以商品名稱為基礎之跨電商平台的商品匹配
(Product Matching Across E-Commerce Platforms Based on Product Names)

李旭清 Xu-Qing Li; 鍾聖倫 Sheng-Luen Chung
Electrical Engineering Department
National Taiwan University of Science and Technology
Taipei, Taiwan
stanley890314@gmail.com; slchung@mail.ntust.edu.tw

摘要

本研究旨在解決消費者在不同電商平台上進行比價時所面臨的商品匹配問題。為此，本研究提出了一個兩階段網路架構：eComMatch，用於跨平台商品匹配。首先，通過三元組網路訓練的限縮 (Block) 網路進行初步過濾，然後由孿生網路訓練的匹配分類 (Match) 網路進一步確定兩件商品提及是否吻合。實驗結果顯示，透過我們所微調的語意編碼器 eComBERT，本兩階段模型除了能夠在合理的時間內匹配商品，還能在顧及計算量的考量下得到良好平衡的商品匹配精確性和效率，特別適用於需要快速更新和適應不斷變化市場的電商平台。

Abstract

This study aims to address the product matching issue faced by consumers when comparing prices across different e-commerce platforms. To solve this problem, this research proposes a two-stage framework: eComMatch, for cross-platform product matching. First, a blocking filtering process is conducted by a similarity network trained by triplet network, followed by further classification through a matching network trained with a Siamese network to determine if the product mentions match. Experimental results show that, with our fine-tuned semantic encoder, our two-stage model not only matches products within a reasonable time frame but also significantly improves the accuracy and efficiency of product matching. This method is particularly suitable for e-commerce platforms that require quick updates and adaptation to the ever-changing market.

關鍵字：電商平台、實體匹配、深度學習
Keywords: E-commerce platforms, Product matching, Deep learning

1 簡介

1.1 商品匹配

「商品匹配」(Product Matching) 指的是根據商品名稱，從一組字面上未必相同的商品名稱中，識別出與其實際上是相同商品的技術。在當今的電商環境中，商品匹配的重要性日益增加。原因在於，商家為了避免消費者比價，會透過更改商品名稱或描述呈現，以增加消費者在不同電家間搜尋同件商品的困難。舉例來說：在 PChome 上有一個商品「EPSON EF-11 雷射便攜投影機」，而該商品在 MOMO 上稱作「EPSON FullHD 雷射微型 3LCD 投影機 1000 流明」，如果是以字面上來看並無法直觀的認為兩者是相同的商品。

「單一比對」(Single Matching) 是商品匹配中最單純的任務：給定一個特定的商品，而要從一組數據庫中有效找出是否有匹配的實體。而涉及像是兩不同電商平台對應同一搜尋詞搜尋結果間多對多的商品配對 (mapping)，則是較普遍的商品匹配問題。

電商平台的消費者，會在跨電商平台之間尋找競品，藉此找到最優惠的價格，而電商營運業者則有需要透過商品匹配找到競品，藉此制定更靈活的商品定價策略，或是參考其他平台同件商品營銷的優缺點來改進自家平台。

1.2 貢獻

本研究提出了一個基於商品名稱的商品匹配方法，其為基於語意編碼器，採用限縮與分類的兩階段架構。該技術依序透過限縮 (Block) 網路和匹配分類 (Match) 網路進行商品匹配，其中限縮網路利用三元組網路訓練，負責將相似商品編碼群聚，而匹配分類網路則透過孿生網路訓練，負責界定出匹配商品。據此，本論文也提出如何由標註的匹配資料合成用來訓練各別網路所需的正負樣本的取樣策略。

作為說明示例，本論文聚焦於 B2C 情境下兩個競爭性電商平台之間的商品匹配問題。為此，本研究實現一匹配資料的標註界面，透過網路搜尋從兩競業電商平台取得的兩組匹配商品集，讓標註者能夠系統性地逐一標註與貯存匹配對組，然後按本論文提出訓練兩網路所需的正負樣本採樣策略，合成對應的子網路訓練資料。

1.3 本文架構

本論文以下各章節內容概述如下：第 2 節「商品匹配」包含了相關文獻的綜合回顧，其中包括實體匹配的研究、度量學習和深度學習在商品匹配中的應用。此外還討論了實體匹配與資料標註的關聯以及兩階段匹配方法的績效分析。第 3 節「兩階段的商品匹配」深入探討了 eComBERT 預訓練模型的訓練過程以及兩階段架構的設計，並介紹了正負樣本在訓練集中的取樣策略。隨後，第 4 節「實驗與結果」展示了如何在不同 B2C 平台之間測試商品匹配的效能，詳細描述了資料集的處理方式，並根據不同資料集和預訓練模型對兩階段模型進行訓練，展示了這些模型的效果比較。最後，第 5 節「結論」對研究成果進行了總結，並探討了其應用意義。

