

國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

使用類別映射的零樣本文本分類

Category Mapping for Zero-shot Text Classification

張秋霞

ZHANG, QIUXIA

指導教授：張智星 博士

Advisor: Jyh-Shing Roger Jang Ph.D.

中華民國 112年 7 月

July, 2023

國立臺灣大學碩士學位論文

口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY



使用類別映射的零樣本文本分類

Category Mapping for Zero-shot Text Classification

本論文係張秋霞君（學號 R10922164）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 112 年 7 月 28 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 28 July 2023 have examined a Master's thesis entitled above presented by ZHANG, QIUXIA (student ID: R10922164) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

張智星

(指導教授 Advisor)

陳經懷

蔡子翔

系主任/所長 Director:

洪士瀨





致謝

當我在鍵盤輕敲下“致謝”二字，已是論文終末，欲書之情化作無盡思緒。或因感激之人過多，或因辭藻匱乏，驀然間竟不知該如何表達。

毋庸置疑，前人是最應當被感激的。淺析 NLP 的歷程，幾十年光陰，革新跌宕，若非前人播下的種，我如何能得到這篇在前人研究基礎之上的果？

還需感念我親愛的導師，張智星教授，他在繁忙的工作中依然抽出時間與我們交流，給予點撥，亦常常鼓勵我們敞開眼界，關注個人成長。他的睿智風範啟我學術成長，更啟我處世之道。

其次，要感謝實驗室同儕張開、炫均。他們在研究中毫不吝嗇地給予幫助與支持，又在生活中給予最真誠的陪伴。我們共同努力，共同進步，成為學術上的夥伴，更成為生活中的摯友。

同時，亦需衷心感謝 NLP 組的同學得郁、德倫、家雯、宇萌。與他們的交流與討論，為我的論文研究與未來學術生涯帶來了寶貴助益。

最後，我要特別感謝父母、姐姐，以及朋友麗娜、陳倩，大專老師志偉哥。我的求學之路曾風雨飄搖，正是因為他們的鼓勵與支持，才得以努力前行，不斷成長，學會自信地面對未來與挑戰。

僅以此致謝。





摘要

現有基於大型的預訓練模型並加入提示進行零樣本文本分類的方法，具有模型自身強大的表示能力和擴展性，但商業可用性相對較差。利用類標籤和已有資料集微調較小的模型進行零樣本分類的方法相對簡便，但存在模型泛化能力較弱等問題。本文使用了三種方法來提高預訓練模型在零樣本文本分類任務上的準確性和泛化能力：1. 使用預訓練語言模型，將其輸入整理成統一的多項選擇格式；2. 利用維基百科文本數據構建文本分類訓練集，對預訓練模型進行微調；3. 提出了基於 GloVe 文本相似度的零樣本類別映射方法，使用維基百科類別代替文本類別。不使用待分類標籤進行微調的情況下，該方法取得了與使用待分類標籤進行微調的最佳模型相當的效果。

關鍵字：自然語言處理、預訓練語言模型、零樣本文本分類、分類、GloVe





Abstract

The existing method of using large pre-trained models with prompts for zero-shot text classification has powerful representation ability and scalability. However, its commercial availability is relatively poor. The method of using class labels and existing datasets to fine-tune smaller models for zero-shot classification is relatively simple, but it may suffer from weaker model generalization ability. This paper proposes three methods to improve the accuracy and generalization ability of pre-trained models in zero-shot text classification tasks: 1) using pre-trained language models and formatting inputs into a unified multiple-choice format; 2) constructing a text classification training set using Wikipedia text data and fine-tuning the pre-trained model; and 3) proposing a zero-shot category mapping method based on GloVe text similarity, using Wikipedia categories to replace textual categories. Without using labeled samples for fine-tuning, the proposed method achieves results comparable to the best models fine-tuned with labeled samples.

Keywords: Natural Language Processing, Pretrained Language Models, Zero Shot Text Classification, Classification, GloVe





目錄

	Page
口試委員審定書	i
致謝	iii
摘要	v
Abstract	vii
目錄	ix
圖目錄	xiii
表目錄	xv
第一章 緒論	1
1.1 研究動機	1
1.2 研究貢獻	3
1.3 章節概述	3
第二章 文獻探討	5
2.1 詞向量	5
2.1.1 靜態詞向量模型	6
2.1.1.1 LSA	6
2.1.1.2 Word2vec	6
2.1.1.3 GloVe	7
2.1.2 動態詞向量模型	9

2.2	語言模型	9
2.2.1	傳統語言模型	10
2.2.2	基於神經網路的語言模型	11
2.2.3	預訓練語言模型	11
2.2.4	基于 Transformer 的預訓練語言模型	14
2.2.4.1	GPT	14
2.2.4.2	BERT	17
2.2.4.3	ALBERT	19
2.2.4.4	RoBERTa	19
2.2.4.5	DeBERTa	21
2.3	現有的零樣本文本分類模型	21
2.3.1	大型生成式語言模型 + 基於提示的方法	22
2.3.2	較小預訓練語言模型 + 微調的方法	23
2.3.2.1	基於自然語言推理的方法	23
2.3.2.2	UniMC	26
第三章	資料集介紹	29
3.1	Yahoo! Answers	29
3.2	AG News	31
3.3	DBpedia	31
3.4	IMDB	32
第四章	研究方法	35
4.1	模型微調	35
4.1.1	開放域訓練資料的獲取	36
4.1.2	模型輸入格式處理	38
4.2	類別映射	41

4.2.1	類別映射前置處理	41
4.2.2	類別映射	42
4.2.3	替代詞列表使用	44
4.2.4	替代詞篩選機制	45
第五章	實驗設計和結果討論	47
5.1	實驗任務	47
5.2	實驗流程及設定	48
5.2.1	實驗流程	48
5.2.2	實驗設定	49
5.3	實驗結果	50
5.3.1	實驗 1：零樣本文本分類模型性能對比實驗	50
5.3.2	實驗 2.1：使用維基百科資料微調模型前後效果比對實驗	52
5.3.3	實驗 2.2：訓練資料類別數量探究實驗	53
5.3.4	實驗 3.1：替代詞列表效果探究實驗	55
5.3.5	實驗 3.2：篩選機制效果實驗	56
5.3.6	實驗 4：維基百科微調與類別映射消融實驗	58
5.3.7	實驗 5.1：研究方法使用前後模型性能對比實驗	59
5.3.8	實驗 5.2：UniMC-WiKi 與最佳模型性能對比實驗	60
第六章	結論和未來工作	63
6.1	結論	63
6.2	未來工作	64
	參考文獻	67





圖目錄

2.1	Transformer 模型架構	13
2.2	GPT 的模型結構及不同下游任務的拼接方式	15
2.3	GPT-3 少樣本學習範例	16
2.4	BERT 模型的預訓練及微調	17
2.5	BERT 輸入的處理	18
2.6	GPT-3 中給出的零樣本翻譯任務提示範例	22
2.7	TE-Wiki 模型結構	24
2.8	UniMC 模型結構	26
2.9	UniMC 模型微調所用的 14 個資料集	27
4.1	類別樹構建示意圖	37
4.2	正負例樣本選擇示意圖	37
4.3	在詞向量空間上將維基百科類別按目標類別歸類示意圖	43
4.4	將維基百科類別映射為目標類別候選項示意圖	43
4.5	替代詞使用示意圖	45
5.1	零樣本文本分類模型性能對比實驗結果條形圖	52
5.2	零樣本文本分類模型性能對比實驗結果條形圖	53
5.3	零樣本文本分類模型性能對比實驗結果條形圖	54
5.4	替代詞列表效果探究實驗條形圖	55
5.5	目標類別與替代詞相似度實例	56
5.6	篩選機制效果實驗結果條形圖	57
5.7	維基百科微調與類別映射消融實驗結果條形圖	59
5.8	研究方法使用前後模型性能對比實驗結果條形圖	60

