，本實驗為了評量 task2 是否具有判斷上下關係方向的能力，將無關的 word pairs 移除，僅留下正向及反向 word pairs 共 1042 筆，其中正向 834 筆，反向 208 筆，正向樣本比例為 80.03%。

由於在所有資料集中，只有 BIBLESS 資料集將反向的上下關係標示出來，為了測試 task2 模型 (已知有上下關係，何者為上位詞、何者為下位詞)，特別將 BIBLESS 中無上下關係的資料移除來評量。

4.4.3 BLESS$_{hyper}$ 實驗結果

實驗結果於表 4.6，與其他方法相比表現並不是非常突出，但是也不差。本論文三種方法相比，BERT+Q 與 BERT+Q+PosNeg 相差不多，BERT 為三種方法裡最差。

表 4.6: task2 在 BLESS_{hyper} 上結果

Method	**BLESS_{hyper}**
	Accuracy
HyperVec[7]	0.92
Roller[33]	96
Le*[34]	0.94
Birre*[11]	**0.98**
BERT	0.945 ± 0.016
BERT+Q	0.962 ± 0.011
BERT+Q+PosNeg	0.959 ± 0.006

* = 該論文回報結果

4.4.4 BIBLESS 實驗結果

實驗結果如表4.7，評量標準為 ROC 曲線下面積，排序方法與實驗一相同。三種模型曲線下面積皆非常高，本論文三種模型難分軒輊，顯示若已經確定有上下關係，模型分辨何者為上位詞、何者為下位詞的能力好。

表 4.7: task2 在 BIBLESS 資料集上結果

Model	**AUC**
BERT	99.27 ± 0.181
BERT +Q	**99.56 ± 0.094**
BERT +Q +PosNeg	99.26 ± 0.366

4.5 實驗三：task1 + task2 評量結果

本實驗共使用 BIBLESS 與 HyperLex 二個資料集。BIBLESS 是從 BLESS[27] 中變形而來，詳細標籤在 section 4.5.1。HyperLex 和 BIBLESS 資料集的差別在於：HyperLex 中樣本的上下關係有程度的差異：0(無上下關係) 至 10 分 (極度具有上下關係)，而且只用這個分數，沒有明確的

標示該樣本是否具有上下關係；而 BIBLESS 資料集相反，只有是否有上下關係或是上下關係方向的標記，而沒有上下關係程度的數值。

4.5.1 評估用資料集 BIBLESS 介紹

本實驗 BIBLESS 資料集與實驗二之 BIBLESS 資料集相同：以 BLESS[27] 資料集為基礎, 共有 1976 筆, 其中關係為關係為下位詞—上位詞者為 1,上位詞—下位詞 (反向) 為 -1、分體詞 (holonym-meronym pairs, part-of relation) 及隨機名詞對皆為 0，BILBESS 資料集共有 Hyper(正向)、reversed-Hyper(反向)、others 三類，任務為判斷樣本為這三類中的哪一種。

表 4.8 為 BIBLESS 資料集的標籤比較，若是正向詞對已經出現在資料集中，則相同兩個詞的反向詞對就不會出現，反之亦然。

表 4.8: 在 BIBLESS 資料集中詞對舉例：× 代表未出現在資料集中，1 為正向詞對，0 為無關詞對，-1 為反向詞對。

example	**BIBLESS**
(fox, animal)	1
(animal, fox)	×
(fox, mouth)	0
(artifact, radio)	-1
(radio, artifact)	×

4.5.2 評估用資料集 Hyperlex 介紹

HyperLex 由 [35] 提出，包含名詞樣本 2,163 組及動詞樣本 453 組。格式為 $(Word1, Word2, Score)$，Score 代表的意義為*"To what degree Word1 is a type of Word2?"*。而因為上下關係應為不對稱的關係, (Word1, Word2) 及 (Word2, Word1) 被視為兩組不同的詞對。每一組詞對由十位真人評分，並把每一個詞對的分數各自平均，分數由 0 分 (無上下關係)

至 10 分 (極度具有上下關係)，但是並無直接標示詞對中的兩個詞是否具有上下關係。

4.5.3 Bibless 資料集實驗結果

表 4.9: task1 + task2 結果：詞對經由 task1 判斷是否有上下關係，若結果為 true，繼續由 task2 判斷何者為上位詞、何者為下位詞。