2 商品匹配

實體匹配：在實體匹配領域，許多研究致力於提高文本中的實體與知識庫中對應條目的匹配準確性。例如 Mihalcea and Csomai (2007) 利用維基百科作為實體連結的資源，提出了一個系統，該系統通過結合 tf-idf 和 Keyphraseness 方法自動提取關鍵字，並將文本中的重要概念自動連結到維基百科的相應頁面。這一系統的自動註釋效果良好，幾乎無法與人工註釋區分。然而，儘管該系統在註釋可靠性上取得了一定成功，但仍需進一步優化以處理更複雜的實體名稱變體和多義性問題。為了解決上述挑戰，Rao et al. (2013) 提出了使用線性回歸學習方法與豐富特徵集來進行實體比對，該技術不僅有效處理了實體名稱變體和多義性，還能預測何時不應進行匹配，顯著提升了實體匹配的準確性。然而，這些基於傳統機器學習方法的技術在處理結構化數據上的表現仍未達到最佳。在這一背景下，Mudgal et al. (2018) 回顧了許多文本處理中的相關匹配任務，並提出了包括 SIF、RNN、Attention 和 Hybrid 在內的四種解決方案。他們發現，儘管深度學習在處理結構化的實體匹配問題上並未超越當前的傳統解決方案，但在面對純文本或髒亂數據時，深度學習方法顯示出了顯著的優勢。為進一步推動實體匹配的研究，Wang et al. (2021) 則建立了一個通用的實體匹配基準，用以評估不同模型的效果，為後續研究提供了標準化的評測工具。

度量學習：實體匹配技術的不斷進步也帶動了度量學習的發展。Chopra et al. (2005) 提出了一種從數據中訓練相似度度量的方法。學習過程中，通過 Siamese 的網路最小化一個判別性損失函數來推動相似度度量，使得來自同一人的人臉對的相似度變小，來自不同人的人臉對的相似度變大。該網絡的架構旨在對幾何變形具有很強的魯棒性。在此基礎上，Hoffer and Ailon (2015) 引入了 Triplet 網路模型，該模型通過比較三個樣本的距離來學習有用的數據表示，並在不使用數據增強的情況下，顯著提高了分類準確率。此外，還有其他研究將 Siamese 架構與度量學習結合，Yuan et al. (2018) 提出了一個新的推薦系統框架，通過視覺特徵與類別特徵的融合，在匹配空間中學習物品之間的距離，從而更準確地識別匹配與不匹配的物品。Shah et al. (2018) 進一步利用 Siamese 神經網絡對 fastText 進行相似性訓練，從大量自行標註的數據中學習產品匹配，為產品匹配提供了一個新的視角，這也為後續的語意模型研究打下了基礎。

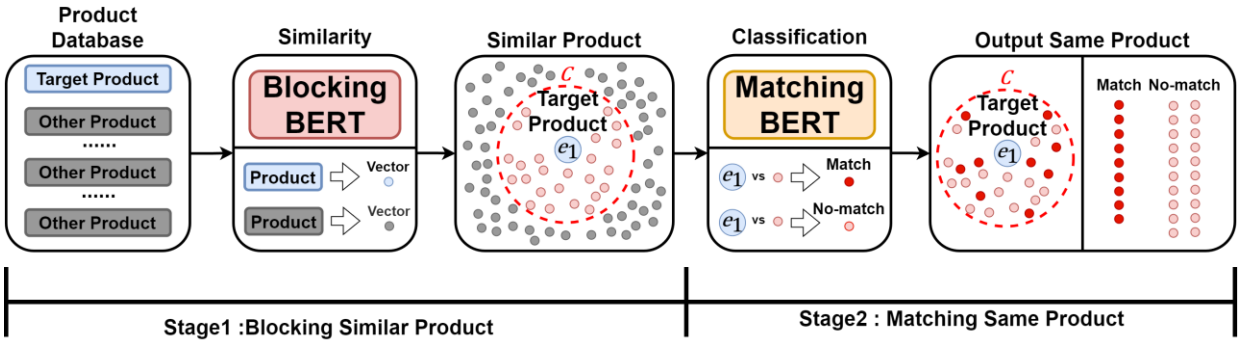

圖 1.兩階段匹配架構 eComMatch 與運作流程

語意模型：隨著度量學習技術的發展，語意模型在處理語意理解和匹配任務中的應用也越來越廣泛，並配合度量學習擁有更好的效能。最一開始，Vaswani (2017) 提出了一個基於 Attention 機制的 Transformer 語意模型，該模型能夠動態聚焦於輸入數據中最相關的部分，從而在語意理解方面展現出色的效果。這一成果為後續的 Transformer 模型奠定了基礎，BERT (Devlin, 2018) 應運而生，作為一種基於 Transformer 架構的自然語言處理模型，BERT 通過雙向編碼器設計同時考慮句子中每個詞的前後文語境，顯著提高了語意理解的精度。BERT 的強大性能主要來自於預訓練和微調兩個階段，這使得該模型在多種自然語言處理任務中表現出色。