5.9 UniMC-WiKi 與最佳模型性能對比實驗結果條形圖	61
---	----





表目錄

3.1	用於評估的資料集概況	29
3.2	Yahoo! Answers 資料集類別	30
3.3	Yahoo! Answers 樣本字段樣例	30
3.4	AG News 資料集類別	31
3.5	AG News 樣本字段樣例	31
3.6	Dpedia 資料集類別	32
3.7	DBpedia 樣本字段樣例	32
3.8	IMDB 資料集類別	32
3.9	IMDB 樣本字段樣例	33
5.1	實驗環境	50
5.2	各模型參數	50
5.3	零樣本文本分類模型性能對比實驗結果	51
5.4	使用維基百科資料微調模型前後效果比對實驗	52
5.5	訓練資料類別數量探究實驗結果	54
5.6	替代詞列表效果探究實驗結果	55
5.7	篩選機制效果實驗結果	57
5.8	加入篩選機制之後每個類別對應平均替代詞列表長度	57
5.9	維基百科微調與類別映射消融實驗結果	58
5.10	研究方法使用前後模型性能對比實驗結果	59
5.11	UniMC-WiKi 與最佳模型性能對比實驗結果	60





第一章 緒論

1.1 研究動機

文本分類是自然語言處理領域的一項任務，在垃圾郵件過濾、信息檢索、個性化推薦、情感分析、輿情監測等領域中都有著廣泛的應用。然而，文本分類任務中往往需要面對資料集中存在未知類別、大量多樣性和複雜性文本資料的情況，這為模型的訓練和分類帶來了難度。在實際應用中，由於資料采集和標注的成本較高，隨著任務所涉及的文本類型和類別增多，標記資料的數量必然會變得更加有限，這導致了傳統的文本分類任務會面臨著資料稀疏性的問題。

為了解決這些問題，關於零樣本文本分類的研究應運而生。零樣本文本分類是一種基於先驗知識或是無監督學習等方法對文本進行分類的技術，與傳統的文本分類不同，它不需要任何預先標記的訓練樣本，而是通過將已有的歸類信息的能力轉移到未知的新類別上，從而實現文本的分類。這種技術的出現，不僅避免了大量手動標記訓練資料的工作，而且可以從已有的資料中提取新的知識來應對文本分類中未知類別的問題。通過研究和實踐零樣本文本分類，不僅可以使模型具備更好的遷移性和可擴展性，還可以加速人工智能和自然語言處理技術在各個實際應用場景的落地，這為文本分類技術的發展提供了未來的方向和動力。

現有的較為先進的零樣本文本分類方法依據模型大小的分類，主要分為兩

類。一類是基於提示 [2]，通過在大型預訓練模型中加入一定的提示信息，如在 GPT-3 的輸入中加入類別相關的關鍵詞或句子，這樣能夠更好地理解和區分不同類別的文本資料，從而提升模型對特定分類任務的理解能力。InstructGPT[16] 和 chatGPT 是這種方法的變體，它們通過利用人類輸入的指令和反饋來改進模型，通過這種方法，模型可以學習到更全面的語言模式和知識，並能夠在新的未知類別下快速進行分類。

另一類方法則是利用已有資料集中的類標籤和豐富的先驗知識，對較小的分類模型進行有針對性的微調。例如，UniMC[23] 通過將自然語言處理領域中大量已有的資料集作為輸入文本，並將輸入文本轉換成多項選擇的格式用於預訓練模型的微調，也有文章提出將已有的資料轉換成蘊含對的方式對預訓練模型進行微調。通過這種方法，模型可以利用先前獲得的知識和先驗信息，對新的資料進行更準確的分類。

需要指出的是，這兩種方法都有其獨特的優點，也有一定的限制。基於大型的預訓練模型並加入提示的方法，具有模型自身強大的表示能力和擴展性，但也存在著對計算資源的依賴性，其效率取決於模型接口調用速度，在實際中商用的可用性相對較差；而利用類標籤和已有資料集微調較小的模型的方法，雖然相對於前者更為簡便，但也存在著模型泛化能力較弱等問題。

本研究基於以下三個方法，一定程度地提升了模型準確性，並提高了模型的泛化能力，該方法在微調不參考待分類標籤的情況下，達到了與 5 月剛被提出的使用了待分類標籤進行微調的最佳模型相當的效果：

- 使用維基百科的文本資料來構建分類訓練集，充分利用現有知識，提升預訓練模型分類能力。
- 提出了一種基於 GloVe 文本相似度的零樣本類別映射方法，將零樣本文本類

別與維基百科類別進行映射，進而將已有知識遷移到零樣本文本分類中。

- 使用統一的多項選擇格式作為輸入格式，以提升模型對各分類類別的感知。

1.2 研究貢獻

本篇論文的主要貢獻如下：

1. 提出了基於 GloVe 文本相似度的零樣本類別映射方法，將其映射至維基百科類別中；
2. 本研究提出的方法達到了與目前最佳方法相當的效果。

1.3 章節概述

本文分為六個章節，內容如下：

- 第一章為緒論：介紹本文的主題、研究動機與研究貢獻。
- 第二章為文獻探討：主要介紹零樣本文本分類的相關研究及方法，以及本文中會使用到的模型。
- 第三章為資料集說明：介紹實驗研究中使用的資料集。
- 第四章為研究方法介紹：詳細介紹本文使用的方法及理論。
- 第五章為實驗設計與結果：詳細介紹實驗任務及實驗的設計，並通過視覺化的方式呈現實驗結果。
- 第六章為結論與未來展望：對實驗內容及結果進行總結，並且討論未來可能之改進。





第二章 文獻探討

本章節將會回顧本研究中使用到的相關模型及知識背景，其中重點介紹預訓練語言模型的發展及分類，以及現有的基於預訓練模型的文本分類技術。

2.1 詞向量

詞向量 (word vector) 也被稱為詞嵌入 (word embedding)，是一種將離散的詞語映射到連續的向量空間中的表示方法。通俗地說，詞向量就是將每個詞語表示成多維向量的形式，這樣可更方便地對文本進行處理和分析。

在以往的文本處理方式中，曾用獨熱編碼 (one-hot encoding)、詞袋模型 (bag-of-words) [8]、TF-IDF 模型 [20] 等方法來表示文本，但用獨熱編碼表示存在維數高、數據稀疏的問題；詞袋模型無法捕捉單詞之間的順序資訊，忽略了單詞之間的文本結構；TF-IDF 模型只考慮了單詞在檔案中的頻率，忽略了單詞的上下文資訊。

詞向量最早是作為語言模型的中間產物出現的。在 Bengio 提出的神經網路語言模型 [1] (NNLM, Neural Network Language Model) 的學習過程中，得到了每個單詞對應的詞向量，這些詞向量被存儲在模型的參數權重矩陣，即詞嵌入 (word embedding) 矩陣中，採用具有一定維度的實數向量來進行單詞的分布式表

示 (distributed representation)，相比起傳統的文本表示方式，詞向量可以從語義層面上反映出單詞之間的關係和相互作用，同時解決了維度災難和資料稀疏性的問題。

隨著詞向量的出現和應用，越來越多的研究者開始探索如何更加高效並準確地獲得詞向量。根據生成方式的不同，詞向量可以分為靜態詞向量和動態詞向量。靜態詞向量的模型在進行時，將每個單詞表示成一個固定的向量，該向量在整個訓練過程中保持不變；而動態詞向量則是基於詞語的上下文動態生成的詞向量，在訓練過程中會根據上下文的不同而產生變化。

2.1.1 靜態詞向量模型

2.1.1.1 LSA

LSA [6] (Latent Semantic Analysis) 的基本思想是基於全域語料，通過奇異值分解 (SVD) 來處理文本資料矩陣，以減少文本資料矩陣的維度，以便更好地理解和分析文本之間的關係，捕捉其潛在的語義特徵。LSA 採用了全域共現信息 (global co-occurrence information)，能夠有效地利用統計信息，但其矩陣分解計算複雜度高，同時在詞類推 (word analogy) 任務上表現相對較差。

2.1.1.2 Word2vec

Word2vec [15] 是基於 NNLM 的改進和優化。該模型的目的是通過大量的無監督資料來訓練得到一個更好的詞向量矩陣，其訓練模型有兩種：CBOW (Continuous Bag-of-Words) 和 Skip-Gram，CBOW 的目標是根據上下文中的單詞 $context(w)$ 預測當前單詞 w ，它將上下文中的詞向量加起來，並通過一個淺層神經網路來預測給定詞的機率。Skip-Gram 的目標是基於當前單詞 w 來預測上下文