Method	**BIBLESS**		
	Accuracy	**task1 AUC**	**task2 AUC**
HyperVec[7]	0.81	0.912	0.935
Roller[33]	0.85	**0.922**	0.920
Le*[34]	0.87	-	-
BiRRE*[11]	**0.92**	-	-
BERT	0.761 ± 0.002	0.769 0.006	0.993 0.002
BERT+Q	0.757 ± 0.008	0.772 ±0.008	**0.996 ±0.001**
BERT+Q+PosNeg	0.759 ± 0.004	0.775 ±0.008	0.993 ±0.004

* = 該論文回報結果

在 BIBLESS 資料集上，在 task1 時將標示為 1 和 -1 的樣本當成正樣本，0 當成負樣本，task2 時標示為 1 的樣本為正樣本，-1 為負樣本，完全適用本論文的 task1、task2。在 BIBLESS 資料集回報的 ROC 曲線下面積使用的排序方法，都是使用該論文提出的給分方式，task1 將所有的樣本排序，task2 將「真正有上下關係」的詞對排序。

Bibless 資料集實驗結果如表 4.9，粗體為表現最佳。模型判斷為正樣本的閾值 BERT、BERT+Q 模型為 0.5 ，BERT+Q+PosNeg 模型則以 positve、negative 兩句分類為 1 機率高者為答案，由表中可以看出模型在 BIBLESS 上的準確度結果並不理想。

BIBLESS 資料集的 ROC 曲線下面積來看，task1 的表現比其他方法差很多，task2 的 ROC 曲線下面積比其他方法好。

在表 4.10 中可以發現，如果將 task1 換成 HyperVec 或是 Roller 模型，task2 用本論文模型，在 BIBLESS 資料集上的結果都變好，由原本大

表 4.10: task1 使用 HyperVec 和 Roller 模型，task2 使用本論文三種模型在 BIBLESS 資料集上 Accuracy 結果

Method	HyperVec	Roller
BERT	0.818 ± 0.012	0.832 ± 0.011
BERT + Q	0.826 ± 0.001	0.839 ± 0.002
BERT + Q + PodNeg	0.827 ± 0.005	0.841 ± 0.005

約 75%、76% 上升至 82%-84%，顯示本論文 task2 分類表現受限於 task1 結果。

最後，附上本論文三種模型 marcro、micro、weighted F1 score 如表4.11，表現最好的模型為 BERT。

表 4.11: Task1+Task2 在 BIBLESS 測試資料集結果

Model	Precision			Recall			F1		
	micro	macro	weighted	micro	macro	weighted	micro	macro	weighted
BERT	**0.762** ±0.003	0.774 ± 0.009	**0.797** ± 0.002	**0.762** ±0.003	**0.763** ±0.001	**0.761** ±0.003	**0.762** ±0.003	**0.758** ±0.003	**0.765** ±0.003
BERT+Q	0.757 ±0.008	**0.777** ±0.013	0.795 ±0.008	0.757 ±0.008	0.759 ±0.007	0.757 ±0.008	0.757 ±0.008	0.756 ±0.010	0.760 ±0.008
BERT+Q+posNeg	0.759 ±0.005	0.773 ±0.010	0.793 ±0.007	0.759 ±0.005	0.762 ±0.005	0.759 ±0.005	0.759 ±0.005	0.757 ±0.006	0.763 ±0.004

4.5.4 HyperLex 資料集實驗結果

為了與 HyperLex 的給分計算相關係數，本論文設計了以下給分方式:

$$score = log\left(\frac{Task1_p}{Task1_n}\right) + (task2\_score + 7), where$$

$$task2\_score = \begin{cases} task2\_score = log(1 * 10^{-7}), if \ \left(\frac{Task2_p}{Task2_n}\right) < 1 * 10^{-7} \\ task2\_score = log(0.9999999), if \ \left(\frac{Task2_p}{Task2_n}\right) > 0.9999999 \\ task2\_score = log\left(\frac{Task2_p}{Task2_n}\right), otherwise. \end{cases}$$

$Task1_p$、$Task1_n$ 分別表示樣本在 task1 被標為有上下關係及沒有上下關係的機率，$Task2_p$、$Task2_n$ 分別樣本在 task2 為被標為上下關係為正向及關係為反向的機率。為了使 task2(已有上下關係，判斷何者為上義詞)的分數大於 0，特別將 task2_score 設計為若小於 $1 * 10^{-7}$ 令其等於 0，若大於 $1 - 10^{-7}$ 則設為 1。最後將取對數的 task2_score 加 7 平移，以確保其數值大於 0。