進一步的研究中，SBERT (Reimers, 2019) 透過在 BERT 上添加 Siamese 網絡結構，使其能夠有效計算句子嵌入，並通過餘弦相似度快速計算大量句子之間的相似度，大幅提升了處理效率和效能。SBERT 在文本分類、語義搜索和對話系統中均表現卓越，並通過引入三元組和雙元組損失函數，進一步提升了模型的精度。Peeters and Bizer (2022) 進一步擴展了監督對比學習在產品匹配中的應用，並特別訓練了 BERT 模型。這顯示了對比學習在這一領域的潛力，同時也利用了多個數據集進行了測試。除此之外，Lin and Chen (2022) 提出了一種結合 ALBERT 的模型和 FastText 的模型的聯合方法 ALFA Matcher。該模型通過多頭自注意力機制篩選出關鍵信息一致的候選項，並利用 FastText 精確捕捉數據細節，在解決實體匹配任務上展現了優越的表現，進一步證明了語意模型在實體匹配中的重要性。

本論文做法：針對以商品名稱為匹配基礎的任務，我們提出了一個基於 Bert 文字編碼的兩階段的商品匹配架構：eComMatch，包括限縮 (blocking) 與匹配 (matching) 兩階段，如圖 1 所示。在 B2C 電商平台的商品匹配情境中，給定平台 A 中之一商品 e^1，我們要找出其在平台 B 中可能有多個的所有相對應的匹配商品 e^2，首先我們利用基於 Triplet Network 訓練的相似性 (Similarity) 比對網路，限縮以過濾掉 B 中較不相關的候選商品。隨後再使用基於 Siamese Network 訓練的匹配分類模型 (Matching Model) 進行匹配。針對以文字為主的商品名稱詞條，在設計以上相似性對比網路，以及匹配分類網路時，我們都採用特別電商情境文本訓練的 eComBert 作為 blocking 與 matching 這兩階段網路的預訓練模型，以有效縮小候選範圍，從而提升整體系統的性能。

，也就是先限縮再比對，兩階段比對流程如圖 1 所示。

2.1 實體匹配的定義

本研究討論 B2C 的電商情境中兩平台間的商品匹配問題。借用文獻中實體匹配的相關定義，並配合應用深度學習技術解決商品匹配時所需的資料標註與訓練、測試等的資料需求，整理出以下的符號。

符號：對應 B2C 的兩平台電商 A 與 B，我們用符號 D 與 D' 表示來自各別對應平台的商品集合，我們用 $e_1 \in D$ 以及 $e_2 \in D'$ 來表示各別商品集合中的商品實體提及 (entity mention)。明確來說：B2C 的商品匹配任務是針對經過挑選某些規則挑選後，有可能匹配的候選商品集 (candidate set)：$C \subseteq D \times D'$ 中的各 $(e_1 \in D, e_2 \in D')$ 的組合，決定 e_1 與 e_2 是否 “match” 或是 “no − match”。

訓練資料：應對以上的分類問題，本研究採取「兩階段匹配架構 eComMatch」的深度學習方法解決。一般深度學習技術，配合訓練的需求，需要先有標註資料(tag)：$T = \{(e_i^1, e_i^2, l_i)\}_{(i=1)}^{|T|}$，其中的 l_i 即為二分類的 "match", "no − match"。在本研究中，我們只選擇性地標註正樣本，即 "match" 的資料：$\{(e_i^1, e_i^2, \text{match})\}_{(i=1)}^{|T|}$。在標註資料的呈現上，我們稱每對正樣本對 (positive matched sample pair) $(e_i^1, e_i^2, \text{"match"})$ 中的 e_i^1 為 Root、e_i^2 為 Leaf，分別來自 D 與 D'。

測試資料：另一方面，為了對訓練好的模型進行測試，我們需要另外一組與訓練資料完全不重疊的測試標註資料，我們之為 $T' = \{(e_i^3, e_i^4, l_i)\}_{(i=1)}^{|T'|}$，其中 $e_i^3 \in D''$，$e_i^4 \in D'''$；為了確保訓練集與測試集中的商品不重疊，我們另外要求：D'' 代表來自平台 A 的另一個產品集，且 $D \cap D'' = \emptyset$；同時 D''' 代表來自平台 B 的另一個產品集，且 $D' \cap D''' = \emptyset$。

檢索的定義：從搜尋的角度來看，如果將平台 A 上的商品視為查詢 (query)，而平台 B 上的商品視為被檢索的文件 (document)，則上述的：$D = D_{train}^q$；$D' = D_{train}^d$；$D'' = D_{test}^q$；$D''' = D_{test}^d$。其中 D 和 D'代表的是訓練時使用的商品集，而 D'' 和 D''' 代表的是測試模型時使用的商品集，在這裡 D 與 D' 和 D'' 與 D''' 皆為互斥關係 $(D \cup D') \cap (D'' \cup D''') = \emptyset$。正樣本對 $(e_i^1, e_i^2, \text{"match"})$ 中的 e_i^1 即為 query、而 e_i^2 即為 matched document，分別來自 D 與 D'。對於標註好正樣本對 e_i^1, e_i^2, "match" 中的 Root e_i^1 即為 query，而 Leaf 的 e_i^2 即為 matched document。

2.2 匹配績效指標

在本節中，我們將詳細說明商品比對兩階段模型的評估方式和所使用的評測指標。兩階段模型 eComMatch 包括第一階段的 Blocking 模型和第二階段的 Matching 模型。每個階段的評估方式各有側重，以下將具體介紹。