中的單詞 $context(w)$ 。它將當前單詞的詞向量作為輸入，通過一個淺層神經網路來預測上下文單詞的機率。相比 NNLM，採用 Skip-Gram 和 CBOW 兩種模型來學習單詞的嵌入表示，可以更好地捕捉到單詞之間的上下文關係，從而使得生成的嵌入向量具有更好的語義表示能力。

此外，為了加快模型對大量單詞的訓練速度，Word2vec 提出了兩種優化方法：Hierarchical Softmax 和 Negative Sampling。Hierarchical Softmax 採用了二叉樹結構，使用一個樹形結構將每個單詞的機率表示為該單詞出現在二叉樹的某條路徑上的每個節點的條件機率乘積。分類單詞的時間複雜度從原先的 $O(V)$ 降低為 $O(\log V)$ ，有效地提高了效率。Negative Sampling 則是採用了一種二分類的策略，其過程與 softmax 函數非常類似，但是不是對所有單詞都計算機率，而是對當前單詞正樣本和 K 個負樣本（負樣本是根據某種分佈隨機採樣的）進行計算，使得訓練過程中的計算複雜度從 $O(V)$ 降低到了 $O(K)$ 。

與 LSA 不同，Word2ve 基於局部共現信息（local co-occurrence information）進行訓練，效率較高，但在利用語料庫信息方面有欠缺，並且其語義表示能力相對較弱。

2.1.1.3 GloVe

GloVe（Global Vectors for Word Representation）的思想是將重點放在全域詞頻的信息上，而不是局部窗口內詞頻的信息，它嘗試在保留詞之間的語法和語義關係的同時，優化詞向量的聚類效果。與 Word2vec 不同，該模型把窗口內的所有共現詞對（co-occurrence word pairs）作為一個輸出向量，通過對共現矩陣（co-occurrence matrix）進行全域矩陣分解，從而最小化某種損失函數，學習每個詞的向量表示。

在 GloVe 模型中，定義詞匯表大小為 V ，若需要學習每個單詞的 d 維向量表示，我們可以將這些向量構成一個矩陣 W ，其中每列對應一個單詞的向量表示。對於任意兩個單詞 i 和 j ，統計它們在全部訓練預料中，在一個固定大小的上下文窗口內共同出現的次數 X_{ij} ，定義 P_{ij} 為單詞 i 和 j 的共現機率，若語料中存在三個單詞 i 、 j 、 k ，當 P_{ik}/P_{jk} 的值很大時，表示 k 和 i 的相關程度比 k 和 j 的相關程度高，若 P_{ik}/P_{jk} 的值更接近 1，表示 k 和 i 的相關程度與 k 和 j 的相關程度相比差不多。GloVe 模型的目標就是設計一個函數 F 來學習這個比值：

$$F(w_i, w_j, \tilde{w}_k) = P_{ik}/P_{jk} \quad (2.1)$$

公式中的 w_i 、 w_j 、 \tilde{w}_k 分別表示 i 、 j 、 k 的詞向量，使用點積運算可簡化得到一下公式：

$$F((w_i - w_j)^T \tilde{w}_k) = F(w_i^T \tilde{w}_k - w_j^T \tilde{w}_k) = P_{ik}/P_{jk} \quad (2.2)$$

指定滿足上式的函數 $F = \exp$ ，得到：

$$\exp(w_i^T \tilde{w}_k - w_j^T \tilde{w}_k) = \frac{\exp(w_i^T \tilde{w}_k)}{\exp(w_j^T \tilde{w}_k)} = P_{ik}/P_{jk} \quad (2.3)$$

令 $\exp(w_i^T \tilde{w}_k) = P_{ik}$ 、 $\exp(w_j^T \tilde{w}_k) = P_{jk}$ ，得到：

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i) \quad (2.4)$$

定義兩個長度為 V 的向量 b_i 和 \tilde{b}_k ，分別表示 i 、 k 在不同上下文中的偏置， \tilde{b}_k 由如下公式計算得到：

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}) \quad (2.5)$$

使用偏置 b_i 將 $\log(X_i)$ 替換，引入權重函數 $f(X_{ij})$ ，並使用加權最小二乘回歸模型進行計算，最終得到目標函數：

$$J = \sum_{i,j=1}^V f(X_{ik}) \left(w_i^T \tilde{w}_k + b_i + \tilde{b}_k - \log(X_{ik}) \right)^2 \quad (2.6)$$

其中引入權重函數的目的是為了平衡高頻詞和低頻詞對目標函數的影響，該函數的計算如下：

$$f(x) = \begin{cases} (x/x_{\max})^\alpha, & \text{if } x < x_{\max} \\ 1, & \text{otherwise} \end{cases} \quad (2.7)$$

GloVe 融合了 LSA 和 Word2vec 的優點，採用了詞共現矩陣（co-occurrence matrix）進行訓練，充分利用了全域統計信息，高效地實現語義編碼。因此，本文中計算文本相似度時使用的模型為 GloVe。

2.1.2 動態詞向量模型

動態詞向量模型採用的是動態計算單詞向量的方式，對於一個單詞，動態詞向量模型會考慮它周圍的上下文單詞，並通過這些上下文單詞的向量來調整它的表示，相比靜態詞向量模型，動態詞向量模型更容易理解多義詞、語法變化和時態變化。目前表現較好的動態詞向量生成模型多為預訓練模型，如 ELMo [17]、BERT [4]，關於預訓練模型，內容詳見章節2.2.3。

2.2 語言模型

在自然語言處理中，語言模型主要用來評估一段文本在統計意義下的機率大小，通常定義為一個機率分佈函數，其輸入為一個文本序列，輸出為該文本序列

出現的機率。語言模型的任務是用一個機率分佈來描述一個句子序列出現的機率，具體來說，給定一個文本序列 w_1, w_2, \dots, w_n ，語言模型的目標就是計算該序列的出現機率 $P(w_1, w_2, \dots, w_n)$ ，即該序列作為一個整體在語言中出現的機率。

2.2.1 傳統語言模型

傳統的語言模型通過將該問題轉化為計算每個單詞在上下文中出現的機率的乘積，即：

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_{i-1}, w_{i-2}, \dots, w_1) \quad (2.8)$$

但是隨著文本長度的增加，乘積中的因數分子隨之增加，使用這樣的乘法機率鏈規則會使得計算的結果迅速趨近於零，加之該方法忽略了單詞出現的順序，難以處理長距離的依賴關係，導致模型的表現不佳。以上的問題可以總結為長文本序列計算效率低下及數據稀疏性的問題。

針對上述問題，n-gram [14] 模型引入了馬爾可夫假設（Markov assumption），通過提取上下文中 n 個連續的單詞作為一個整體來計算條件機率，從而降低了計算複雜度。n-gram 模型假設第 i 個單詞的出現只與前 $n-1$ 個單詞有關，對於一個 n 元組合 $(w_{i-n+1}, w_{i-n+2}, \dots, w_i)$ ，它出現的機率可以被估計為：

$$P(w_i | w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1}) = \frac{C(w_{i-n+1}, w_{i-n+2}, \dots, w_i)}{C(w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1})} \quad (2.9)$$

，其中 $C(w_{i-n+1}, w_{i-n+2}, \dots, w_i)$ 是 n-gram 的計數，表示 $(w_{i-n+1}, w_{i-n+2}, \dots, w_i)$ 在訓練資料中出現的次數， $C(w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1})$ 是 $(w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1})$ 在訓練資料中出現的次數。

但是，n-gram 模型並沒有解決所有的問題。例如，在處理長距離依賴的任務中，會存在諸如主謂賓關係、名詞修飾等複雜語法結構，n-gram 模型難以捕捉這

些結構。而且 n -gram 模型需要預先指定 n 的大小，隨著 n 的增大，其模型參數空間呈指數級增長，容易導致過擬合。

因此，面對 n -gram 模型存在的問題，自然語言處理領域開始探索新的語言模型，這也促進了神經網路語言模型的發展。

2.2.2 基於神經網路的語言模型

基於神經網路建立的語言模型是一種通過神經網路來預測單詞在語言序列中出現的機率的方法。相較于傳統的 N -gram 模型，基於神經網路的語言模型不需要預先指定 n 的大小，通過引入更多上下文信息並建立非線性的映射關係，可以更好地捕捉語言的語義信息，對長尾資料和多樣性語境也具有更好的適應性。此外，模型的參數量並不取決於 n ，而是由網路結構和訓練資料的大小所決定，因此，基於神經網路的語言模型可以保持較小的模型參數量，從而更好地克服模型複雜度問題，避免過擬合。