表 4.12: task1 + task2 相關係數結果

Model	**Spearman's ρ**
CBOW◇[16]	0.11
SGNS◇[2]	0.11
GloVe◇[3]	0.23
Retrofit◇[4](CBOW)	0.07
Retrofit◇[4](SGNS)	0.13
JointRep◇[6][17]	0.08
HyperVec*[7]	0.540
Poincaré*[8]	0.512
LEAR*[5]	0.686
HWE◇[9]	0.46
BERT	0.686 ± 0.0034
BERT +Q	**0.690 ± 0.0048**
BERT +Q +Pos-neg	0.685 ± 0.0038

* = 該論文回報結果
◇ = HWE 回報結果

HyprLex 結果如表4.12，評量標準為 HyperLex 的分數與本論文給分方式 score 的斯皮爾曼等級相關係數 (Spearman's rank correlation coefficient)，越高越好。表中 HWE[9] 提出使用 $score_{HWE}(x, y) = dcos(\boldsymbol{x}, \boldsymbol{y}) + (|\boldsymbol{x}| - |\boldsymbol{y}|)$ 作為評分標準而得出的相關係數, CBOW、SGNS、GloVe、Retrofit、JointRep、HWE 同樣使用 $score_{HWE}(x, y)$ 評分，其中 CBOW、SGNS、GloVe 訓練詞向量時無上下關係資訊，Retrofit 使用同義詞資訊。

由表4.12 中可知 BERT、BERT+Q、BERT+Q+PosNeg 三種模型

的相關係數遠高於其他比較模型，最好的結果與 LEAR[5] 差不多。在所有模型中，BERT+Q 模型為最佳。

4.6 實驗四：task1 Pos-neg 接 task2(Q, Pos-neg, AB)

因為在 section 4.3.2中，在 Task1 上的表現 Pos-neg 模型最佳，而本論文的三種模型 (BERT、BERT+Q、BERT+Q+PosNeg) 在 task1 與 task2 的訓練皆為分開進行，因此本實驗想要了解在固定 task1 模型為 Pos-neg 的情況下，task2 用哪一個模型表現較好。本實驗使用的資料集與 section 4.5的、BIBLESS 及 Hyperlex 相同。

4.6.1 BIBLESS 資料集實驗結果

表4.13 中僅僅使用 BIBLESS 而沒有使用 BLESS_{hyper} 資料集，因為 BLESS_{hyper} 資料集如表 4.8 中所示，只需要分辨何者為上位詞 (即本論文 task2)，並不需要 task1 模型。

由表4.13中看出,固定 task1 模型的情況下,task2 使用 BERT+Q+PosNeg 模型仍然為最佳，但三種模型的準確度相差不多。

表 4.13: task1(Pos-neg) + task2 結果

Method	**BIBLESS**
	Accuracy
BERT	0.757 ± 0.005
BERT +Q	0.758 ± 0.005
BERT +Q +PosNeg	**0.759 ± 0.004**

4.6.2 HyperLex 實驗結果

由表4.14中看出，固定 task1 模型的情況下，task2 使用 BERT+Q 模型為最佳，但三種模型的與原本 HyperLex 給的分數比較得出的相關係數仍相差不多；但與 task1 使用 BERT+Q 模型、task2 也使用 BERT+Q 模型 (結果於4.12) 比較，後者較好。

表 4.14: task1(BERT+Q+Pos-neg) + task2 相關係數結果

Model	**Spearman's ρ**
BERT	0.684 ± 0.002
BERT +Q	**0.686 ± 0.001**
BERT +Q +PosNeg	0.685 ± 0.003

4.7 實驗五：task1+task2 用於樹狀結構預測

為了瞭解本論文 task1+task2 是否可以預測上下階層關係，我們使用一個小小的測試資料集，此資料集為從 wordnet 上下關係中抽取出來，有關靈長類個上下階層樹狀圖。此樹種圖共包含 107 個點和 106 條邊，為有上下關係的正樣本；另外沒有上下關係的負樣本為從點與點之間彼此的 siblings 抽樣，共 106 筆。預測的結果如表4.15，三種模型準確度都不錯 BERT 與 BERT+Q+PosNeg 比 BERT+Q 來的好。