Blocking 模型的評估方式：在第一階段的 Blocking 模型中，我們的主要目標是評估模型在辨識實際相同商品時的效果。我們使用 B2C_Block_test 資料集進行測試，並在第 4.2 節中詳細說明該資料集的處理流程。在評估 Blocking 模型表現時，我們採用 Recall@K 作為主要指標，其具體計算公式如下：

$$\text{Recall@K} = \frac{|\{\text{Relevant items}\} \cap \{\text{TopK retrieved items}\}|}{\{\text{Relevant items}\}} \quad (1)$$

其中 {Relevant items} 代表所有實際相同的商品集合。{TopK retrieved items}代表前 K 個檢索結果的商品集合。這個指標幫助我們評估模型在初步過濾階段的效果，確保在大量的候選商品中能夠有效地檢索到實際相同的商品。

Matching 模型和兩階段的評估方式：在第二階段的 Matching 模型當中，我們使用 B2C_Match_Test 做為測試資料。針對 B2C 的 $D \rightarrow D'$ 多對多商品匹配問題，績效是針對所有 $D \times D'$ 組合，按判定 "$match$" 或是 "$no - match$" 分類結果與實際為匹配的 $(e^1, e^2, \text{"match"})$ 所計算出的 precision, recall, accuracy 以及 F1 score 決定。

3 兩階段的商品匹配架構

在電子商務平台中，如何快速且準確地找到匹配的商品是影響用戶體驗與交易成功率的重要因素。傳統的商品匹配方法通常是直接對比候選商品，這不僅耗時還可能導致匹配效果不理想。為了解決這一問題，我們提出了一種兩階段的商品匹配架構。在我們的架構中，我們首先採用了基於 Triplet Network 的限縮模型 (Blocking Model)，以過濾掉不相關的候選商品。隨後我們使用基於 Siamese Network 的匹配模型 (Matching Model) 來進行精細匹配。這樣的兩階段設計能夠有效縮小候選範圍，從而提升整體系統的性能。

3.1 eComBERT

目前已有多種基於不同語言和領域的 BERT 預訓練模型。，在台灣 CKIP Lab 中文詞知識庫小組有提供透過 ZhWiki 與 CNA 資料集進行預訓練的模型。然而，由於領域的差異 CKIP-BERT 並不適用於中文的電子商務環境。因此本論文使用 2 億 6 千多萬筆的中文電子商務領域的文本進行再次預訓練，最終提出了一個

適應於中文電子商務文本特徵的模型 eComBERT。

這樣的設計不僅繼承了 BERT 模型的優勢，還針對中文電子商務領域的特點進行了專門調整，使模型在這一特定領域的應用中能夠展現出更高的準確度和效能。透過這一過程，本研究展示了如何通過結合預訓練和轉移學習，針對特定領域需求進行模型優化，從而提升下游任務的表現

3.2 兩階段架構: eComMatch

本文針對商品匹配問題提出了一個兩階段的網路架構 eComMatch，解決如何從集合中找到與給定商品 e^1 匹配的商品 e^2。相比於傳統逐一比對集合中的所有商品，我們所設計的架構如圖 1 所示，透過先過濾後匹配的策略提升效率。第一階段利用經 Triplet 網路訓練的限縮模型 (Block) 進行初步篩選，過濾出可能匹配的候選商品。接著，第二階段使用由 Siamese 網路訓練的匹配模型進行更精確的匹配判斷，最終確定兩個商品描述是否一致。

在這兩種網路模型中，Siamese 和 Triplet 都屬於度量學習模型，專注於透過樣本的相似性訓練，提升模型對同類樣本間區別的敏銳度。其中 Siamese 網路每次輸入一對樣本，樣本可能屬於同類或異類，以學習樣本之間的相似度或差異性。而 Triplet 網路則同時輸入三個樣本：一個錨定樣本、一個正樣本和一個負樣本，透過優化三者之間的距離來進行訓練。本文詳細說明了兩階段架構中各模型的角色，其中，第一階段的 Triplet 模型負責排序和過濾，第二階段的 Siamese 模型負責分類與最終匹配判斷。

Siamese Network (Matching 模型)：Siamese 網路的架構設計會將兩個需要比較的樣本分別輸入到共享權重的子網路中，經過相同的特徵提取過程後生成對應的特徵向量。接著，這些特徵向量會被輸入到分類器中，並最終輸出 “match” 或 “no − match” 的分類結果。由於兩個子網路之間共享相同的參數權重，該結構能確保兩個樣本在同一特徵空間中進行比較，從而提高模型對樣本相似性的敏感度及判斷精確度。

在本文提出的兩階段架構中，第二階段的 Matching 網路模型使用 Binary Cross Entropy 作為損失函數來進行訓練。Binary Cross Entropy 損失函數是一種適用於二分類問題的標準損失函數，其目標是通過最小化模型預測值與真實標籤之間的差距來優化模型。該損失函數的數學定義如下：

$$Loss = -(ylog(\hat{y}) + (1-y)log(1-\hat{y})) \quad (2)$$

Triplet Network (Blocking 模型)：在第一階段的商品初步過濾中，我們採用了 Triplet 網路結構進行模型訓練。Triplet 網路的核心思想是利用三個元素 (Anchor、Positive、Negative) 組成的樣本組合進行訓練，以優化樣本間的相對距離。這種設計能夠有效提升模型對相似與不相似樣本的區分能力。