最早應用神經網路構建的語言模型的是 NNLM [1]，其使用的結構為前饋神經網路（feedforward neural network），也稱全連接神經網路（fully connected neural network），該模型輸入是一個固定長度的單詞序列的，如當前單詞前面的 k 個單詞，對該單詞序列進行 one-hot 編碼並進行向量化處理。之後，神經網路將這些向量輸入多個全連接層，並在每個層中進行非線性化處理，最終得到下一個單詞的預測機率分佈。

2.2.3 預訓練語言模型

預訓練語言模型是神經網路語言模型的一種，這類模型將大規模無標記的資料用於訓練，從大量語料中學習到語言的通用特徵和結構，並將這些通用表示用

於下游任務中。

繼全連接層被應用於語言模型之後，CNN [12] (Convolutional Neural Network)、RNN [21] (Recurrent Neural Network)、LSTM [10] (Long Short-Term Memory) 等網路結構也被相繼用於進行特徵提取 (feature extraction)。ELMo [17] (Embeddings from Language Models) 採用了一個兩層的雙向 LSTM 作為模型的骨幹，輸入是經過字符級卷積處理過的單詞表示。該模型訓練的目標是最大化給定語料庫的雙向語言模型的機率。在每一層，ELMo 使用了一個殘差結構，從而允許對深度網路中的每一層進行有效的信息傳遞。ELMo 模型最終會學習到一個權重矩陣，它將每個單詞的向量表示映射到一個高維空間中。在使用 ELMo 時，可以將這個權重矩陣與任何標準神經網路模型結合使用，以生成在不同自然語言處理任務上的特定表示。

使用 ELMo 模型可以克服傳統詞向量表示中的一詞一向量的缺陷，解決一詞多義的問題。然而，ELMo 模型中的雙向 LSTM 模型依然需要按照固定的順序逐個考慮輸入序列中的每個詞語，對於相對較長的序列，這種做法複雜度高、計算成本高昂，也容易出現梯度消失和梯度爆炸等問題。

2017 年，Transformer [22] 被提出。Transformer 是一種使用自注意力 (self-attention) 機制的深度神經網路模型，它的模型結構如圖2.1，該模型由編碼器 (encoder) 和解碼器 (decoder) 組成。輸入文本經過初始詞向量和位置編碼融合後，輸入到編碼器中進行特徵提取和 self-attention 計算。編碼器包括多個層，每個層都由兩個子層組成：多頭自注意力 (multi-head self-attention) 層和前饋網路 (feed forward) 層。多頭自注意力層和前饋網路層。在多頭自注意力層中，模型通過並行執行多個不同參數的自注意力來學習詞與詞之間的關係；前饋網路層則通過對當前上下文信息進行非線性變換生成更複雜的特徵以支持下一步任務；每個

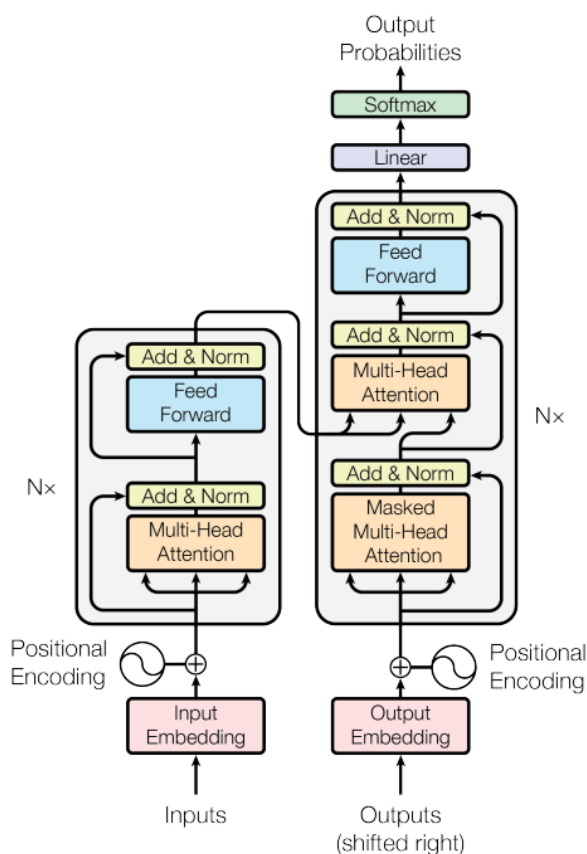


Figure 2.1: Transformer 模型架構

子層之間都有殘差連接和歸一化操作來增強網絡表達能力和穩定性，並且具有高度可訓練性。

經過若干個編碼器後，將提取到的特徵輸入到解碼器中。為了避免生成過程中訪問後面的信息，需要進行遮蔽操作，因此，解碼器使用 masked multi-head attention 來代替 multi-head attention，在計算當前解碼位置的注意力分佈時，只需要考慮該位置之前的所有位置，而忽略之後的所有位置，這是一種自監督學習方法，可以幫助模型學習如何利用上文資料來解決缺失的資料。在這種機制下，隨機選擇一些輸入序列中的單詞並把它們及其後的文本“遮蓋”掉，然後讓模型預測這些遮蓋掉的單詞。另外，相較編碼器，解碼器多了一個 Encoder-Decoder 的多頭注意力層，它將編碼器中所有位置向量和解碼器得到的目標詞匯向量作為輸入，利用自注意力機制獲得需要被關注的編碼器位置向量的權重，再利用這些權重將編碼器位置向量彙聚為一個向量，通過這種方式，自動地從輸入序列中提取

信息，並根據解碼器的當前狀態為其提供有針對性的上下文來生成目標序列。

Transformer 使用自注意力機制來直接捕捉序列中不同位置之間的長程依賴關係，從而可以對整個序列進行並行處理，大大減少了計算成本，同時還能更好地捕捉全域的語義關係，使得模型在自然語言處理任務中表現出了更強的性能。隨著 Transformer 模型在自然語言處理領域的成功應用，越來越多的研究人員和工業界開始使用 Transformer 模型進行預訓練，並在此基礎上設計出各種預訓練語言模型，相繼提出了 GPT [18]、BERT [4]、XLNet [24]、RoBERTa [13] 等模型。此後，預訓練語言模型的研究和應用迅速蔓延，開拓了自然語言處理領域的新局面。

2.2.4 基于 Transformer 的預訓練語言模型

2.2.4.1 GPT

GPT-1 [18] 開創了基於預訓練模型對下游任務進行微調的範式的先河，具體來說，該模型先使用大量無標注的 Web 文本和書籍文本作為訓練語料，使模型從無標注資料中學習文本序列和上下文之間的關係；再將預訓練好的模型在特定任務的有標注資料上進行微調，通過下游模型獲得針對該任務的結果。

GPT-1 模型由 12 個解碼層組成，這些解碼層在 Transformer 解碼層的基礎上，去除和編碼層交互的多頭注意力機制，第一個解碼層接受文本序列的詞嵌入和位置嵌入，通過多個解碼層得到要預測的下一個單詞的機率分佈，因此，這個語言模型是一個自回歸模型，它可以單向地對當前序列位置的下一個元素進行預測。