表 4.15: WordNet 靈長類樹狀結構預測結果

Model	**Accuracy**
BERT	0.898 ±0.021
BERT+Q	0.873 ± 0.000
BERT+Q+PosNeg	**0.898±0.005**

五、 總結

5.1 結論

本論文提出了使用輔助語句將上下關係加入 BERT 語言模型進行微調的方法，調整 BERT 語言模型的預訓練詞向量，使模型具有判斷上下關係的能力。

為驗證本方法的效果，我們設計實驗證明的以此方法將上下關係加入 BERT 語言模型之 task1、task2 確實有效，使用輔助語句的 BERT + Q、BERT + Q + PosNeg 模型分類效果比僅使用詞對進行分類得 BERT 模型好。在相關系數的任務上，本論文得給分方式與 HyperLex 資料集分數之間的的斯皮爾曼等級相關係數為所有方法中最高，顯示這樣的給分方式與人們感覺的詞對上下關係程度較為相符。

在模型 task1、task2 之間的混合搭配中，首先與其他方法的模型搭配，若是以 task1 ROC 曲線下面積較好的模型 HyperVec、Roller 進行 task1，再加上本論文的 task2，相比直接使用本論文 task1+task2 準確度有明顯提升，顯示本論文 task1 分類需要加强；在與本論文模型搭配上，使用本論文在 task1 表現最好的 BERT + Q + Posneg 模型搭配其他兩種模型，在分類任務及相關係數任務上的結果與沒有混合搭配的結果相差不大。

5.2 未來展望

我們認為本論文提出的分類模型有效，但是在 task1 上的表現需要加强，在上下關係語料庫的蒐集 (如加入 WordNet 以外的上下關係資料集)、輔助語句的建構方式進行改善，有機會提升本論文效果。經過本論文調整過後的詞向量可在下游的任務上測試，以驗證調整對詞向量是否具有正面影響；另外本論文僅僅在字詞層面的任務上評估，未來也應在句子層面的下游任務上做評估。

本論文在 task1 構成的輔助語句 “A and B are hierarchically related” 是我們構成的，而 task2 的輔助語句 “A is a type of B” 是參考 HyperLex 測試資料集 [21] 給受試者的問題 “To what degree A is a type of B” 建構而成。未來可以嘗試更多不同的輔助語句構成方法，已找到 task1、task2 合適的輔助語句。我們也可以建構不同輔助語句在不同的分類任務上，如分辨同反義詞，來驗證在其他任務上輔助語句是否有效。

本論文以分別訓練 task1(是否具有上下關係)、task2(已知有上下關係，何者為上位詞、何者為下位詞) 來判斷上下關係，如果能找到合適的輔助語句，並僅進行一次分類分出無上下關係、正向上下關係、反向上下關係三類，或許能夠得到更好的分類結果。

本論文提出的方法可以延伸至用到嵌入 (embedding) 或是向量 (vector) 來表示實體的應用, 例如在電商平台能以模型判斷推薦系統要推薦的商品間是否有上下關係，並給每一個商品對應的上下關係分數，作為推薦的依據之一。

參考文獻

[1] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *arXiv preprint arXiv:1310.4546*, 2013.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[4] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, "Retrofitting word vectors to semantic lexicons," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 1606–1615.

[5] I. Vulić and N. Mrkšić, "Specialising word vectors for lexical entailment," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1134–1145.

[6] M. Alsuhaibani, D. Bollegala, T. Maehara, and K.-i. Kawarabayashi, "Jointly learning word embeddings using a corpus and a knowledge base," *PloS one*, vol. 13, no. 3, e0193094, 2018.

[7] K. A. Nguyen, M. Köper, S. S. i. Walde, and N. T. Vu, "Hierarchical embeddings for hypernymy detection and directionality," *arXiv preprint arXiv:1707.07273*, 2017.

[8] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6341–6350.

[9] M. Alsuhaibani, T. Maehara, and D. Bollegala, "Joint learning of hierarchical word embeddings from a corpus and a taxonomy," in *Automated Knowledge Base Construction (AKBC)*, 2018.

[10] V. Shwartz, Y. Goldberg, and I. Dagan, "Improving hypernymy detection with an integrated path-based and distributional method," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2389–2398.