在 Blocking 模型中，我們使用 Triplet Loss 作為損失函數進行訓練。Triplet Loss 的主要目標是通過強化錨點與正負樣本之間的距離關係，使得模型能夠正確學習樣本之間的相似性和差異性。具體來說，Triplet Loss 希望最小化錨點與正樣本之間的距離，同時最大化錨點與負樣本之間的距離。其損失函數定義如下：

$$Loss(a,p,n) = max(0, d(a,p) - d(a,n) + \alpha) \quad (3)$$

3.3 訓練集正負樣本取樣策略

eComMatch 的模型訓練是透過個別訓練 Triplet 網路與 Siamese 網路完成，而個別網路的訓練資料是由反應是否匹配的正負樣本組合而成。正負樣本的來源為先前於提到的標註資料 $T = \{(e_i^1, e_i^2, l_i)\}_{i=1}^{|T|}$。正負樣本的取樣難度直接影響訓練效果：太簡單的樣本不能有效提升發路的鑑別度；而太難的樣本又容易造成網路的學習不穩定。適當的樣本才能夠顯著提升模型的準確度。

Triplet 的訓練樣本，對應之後第 4.1 節中的 B2C_block_Train，是由三元組資料 (Positive, Anchor, Negative) 組成，而 Siamese 的訓練樣本由(Anchor, Positive) 或 (Anchor, Negative) 組成：Siamese 的訓練資料則可以從 Triplet 的樣本中切割得來。以下說明如何收集到的資料製作成 $(e^1, e^2, \text{“match”})$ 樣式的標註正樣本集，透過兩種樣本取樣策略，建構 Triplet (Block) 和 Siamese (Match) 的訓練集。

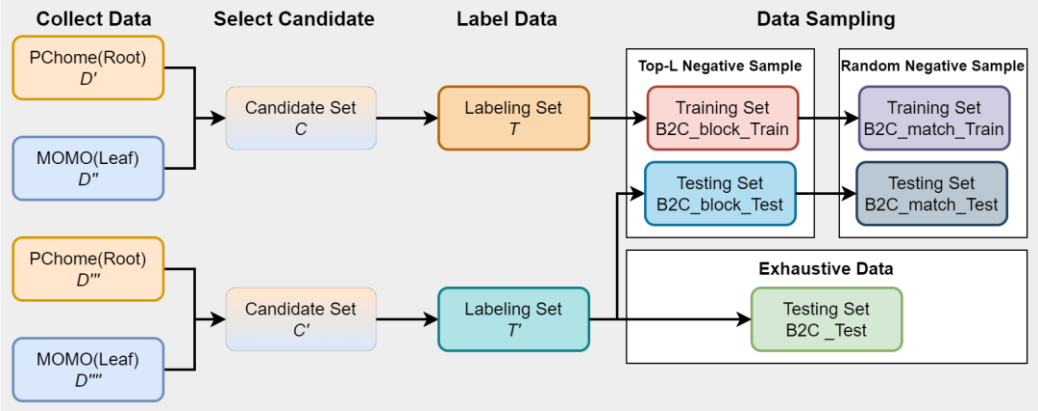

圖 2.訓練資料與測試資料處理流程

字面上最像的負樣本(Lexical Top-L Negative Sample)：首先我們透過標記先取出所有的正樣本組成(Anchor, Positive)，而在選擇負(Negative)樣本時，為了讓模型能學習認出字面上相似的負樣本商品名稱，我們採取 Lexical Top-L Negative Sample 策略，用意是為了從其他商品中利用輕量化的 BM25 算法找到字面上最相似，而實際上不同的商品作為負樣本。在這裡 Top-L 的 L 值取決於正樣本的數量。具體取樣步驟如下。

1. 詞距計算：同樣地對於每一對商品名稱，我們計算其 BM25 詞距分數。

2. 選取最相似樣本：根據計算出的 BM25 詞距，我們選取字面上最相似的前 L 筆樣本作為負樣本。

3. 樣本組合：這些負樣本將與 (Anchor, Positive) 樣本一起組成 L 筆的訓練樣本 (Anchor, Positive, Negative)，通過 Triplet Loss 來訓練模型。

簡易隨機負樣本(Simple Random Negative Sample)：用於 Siamese 分類模型的訓練資料，對應之後第四節中的 B2C_match_Train，可由先前組合成的 Triplet 的訓練樣本直接取得，並另外再使用本簡易隨機負樣本法。此方法旨在透過額外生成簡單負樣本，避免模型僅針對困難樣本進行訓練，從而提升模型在簡單情境 (例如完全不相關的商品描述) 下的泛化能力。具體來說，我們會基於已經生成的 N 筆 Triplet 訓練資料，透過以下步驟獲取兩倍於 N 的簡單負樣本 $2N$，詳細操作流程如下。