定義無標注語料 \mathcal{U} 包含文本序列 $\{u_1, \dots, u_n\}$ ，則 GPT-1 的目標函數是最大化下列函數：

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}) \quad (2.10)$$

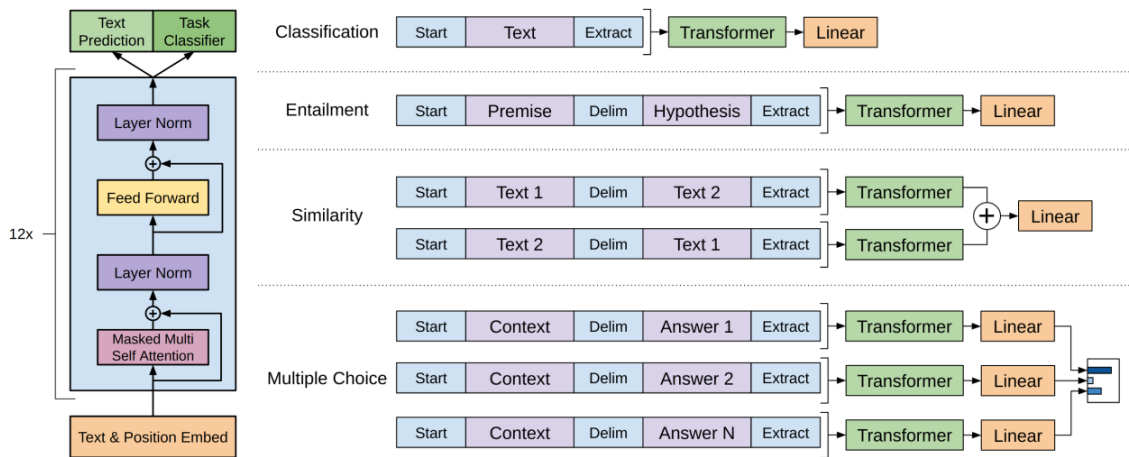


Figure 2.2: GPT-1 的模型結構及不同下游任務的拼接方式

在對不同下游任務進行微調時，為了方便模型處理文本序列中的不同部分，並且給予模型更加準確的定位信息，GPT-1 使用 $\langle s \rangle$ 、 $\langle e \rangle$ 和 $\langle \$ \rangle$ 分別作為文本開頭、文本結尾、兩文本之間拼接的標記符號，GPT-1 的模型結構及不同下游任務的拼接方式如圖2.2。給定一個有標注的資料集 \mathcal{C} ，該資料集的每個樣本中包含文本 $\{x^1, \dots, x^m\}$ 和標籤 y ，微調模型的目標函數是最大化下列函數：

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y | x^1, \dots, x^m) \quad (2.11)$$

若將上游模型作為輔助目標，有助於提高模型的泛化能力並加快模型的收斂速度，定義任務的權重係數為 λ ，最終的微調目標函數如下：

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda L_1(\mathcal{C}) \quad (2.12)$$

相比起 GPT-1，GPT-2 [19] 的主要結構沒有改變，但是它將層正則化從整個模型中移動到子模塊之前，並且在最後的自注意力模塊之後添加了一個新的層正則化，這種改進可以減少訓練過程中內部協方差的影響。GPT-2 在 GPT-1 的基礎上，通過增加解碼層數，使得模型更好地學習輸入序列中的複雜模式，同時，GPT-2 通過使用更大的語料，學習到更多的模型參數，提高了模型的預測能力，

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

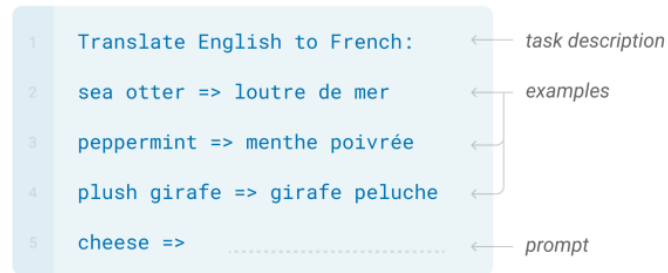


Figure 2.3: GPT-3 少樣本學習範例

在 GPT-2 的論文中也證實了更多模型參數能夠提到更好的性能這一點。

在下游任務方面，GPT-2 提出了多任務學習的概念，即在多個不同任務的資料集上面進行訓練至收斂，能夠大幅提升模型的泛化能力，從而實現在各個任務上都能取得良好表現。此外，零樣本 (zero-shot) 任務也是 GPT-2 的一個關鍵焦點，為了提升在零樣本任務上的性能，GPT 通過在需要預測的文字中加入純自然語言，對任務進行說明，從而引導模型完成零樣本任務，這種加入純自然語言引導模型預測的方式，後來被稱作為提示 (prompt)。

GPT-3 [3] 使用稀疏自注意力機制代替了 GPT-2 中使用的自注意力機制，使用了更大的語料和更多的層數來進行訓練。GPT-3 也訓練了不同參數量的模型，由於參數量大的模型很難微調，作者在文中提出了上下文學習 (in-context learning) 的概念2.3，在使用預訓練模型對下游任務進行預測時，使用幾個訓練樣本，整理成任務樣例，並加上任務描述，來引導模型完成相關的下游任務，這種使用極少量樣本進行學習的任務成為少樣本 (few-shot) 任務。

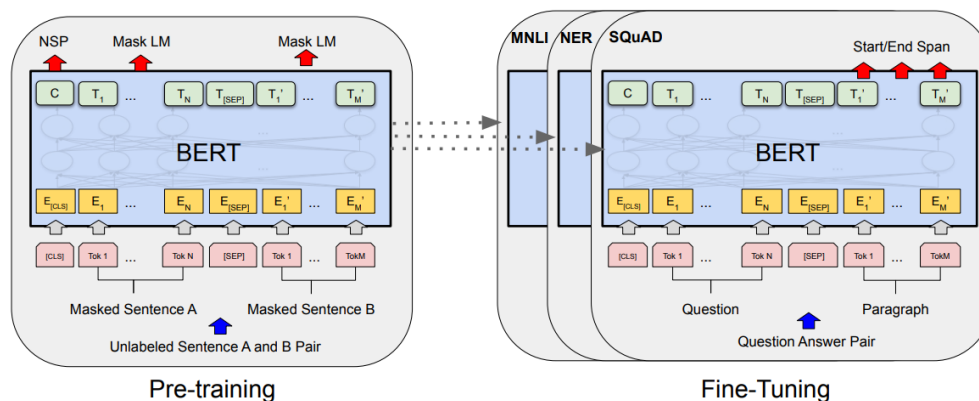


Figure 2.4: BERT 模型的預訓練及微調

2.2.4.2 BERT

BERT 的核心做法和 GPT 一致，都是通過無監督的大規模預訓練來學習通用的語言表示，然後在具體任務上進行微調^{2.4}。GPT 模型在預訓練階段只能單向地看到上文信息，而忽略了下文信息；而 ELMo 雖然採用了雙向 RNN，但兩個 RNN 之間是獨立的，沒有直接的信息交流。因此，BERT 採用了雙向上下文建模的方法，即在預訓練階段使用了雙向 Transformer 編碼器，使得每個單詞在預訓練過程中能夠同時利用其前後文的信息，從而更好地捕捉詞匯的含義和句子的語義。

BERT 的預訓練過程分為兩個階段：遮蔽語言模型（Masked Language Model，MLM）和下一句預測（Next Sentence Prediction，NSP）。

由於 BERT 的雙向 Transformer 結構，每個 Transformer 的輸出可以看到整個句子的信息，這導致在正常的語言模型預訓練中，模型在預測下一個單詞時可以“看見”參考答案，即存在數據洩漏問題。為了解決這個問題，在 MLM 階段，BERT 輸入的句子中的一部分單詞會被隨機遮蔽（例如用“[MASK]”標記替代），模型的目標是預測這些被遮蔽的單詞是什麼。通過這種方式，即使模型“看見”了被遮蔽單詞的位置，它無法直接從 Transformer 輸出中獲取到準確的答案，而是