[11] C. Wang and X. He, "BiRRE: Learning bidirectional residual relation embeddings for supervised hypernymy detection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 3630–3640. DOI: 10.18653/v1/2020.acl-main.334. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.334.

[12] C. Sun, L. Huang, and X. Qiu, "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 380–385. DOI: 10.18653/v1/N19-1035. [Online]. Available: https://www.aclweb.org/anthology/N19-1035.

[13] S. Yu, J. Su, and D. Luo, "Improving bert-based text classification with auxiliary sentence and domain knowledge," *IEEE Access*, vol. 7, pp. 176 600–176 612, 2019. DOI: 10.1109/ACCESS.2019.2953990.

[14] K. A. Nguyen, S. Schulte im Walde, and N. T. Vu, "Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 454–459. DOI: 10.18653/v1/P16-2074. [Online]. Available: https://aclanthology.org/P16-2074.

[15] S. Roller and K. Erk, "Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2163–2172. DOI: 10.18653/v1/D16-1234. [Online]. Available: https://www.aclweb.org/anthology/D16-1234.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[17] D. Bollegala, M. Alsuhaibani, T. Maehara, and K.-i. Kawarabayashi, "Joint word representation learning using a corpus and a semantic lexicon," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[18] G. Glavaš and S. P. Ponzetto, "Dual tensor model for detecting asymmetric lexico-semantic relations," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1757–1767.

[19] C. Fellbaum, "Wordnet," in *Theory and applications of ontology: computer applications*, Springer, 2010, pp. 231–243.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] I. Vulić, D. Gerz, D. Kiela, F. Hill, and A. Korhonen, "Hyperlex: A large-scale evaluation of graded lexical entailment," *Computational Linguistics*, vol. 43, no. 4, pp. 781–835, 2017.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web*, K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735, ISBN: 978-3-540-76298-0.

[24] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12 Companion, Lyon, France: Association for Computing Machinery, 2012, pp. 1063–1064, ISBN: 9781450312301. DOI: 10.1145/2187980.2188242. [Online]. Available: https://doi.org/10.1145/2187980.2188242.

[25] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

[26] L. KOTLERMAN, I. DAGAN, I. SZPEKTOR, and M. ZHITOMIRSKY-GEFFET, "Directional distributional similarity for lexical inference," *Natural Language Engineering*, vol. 16, no. 4, pp. 359–389, 2010.

[27] M. Baroni and A. Lenci, "How we blessed distributional semantic evaluation," in *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, 2011, pp. 1–10.

[28] M. Baroni, R. Bernardi, N.-Q. Do, and C.-c. Shan, "Entailment above the word level in distributional semantics," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 23–32.

[29] O. Levy, I. Dagan, and J. Goldberger, "Focused entailment graphs for open ie propositions," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 2014, pp. 87–97.

[30] M. Zhitomirsky-Geffet and I. Dagan, "Bootstrapping distributional feature vector quality," *Computational linguistics*, vol. 35, no. 3, pp. 435–461, 2009.

[31] A. Lenci and G. Benotto, "Identifying hypernyms in distributional semantic spaces," in ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 2012, pp. 75–79.

[32] D. Kiela, L. Rimell, I. Vulic, and S. Clark, "Exploiting image generality for lexical entailment detection," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, ACL; East Stroudsburg, PA, 2015, pp. 119–124.

[33] S. Roller, D. Kiela, and M. Nickel, "Hearst patterns revisited: Automatic hypernym detection from large text corpora," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 358–363. DOI: 10.18653/v1/P18-2057. [Online]. Available: https://www.aclweb.org/anthology/P18-2057.

[34] M. Le, S. Roller, L. Papaxanthos, D. Kiela, and M. Nickel, "Inferring concept hierarchies from text corpora via hyperbolic embeddings," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3231–3241. DOI: 10.18653/v1/P19-1313. [Online]. Available: https://www.aclweb.org/anthology/P19-1313.

[35] I. Vulić, D. Gerz, D. Kiela, F. Hill, and A. Korhonen, "HyperLex: A large-scale evaluation of graded lexical entailment," *Computational Linguistics*, vol. 43, no. 4, pp. 781–835, Dec. 2017. DOI: 10.1162/COLI_a_00301. [Online]. Available: https://www.aclweb.org/anthology/J17-4004.