1. 正負樣本拆分：將 Triplet 的 N 筆訓練資料 (Anchor, Positive, Negative) 拆分成 N 筆 (Anchor, Positive, "match") 和 N 筆 $(Anchor, Negative, \text{“no}-\text{match”})$，總共能拆分出 $2N$ 筆訓練資料。

2. 隨機選取：對於每個 Anchor 樣本，我們從其他商品中隨機選擇兩個樣本作為簡單負樣本，總共再取得 $2N$ 筆訓練資料。

3. 樣本組合：這些簡單負樣本將與 Anchor 樣本和正樣本一起組成訓練樣本，增加了 $2N$ 筆的簡單負樣本。

4 實驗與結果

為了進行實驗，我們首先需要明確設定實驗的情境，並建立相應的資料集。準備完成後我們才能對兩階段模型的效果進行評估。

4.1 情境：跨平台的商品配對

我們以 PChome 與 MOMO 這兩個 B2C 平台為例，說明如何解決兩平台間的商品配對問題，並檢視本技術測試商品配對的成效。我們總共有四個商品集合 D、D'、D'' 和 D'''，其中訓練的商品集為 $(D \cup D')$，測試的商品集為 $(D'' \cup D''')$，為確保訓練與測試的獨立性，訓練和測試商品集為互斥關係$(D \cup D') \cap (D'' \cup D''') = \emptyset$。符號 e 則用來表示商品的具體實體提及，其中 $(e^1 \in D, e^2 \in D', e^3 \in D'', e^4 \in D''')$ 意味著每一個實體提及都是各自商品集合中的元素。

接下來，我們使用獨立於訓練數據集的商品集合 D'' 和 D''' 來製作兩階段的評估測試集。首先從商品集合 D'' 中選擇一個商品 $e_i^3 \in D''$，並從 D''' 中找到所有的相同商品，構成測試數據集，其資料的處理流程如圖 2 所示。

4.2 資料集

eComMatch 是由 Triplet (Block) 與 Siamese (Match) 兩個網路串接而成。雖然它們個別獨立訓練和驗收，但最後是否符合電商商品匹配的任務，則需要合併這兩個網路以整體 eComMatch 來看。

我們首先從 PChome 和 MOMO 這兩個電商平台上蒐集並標註了 7,691 筆商品資料，其中 3,526 筆商品 ($D \cup D''$) 為完全不同的商品 (稱為 Root)，其餘 4,165 筆商品 ($D' \cup D'''$) 則是各自與 Root 商品相同的商品，我們將這部分稱為 Leaf (如 2.1 所定義)。在這裡我們透過 D 和 D'' 標註的資料為 $T = \{(e_i^1, e_i^2, \text{match})\}_{(i=1)}^{|T|}$，透過 D' 和 D''' 標註的資料則是 $T' = \{(e_i^3, e_i^4, l_i)\}_{(i=1)}^{|T'|}$。

為了構建資料集，我們將商品依照 Root 分成訓練集與測試集，比例為 5:1。訓練集僅包含 D 與 D'，測試集僅包含 D'' 與 D'''。在這裡，我們確保訓練與測試資料集互不重疊。基於本論文所處理的 B2C 任務場景，我們最終構建了五種資料集，詳細說明如下。

B2C_Block_Train：為了獲取 Block 模型的訓練資料，我們根據 3.3 節描述的負樣本取樣方法，透過 2,938 筆商品 D 從 3,329 筆商品 D' 提取了 4,088 筆三元組資料 (Negative, Anchor, Positive)，並將其納入 B2C_Block_Train 資料集中。

B2C_Block_Test：在處理完訓練集後，考慮到 Block 模型的主要目的是排序，我們直接將資料集 D'' 和 D''' 組合形成 B2C_Block_Test 資料集。在使用這些資料時，我們選擇了 588 筆的商品資料集 D'' 作為排序檢索的查詢數據，並將其與 836 筆商品資料集 D''' 進行排序。最後通過標註結果來評估 Recall@K 的分數。

B2C_ Match _Train：針對第二階段的 Match 模型我們也需要準備資料集，除了使用 3.3 節中提到的負樣本取樣方法外，我們還根據 3.3 節描述的簡單隨機負樣本方法額外取得訓練樣本。首先我們將 B2C_Block_Train 資料集拆解為適用於 Match 分類模型的樣本，生成了 4,088 筆 (Anchor, Positive, "match") 和 4,088 筆 (Anchor, Negative, "no-match") 的數據。接著，透過 3.3 的方法我們隨機抽取了 $2 * 4{,}088 = 8{,}176$ 筆負樣本，與前述的數據結合，最終構建了包含 16,352 筆訓練資料的 B2C_Match_Train 資料集中。所有用來訓練與測試 eComMatch 兩階段的資料都是由我們針對 PCHome 與 Momo 兩平台上爬文，然後經由人眼比對後所產的 4,165 筆的正樣本對 (positive matched sample pair) $(e_i^1, e_i^2, \text{“match”})$ 所組成的 $T = \{(e_i^1, e_i^2, l_i)\}_{(i=1)}^{|T|}$，再經由上節中的正負樣本取樣策略所組成。