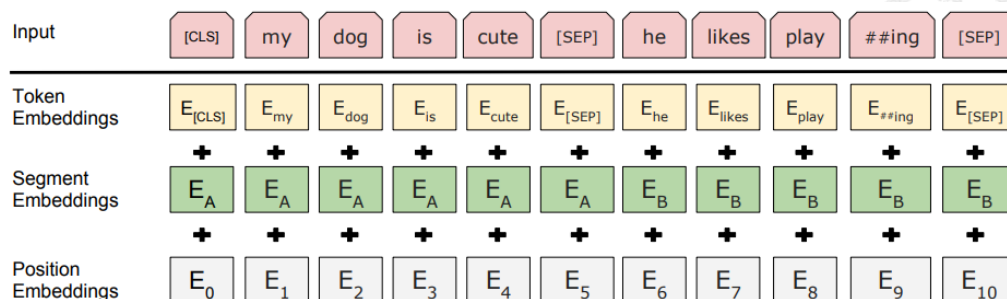


Figure 2.5: BERT 輸入的處理

需要根據上下文來進行推斷。

NSP 任務的目標是判斷兩個句子是否在語義上是連續的，即判斷第二個句子是否是給定第一個句子的下文。在預訓練階段，對於每個訓練樣本，BERT 會隨機選擇一對句子，並為這對句子生成輸入序列。輸入序列包括兩部分：第一個句子 A 和第二個句子 B。其中，句子 A 和句子 B 之間通過特殊的"[SEP]"標記進行分隔。在輸入序列中，BERT 還會添加一個特殊的"[CLS]"標記作為整個序列的起始標記，BERT 的輸入處理如圖2.8。為了進行 NSP 任務，BERT 還引入了額外的標記，即一個二分類的標籤：IsNext（下一句）和 NotNext（非下一句）。對於正樣本，即連續的兩個句子，標籤被設置為 IsNext；而對於負樣本，即非連續的兩個句子，標籤被設置為 NotNext。在預訓練過程中，BERT 模型將整個輸入序列輸入到 Transformer 網絡中，並得到最後一個隱藏層的輸出。然後，使用最後一個隱藏層的輸出對標籤進行預測，判斷兩個句子是否連續。

通過 NSP 任務，BERT 模型能夠學習到句子級別的語義關係，包括上下文信息和句子之間的連貫性。這使得 BERT 能夠更好地理解句子之間的語義關聯，並為多項自然語言處理任務提供更準確和全面的語義表示。

BERT 的微調階段和 GPT 階段差不多，但對於分類模型，由於"[CLS]"學習到了整個句子層面的知識，所以 BERT 會將"[CLS]"標記的表達傳入輸出層進行分類。

2.2.4.3 ALBERT

ALBERT [11] (A Lite BERT) 在 BERT 的基礎上通過參數共享、因式分解和引入 SOP 任務等改進，實現了模型參數量的減少、計算效率的提高和更好的句子連貫性預測，以下是 ALBERT 改進的具體內容：1. 通過因式分解和參數共享減少參數：ALBERT 通過將詞表的 embedding size (E) 和 transformer 層的隱層 hidden size (H) 分離，採用獨立於上下文的 embedding 進行計算，並投影到隱層空間上，實現了參數量的降低。這種因式分解的方法可以減少存儲空間和計算量，並且在實驗中表現良好。此外，ALBERT 採用了跨層參數共享 (cross-layer parameter sharing) 的策略，將所有層的 transformer 共享為一個模塊，使得每個層的參數實際上是同一個變量。這種跨層參數共享方式能夠顯著降低參數量，並提高模型的計算效率和訓練穩定性。2. 引入句子順序預測 (Sentence-Order Prediction, SOP)：ALBERT 針對 NSP 任務的限制性和主題偏差問題進行了改進。通過引入 SOP 任務，ALBERT 替代了傳統的 NSP 任務，要求模型判斷兩個句子的順序是否被交換過，從而更準確地評估句子之間的連貫性。這樣做可以避免 NSP 任務中的主題偏差，並提高模型對句子關係的理解和預測能力。ALBERT-xxlarge 在比 BERT-large 參數更小的情況下，效能更佳。

2.2.4.4 RoBERTa

RoBERTa [13] (A Robustly Optimized BERT Pretraining Approach) 在 BERT 的模型基礎上，進行了如下改進：

1. 動態遮蓋 (dynamic masking)：BERT 預處理中對每個樣本進行隨機的單詞遮蓋，後續訓練進行輸入時都使用相同的被遮蓋住的樣本，這樣的 mask 被稱作靜態遮蓋，因為對於整個訓練過程中，所使用的 mask 都是固定的。而

RoBERTa 在每次向模型提供輸入時會重新挑選 15% 的 token 進行隨機 mask，相比 BERT 的靜態遮蓋，這樣更加靈活和豐富，能夠更好地利用訓練數據進行訓練。

2. 去除下一句預測 (Next Sentence Prediction)：RoBERTa 去除了 NSP 任務，改為每次輸入連續的多個句子，通過全文訓練來學習句子之間的關係。相比於 BERT 的 NSP 任務，RoBERTa 更加直接且有效地學習了句子之間的關係。
3. 使用更多資料及更大 batch size 進行訓練：RoBERTa 在 BERT 的 16GB 訓練資料的基礎上，將訓練資料增加到 160GB，並使用更大的 batch size 以及更長的訓練時間，以提高模型性能和訓練效率。
4. 文本編碼方式：在 BERT 中，文本首先會被分成單個字符，然後通過 BPE 演算法將字符逐步合併形成詞匯表，例如將”un”和”tied”逐步合併成單詞”untied”。BERT 的詞表大小約為 30,000，這種字符級別的 BPE 編碼方式可以處理詞庫中不存在的單詞，但對於以字母形式拼寫不是最優的單詞或者多音字等存在問題的單詞則不是非常友好，也不容易解決 OOV (Out-of-Vocabulary) 問題。RoBERTa 採用的是字節級別的 BPE，首先將每個字符轉換為 ASCII 字節表示，再將字節編碼成 BPE。它的詞表大小為 50,000，比 BERT 更細粒度，這種方式對於處理單詞形式不規範的語言文本，以及處理 OOV 問題以及更為優秀。RoBERTa 使用了字節級別的 BPE 進行文本編碼，詞表大小為 5 萬，相比 BERT 的字符級別 BPE，這種方式更加細粒度 (fine-grained)，詞表更加豐富，能夠更好地解決 OOV 問題。



2.2.4.5 DeBERTa

DeBERTa [9] (Decoding-enhanced BERT with Disentangled Attention) 對 BERT 基礎上做了如下修改：

1. 分離的注意力機制 (disentangled attention) DeBERTa 在輸入層對 BERT 模型進行了改造，使用了分離的注意力機制，BERT 使用單詞嵌入和位置嵌入的和來表示單詞向量，而 DeBERTa 將單詞的位置編碼和內容編碼分開表示，使用兩個向量對單詞進行編碼，分別對應單詞的位置和內容。然後，通過解耦的矩陣來計算單詞之間的注意力得分。這樣做的好處是，在注意力分配時考慮了單詞的相對位置，更好地捕捉到了上下文語義信息。
2. 輸出層使用增強的掩碼解碼器 (enhanced mask decoder) 代替 Softmax：DeBERTa 用增強的掩碼解碼器代替 BERT 的 softmax 層來預測被遮住的 token。在微調時，一般會把 Bert 的輸出和特定任務的解碼器連接起來，但是在訓練時沒有接入解碼器，而是通過 softmax 將結果輸出。因此，在預訓練時，文中採用了一個兩層的 Transformer 解碼器和 softmax 作為增強的掩碼解碼器，並且在解碼器的 softmax 層之前，添加了單詞的絕對位置嵌入，以緩解預訓練和微調時的不匹配。

通過這兩個改變，DeBERTa 的訓練效率更快，相較 RoBERTa-large，僅使用一半訓練資料的 DeBERTa 在下游任務的表現較 RoBERTa 更好。

2.3 現有的零樣本文本分類模型

文本分類任務是預訓練模型下游任務的一種，其目標是將給定的文本實例分配到預定義的類別或標籤中。該任務根據文本內容的特徵和上下文，對文本進行

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



Figure 2.6: GPT-3 中給出的零樣本翻譯任務提示範例

分類，以便進行信息檢索、情感分析、垃圾郵件過濾、等應用。而零樣本文本分類（zero-shot text classification）是指在沒有任何訓練樣本的情況下，將給定的文本實例分配到預定義的類別中，由於沒有任何標記數據可以用於訓練模型，我們可以利用預訓練的語言模型和詞嵌入向量，通過將文本與類別之間的語義關係進行對齊，從而實現零樣本分類。使用預訓練模型實現零樣本文本分類的方式，根據模型的大小主要分為兩種：一種是基於大型模型，加入提示詞引導模型生成所需答案；另一種則是基於較小的模型進行微調。