B2C_ Match _Test：同樣地，我們可以從 D 和 D' 資料集中，依照 3.3 提到的方法透過 588 筆商品 D'' 從 836 筆商品 D''' 提取了 1,613 筆三元組樣本。再來將其拆解為 1,613 筆 (Anchor, Positive, "match") 和 1,613 筆 (Anchor, Negative, "no-match") 的數據，並將其透過 3.3 的方法抽取 3,226 筆負樣本，最終獲得了 6,452 筆 Match 模型的測試資料，組成了 B2C_Match_Test 資料集。

B2C_ Test：最後是我們的兩階段模型評估的測試集 B2C_Test。我們透過兩個商品集 D'' 和 D''' 來製作我們的兩階段評估測試集 B2C_Test。首先我們從商品集 D'' 中取出一個商品 $e_i^3 \in D''$，然後標註從 $e^4 \in D'''$ 找出所有相同的商品，組成(Anchor,Positive,"match")，而剩下不同的商品則組成 (Anchor, Positive,"no-match")，將此流程經過 i 次就能得到所有的測試資料。

由於我們的 D'' 總共有 588 筆資料，而 D''' 有 836 筆資料，於是就能夠得到 588*836=491,568 筆測試兩階段績效的資料，也就是 B2C_Test 資料集。資料型格式為 $\{(e^3, e^4, l) | l \in (\text{“match”}, \text{”no-match”})\}$。整體的資料集如表 1 所示 (見第 2 頁)。

4.3 實驗結果

本章節旨在呈現不同模型在商品比對任務中的表現。我們針對 B2C 的情境並評測第一階段中不同的 K 值對於模型的影響，以及使用兩階段和單一階段模型的運算時間和準確度，藉此驗證兩階段的必要性。

資料集名稱	資料數量	資料格式
T(“PChome”, “Momo”)	Root :3,526 Leaf:4,165	$(D\ U\ D')$, $(D''$ U $D''')$, $T = \{(e^1, e^2, \text{“}match\text{”})\}$
B2C_block_Train	4,088 筆	D, D' Triplet (Negative, Anchor, Positive)
B2C_block _Test	588 個 Query 836 個 Leaf	D'', D''' 關聯資料 (Product Name, Connect)
B2C_match _Train	16,352 筆	D, D' 分類標籤 (Anchor, Positive, "match") U (Anchor, Negative, "no-match")
B2C_match_Test	6,452 筆	D'', D''' 分類標籤 (Anchor, Positive, "match")
B2C _Test	491,568 筆	D'', D''' $\{(e^3, e^4, l) \mid l \in (\text{"}match\text{"}, \text{"}No\_match\text{"})\}$

表 1. 針對不同任務的資料集

B2C 兩階段不同預訓練模型實驗：在這個實驗中，我們比較由中文新聞和維基百科資料訓練的 CKIP-BERT 和我們自己透過商品名稱訓練的 eComBERT 作為預訓練模型於兩階段模型的效果。

我們使用 B2C_Block_Train 資料集作為訓練第一階段 Blocking 模型的資料集，並使用 C2C_Match_Train 資料集作為第二階段 Matching 模型的訓練資料，最後我們將其測試在 B2C_Test 的資料集上，在這裡第一階段過濾的商品數量 $K = 50$，實驗結果如下表 2 所示。

	Accuracy	Recall	Precision	F1
eCom BERT	0.942	0.981	0.872	0.923
CKIP BERT	0.928	0.827	0.735	0.778

表 2. B2C 兩階段不同預訓練模型實驗數據表

B2C Blocking 不同預訓練模型實驗：我們針對第一階段的 Blocking 模型使用 B2C_Block_Train 訓練，在這個實驗中我們主要關注 Recall@K 的指標，因為在此我們更注重模型是否能幫我們找出實際上相同的商品。同樣的我們也會比較不同預訓練模型的效果。實驗結果如下表 3 所示。

Recall	@1	@5	@10	@20	@50
eComBERT	0.765	0.851	0.893	0.927	0.998
CKIP BERT	0.635	0.824	0.879	0.939	0.996

表 3. B2C Blocking 不同預訓練模型實驗數據表

B2C Matching 不同預訓練模型實驗：在第二階段的 B2C Matching 模型實驗中，我們使用 B2C_Match_Train 進行訓練，並用 B2C_Match_Test 測試模型，在這裡我們也比較使用不同預訓練模型的效果。實驗結果如下表 4 所示。

	Accuracy	Recall	Precision	F1
eCom BERT	0.982	0.992	0.876	0.930
CKIP BERT	0.926	0.974	0.758	0.853

表 4. B2C Matching 不同預訓練模型實驗數據表

兩階段不同 K 值對比實驗：

在第一階段中過濾的商品數量 K 會對第二階段的效能產生顯著影響，因此我們針對不同的 K 值進行實驗，這些 K 值代表第一階段過濾後傳遞給第二階段的商品數量，我們將我們訓練的模型測試在 B2C_Test 測試集上，其結果如下表 5 所示。

	Accuracy	Precision	Recall	F1
K=10	0.946	0.976	0.873	0.922
K=50	0.942	0.981	0.872	0.923
K=100	0.945	0.943	0.853	0.896

表 5. 兩階段不同 K 值對比實驗數據表

兩階段與單階段績效對比實驗：商品匹配的方法可分成以下三種，分別是：(1) Only Similarity：將所有的商品透過 BERT 模型轉換成向量之後，透過相似度計算來尋找相似商品，並使用閥值來判定是否為相同商品，該方法只需要將所有商品推論一次即可；(2) Only Classify：將兩個商品名稱分別作為

BERT 的輸入，並利用輸出 [CLS] Token 通過分類器來判斷商品是否相同，此方法需要對所有商品進行 BERT 模型的推理；(3) Two Stage：本論文採用的兩階段匹配方法。

由於本論文所採用兩階段方法的原因主要是計算時間以及匹配精確度的協調。為此，我們分別針對使用相似度、分類器以及兩階段的作法，計算單一個商品 $e_i^1 \in D$ 在 10,000 筆的 $e^2 \in D'$ 中尋找並辨識相同商品所需的時間，以及利用 B2C_Test 測試集測試所得的 F1 Score 分數。下表 6 為對比數據表。

	Time Type	10,000 Products	F1 Score
Only Similarity	Inference Time	35.091s	86.709
	Similarity Dot Product Time	2.061s	
Only Classify	Inference Time	1002.971s	95.231
Two Stage	Inference Time	19.353s	93.893
	Similarity Dot Product Time	0.2087s	
	Classify Time	11.7073s	

表 6.兩階段與單階段績效對比實驗對比表

實驗結果分析：根據實驗結果，我們可以明確地發現，基於 eComBERT 微調的模型在效能上超過基於 CKIP BERT 微調的模型許多。因此對於不同領域的 In-Domain 預訓練對於模型的下游任務尤為重要。此外，針對兩階段的方法，雖然單純使用分類模型的效果最佳，但運算時間非常長，無法滿足商品比對的即時性需求。而僅使用語意相似度的搜尋方法，雖然速度最快，但效果明顯不如兩階段方法。綜合來看，我們選擇的兩階段方法能夠在相對較短的時間內，提供效果適中的解決方案，是較為理想的選擇。

5 結論

在電子商務領域，構建精準的產品匹配模型需要大量高質量的訓練數據，但現有資料集無法滿足台灣電商市場的需求。為解決這一問題，我們設計了一個專門用於電商商品收集與標註的平台，可以自動化地從不同電商平台搜尋並對比商品，進行精確的匹配與標註，生成有效的正負樣本。

雖然目前已有多種基於不同語言和領域的 BERT 預訓練模型，但它們不完全適用於台灣電商環境，例如 CKIP-BERT 使用通用中文語料進行預訓練，無法捕捉電商文本中的專業特徵。因此，我們使用 2.6 億筆中文電商文本進行重新預訓練，提出了針對電商領域優化的預訓練模型 eComBERT。

基於 eComBERT，我們進一步設計了一個專為電商商品匹配問題量身打造的兩階段架構 eComMatch。該架構包含兩個模組：在第一階段，我們採用結合 Triplet 結構的 BERT 模型進行初步過濾，快速篩選潛在匹配商品；在第二階段，我們利用 Siamese 網絡結構進行精確匹配，判斷篩選後商品之間的相似性。這樣的設計提升了匹配效率與準確度，為電商商品匹配領域的研究與應用奠定了堅實的基礎。

References

Chopra, S., Hadsell, R., & LeCun, Y. (2005). *Learning a similarity metric discriminatively, with application to face verification.* Paper presented at the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05).

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hoffer, E., & Ailon, N. (2015). *Deep metric learning using triplet network.* Paper presented at the Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3.

Lin, Z., & Chen, W. (2022). *ALFA Matcher: Supervised and Unsupervised Product Entity Matching Model.* Paper presented at the 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA).

Mihalcea, R., & Csomai, A. (2007). *Wikify! Linking documents to encyclopedic knowledge.* Paper presented at the Proceedings of the sixteenth ACM conference on Conference on information and knowledge management.

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., . . . Raghavendra, V. (2018). *Deep learning for entity matching: A design space exploration.* Paper presented at the Proceedings of the 2018 international conference on management of data.

Peeters, R., & Bizer, C. (2022). *Supervised contrastive learning for product matching.* Paper presented at the Companion Proceedings of the Web Conference 2022.

Rao, D., McNamee, P., & Dredze, M. 2013. Entity linking: Finding extracted entities in a knowledge base. *Multi-source, multilingual information extraction and summarization*, 93-115.

Reimers, N. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

Shah, K., Kopru, S., & Ruvini, J. D. (2018). *Neural network based extreme classification and similarity models for product matching.* Paper presented at the Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers).

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, J., Li, Y., & Hirota, W. (2021). *Machamp: A generalized entity matching benchmark.* Paper presented at the Proceedings of the 30th ACM International Conference on Information & Knowledge Management.

Yuan, H., Liu, G., Li, H., & Wang, L. (2018). *Matching recommendations based on siamese network and metric learning.* Paper presented at the 2018 15th International Conference on Service Systems and Service Management (ICSSSM).