2.3.1 大型生成式語言模型 + 基於提示的方法

基於提示的方法需要預先根據下游任務內容，為其設計一個描述該任務的提示，該提示以自然語言的方式添加在文本的輸入中，通過限制和指導模型的輸出，引導模型完成特定任務。GPT-3 論文中提供如何使用提示來引導零樣本翻譯的樣例，該樣例如圖。同樣地，如果需要將基於提示的方法用於零樣本文本分類，也可以為其設計一個任務描述的提示詞，如“下列文本屬於哪個類別”，語言模型會根據提示生成相應的內容。

InstructGPT [16] 和 chatGPT 所使用到的指令微調（instruction tuning）是基於提示的方法的一種，它們的目標都是通過挖掘語言模型本身的能力，生成所需要的答案。基於提示的方法通過給定格式的關鍵詞或短語來影響文本生成，更多的是利用語言模型的填空能力；指令微調給到模型的任務描述更加具體，此外，它

們利用人類輸入的指令和反饋來改進模型，通過使用指令式交互來控制生成的輸出內容，更多挖掘到的是語言模型的理解能力。

使用生成式語言模型加入提示的方式，可以充分利用到模型的上下文的 attention 機制，讓模型根據相關的提示快速地生成所需的答案，但是由於 prompt 的設計不存在通用的模板，因此對人類的專業水平有要求；另外若上游模型為生成式模型，則容易產生與答案無關的內容，從而產生誤差和噪聲；此外，由於目前的大型語言模型通常需要通過調用接口才能進行使用，其速度依賴於模型接口的調用速度，實際商用中可用性較差。

2.3.2 較小預訓練語言模型 + 微調的方法

通過使用公開的資料、目標資料集類別，或是目標資料集未標注資料來構建訓練資料，對較小的模型進行微調，也是解決零樣本文本分類的一種方法。但由於較小的易於微調的模型參數量較少，其泛化能力往往較弱，準確率低。

2.3.2.1 基於自然語言推理的方法

TE-Wiki [5] (Textual Entailment formulation with Wikipedia finetuning) 這個模型利用開源的維基百科文本構建訓練資料，將維基百科的文本作為前提 (premise)，維基百科的類別作為假設 (hypotheses)，按照 "[Text] Entails [Label_{*i*}]" for $i \in [n]$ 的格式整理，以進行對某文本是否蘊含某類別的二分類。

在具體實施中，研究人員從維基百科的目錄中選擇了 674 個頂級目錄作為分類樹的根節點。通過深度優先搜索方法，以深度為 2 向下遍歷分類樹，獲取其子類別。每篇文章的前 128 個標記 (token) 被用作訓練文本，文章所屬的類別被作為正例標籤 (positive label)，從其他類別中隨機選擇一個作為負例標籤 (negative

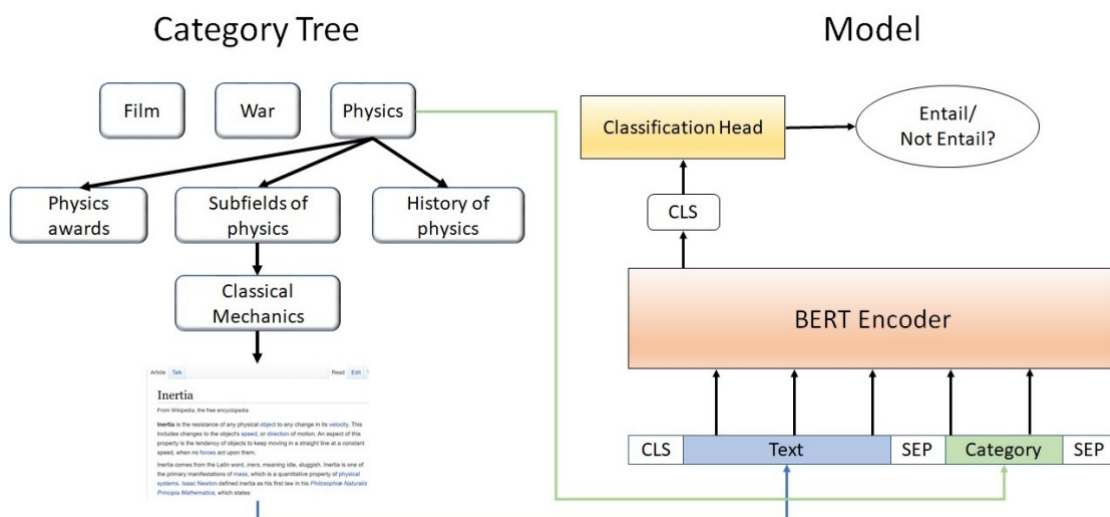


Figure 2.7: TE-Wiki 模型結構

label)，每個文本構成兩個文本-類別對，用於對 BERT 模型進行微調的訓練資料。該模型在零樣本文本資料集上進行推斷，並在 AG News 資料集的測試集上獲得了 0.796 的準確率。

Zero-Shot Text Classification with Self-Training [7] 一文中，也使用了這樣基於自然語言推理的方法（Natural Language Inference, NLI），不同于 TE-Wiki，文章中使用的訓練資料是要預測的無標籤資料。給定一個模型 M 、未標記樣本集 U ，其中 u 是 U 中的一個樣本，以及目標類別集合 C ，其中 c 是 C 中的一個類別，其訓練方法如下：

1. 使用原始模型 M 對未標記樣本集 U 進行預測，將預測結果作為訓練資料來訓練一個新模型，得到模型 M' 。
2. 使用樣本集 U 來訓練模型 M' ，得到模型 M'' 。
3. 將正例樣本和負例樣本添加到訓練資料中，重複上述步驟一次。

選擇正例樣本的方法是在訓練過程中從預測結果中篩選出對當前類別最有信心的樣本。具體步驟如下：

對於每個未標記樣本 u ，模型 M 會給出它屬 \square 每個類別 c 的置信度得分 S_{uc} 。

首先，選擇具有最高得分的類別作為最佳類別，即 $S_{uc} > S_{uc'}, \forall c' \neq c$ ，其中 c' 為其它類別。

再使用”最佳與次佳比較”的方法，假設次佳類別為 c' ，計算得分差值 $\delta_{uc} = S_{uc} - S_{uc'}$ 。這個差值反映了模型對最佳類別的置信度相對於次佳類別的相對置信度。

最後對所有樣本的 δ_{uc} 進行排序，選擇具有最高前 100 個差值的樣本作為正例樣本。選擇負例樣本的方法有四種：

1. Contrast-random：隨機選擇一個除最佳結果外的樣本。
2. Contrast-closest：選擇次佳結果的樣本。
3. Contrast-furthest：選擇具有最低 S_{uc} 值的樣本。
4. Contrast-all：選擇除了最佳樣本所有被預測為類別 c 的樣本。

實驗證明隨機選擇或是全選的方法能取得較好的結果，由於隨機選擇更節約運算資源，因而採用了隨機負例樣本的方式。

該模型提出的自訓練方法，通過預見標籤，使用標籤和已有的未標記資料來微調預訓練模型，最好的效果是對 DeBERTa 進行微調，在 AG News 的測試集上獲得了 0.814 的準確率。

基於自然語言推理的方法結構相對簡單，易於處理；然而，這種方法僅採用二分類的方式對文本類別進行分類，每一次輸入模型都僅考慮當前輸入的標籤，無法捕捉更多類別之間的關係。此外，隨著文本類別的增加，計算時間複雜度也

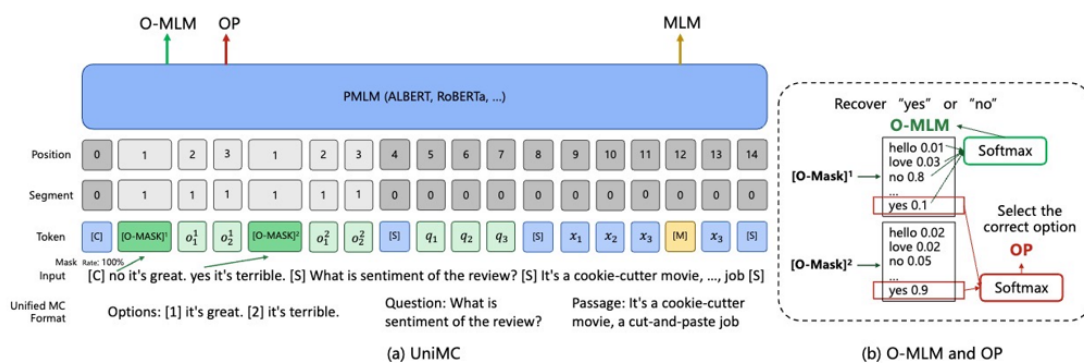


Figure 2.8: UniMC 模型結構


會增加，限制了該方法的可擴展性。因此，需要探索更加高效且能夠捕捉多類別關係的方法，以提高文本分類的準確性和可擴展性。

2.3.2.2 UniMC

UniMC [23] 的結構如圖所示，它採自編碼結構，把基於標籤 (label) 的自然語言理解 NLU 任務都轉換成統一的多項選擇 (Unified Multiple-Choice) 的格式，把標籤作為一個選項 (option)，在選項前添加一個用於預測選項機率的 token [O-MASK]，該 token 通過複用 [MASK] token 而來，目的是用來表示是否要選擇該選項，並在其後添加了可選擇的 question prompt 以用於描述任務。

該模型使用 14 個 NLU 任務資料集 2.9 作為訓練資料，並在預訓練模型 ALBERT 上進行微調，在微調過程中，模型取每個 [O-MASK] 輸出的 'yes' 的 logit 進行 softmax，得到每個選項的機率，並以機率最大的選項作為預測答案和標準答案求交叉熵損失，並在未見任務上進行零樣本預測，最終在 AG News 資料集上獲得 0.813 的準確率。

UniMC 同時考慮到了文本的內容和文本的類別標籤，可以獲得更加準確的文本信息，能夠更好地表達文本與類別、類別與類別之間的相關性；此外，將輸入整理成多項選擇的方式，並在文本處理過程中加入了提示詞 (prompt) 的概念，



Datasets	# of option	# of examples
ARC	4	3.37k
CommonsenseQA	5	9.7k
Cos-E	5	10.9k
CosmosQA	4	25.2k
Dream	4	10k
Mctest	4	2.4k
MultiRC	multiple	12k
OpenbookQA	4	9.9k
PIQA	2	16.1k
QASC	8	8.1k
Race	4	87.8k
Socail IQa	3	33.4k
WikiHop	multiple	43.7k
WIQA	3	36.7k

Figure 2.9: UniMC 模型微調所用的 14 個資料集

加強了模型對分類任務的感知，能夠提高模型對分類任務的準確性。





第三章 資料集介紹

為驗證零樣本文本分類模型的效能，本研究在四個分類任務的資料集中選取測試資料進行零樣本模型的評估，其中包含三個主題分類的資料集和一份情緒分類資料集，整體資訊如表3.1。下面將對這些資料集進行詳細介紹。

3.1 Yahoo! Answers

該資料集是由 Xiang Zhang[25] 等人根據 Yahoo! Answers Comprehensive Questions and Answers (YACQA) version 1.0 的資料集構建的，YACQA 由 Yahoo 公司根據真實用戶在 Yahoo 上的提問整理而成，這些問題整理了截至至 2007 年 10 月 25 日前的問答資料庫，語料由中包含 4,483,032 個問題和這些問題的答案。Xiang Zhang 等人從中選取了 10 個問題數量最大的主題，根據這些主題及這些主題下的問答構建了一個主題分類資料集。其中每個類包含 140,000 個訓練樣本及 6,000 個測試樣本，每個樣本所包含的字段包括問題的標題、問題的內容和最佳答案。

該資料集的文本類別如表3.2，樣本字段樣例如表3.3。

資料集名稱	資料集屬性	類別數量	測試資料數量	資料分佈情況
Yahoo! Answers	問答主題分類	10	60,000	均勻分佈
AG News	新聞主題分類	4	7,600	均勻分佈
DBPedia	維基百科主題分類	14	70,000	均勻分佈
IMDB	電影評論情緒分類	2	25,000	均勻分佈

Table 3.1: 用於評估的資料集概況



類別標籤	類別名稱
0	Society & Culture
1	Science & Mathematics
2	Health
3	Education & Reference
4	Computers & Internet
5	Sports
6	Business & Finance
7	Entertainment & Music
8	Family & Relationships
9	Politics & Government

Table 3.2: Yahoo! Answers 資料集類別

id	topic(class label)	question_title	question_content	best_answer
0	8 (Family & Relationships)	" What makes friendship click?"	" How does the spark keep going?"	"good communication is what does it. Can you move beyond small talk and say what's really on your mind. If you start doing this, my experience is that potentially good friends will respond or shun you. Then you know who the really good friends are."

Table 3.3: Yahoo! Answers 樣本字段樣例

類別標籤	類別名稱
0	World
1	Sports
2	Business
3	Sci/Tech

Table 3.4: AG News 資料集類別



text	label (class label)
"Fears for T N pension after talks Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul."	2 (Business)

Table 3.5: AG News 樣本字段樣例

3.2 AG News

AG 是由 ComeToMyHead 學術新聞搜索引擎從 2000 多個新聞來源整理出來的超過 100 萬篇文章的集合，Xiang Zhang[25] 等人根據 AG 構建了用於文本分類的資料集 AG News，這個資料及從原始語料庫中選擇了四個包含內容最多的類別，其中每個類包含 30,000 個訓練樣本及 1,900 個測試樣本，每個樣本所包含的字段包括問題的標題和問題的描述。

該資料集的文本類別如表3.4，樣本字段樣例如表3.5。

3.3 DBpedia

DBpedia 是一個從維基百科中提取結構化資料的項目，本文中所使用的 DBpedia 資料集是由 Xiang Zhang[25] 等人根據 DBpedia 2014 資料集構建而來，該資料集包含了從 DBpedia 2014 中隨機挑取的 14 個非重複性類別，每個類別包含隨機選擇的 40,000 個訓練樣本和 5,000 個測試樣本，每個樣本中的字段包含維基百科文章的標題和摘要。



類別標籤	類別名稱
0	Company
1	EducationalInstitution
2	Artist
3	Athlete
4	OfficeHolder
5	MeanOfTransportation
6	Building
7	NaturalPlace
8	Village
9	Animal
10	Plant
11	Album
12	Film
13	WrittenWork

Table 3.6: Dpedia 資料集類別

label(class label)	title (string)	content (string)
0 (Company)	" CNet Technology"	" CNet Technology is a Taiwanese company that manufactures network equipment such as network cards switches and modems."

Table 3.7: DBpedia 樣本字段樣例

該資料集的文本類別如表3.6，該資料集的樣本字段樣例如表3.7。

3.4 IMDB

資料集由史丹佛大學團隊根據互聯網電影資料庫 IMDB 的 50000 條嚴重兩極分化的評論整理而來。這個資料集的訓練集和測試集分別有 25,000 條，訓練集和測試集都包含 50% 的正面評論和 50% 的負面評論，其中正面評論類別標籤為 1，類別名稱為 "pos"；負面評論類標籤為 0，類別名稱為 "neg"。

該資料集的文本類別如表3.8，該資料集的樣本字段樣例如表3.9。

類別標籤	類別名稱
0	neg
1	pos

Table 3.8: IMDB 資料集類別



text	label (class label)
"Technically I'am a Van Damme Fan, or I was. this movie is so bad that I hated myself for wasting those 90 minutes. Do not let the name Isaac Florentine (Undisputed II) fool you, I had big hopes for this one, depending on what I saw in (Undisputed II), man.. was I wrong ??! all action fans wanted a big comeback for the classic action hero, but i guess we wont be able to see that soon, as our hero keep coming with those (going -to-a-border - far-away-town-and -kill -the-bad-guys- than-comeback-home) movies I mean for God's sake, we are in 2008, and they insist on doing those disappointing movies on every level. Why ??!!! Do your self a favor, skip it.. seriously."	0 (neg)

Table 3.9: IMDB 樣本字段樣例





第四章 研究方法

本文中的研究方法基於預訓練模型和開放域資料集，通過微調和類別映射的方式提高文本分類任務的性能。具體來說，研究方法主要包括兩個重點：

1. 模型微調：使用開放域的維基百科文本資料，將其構建成分類任務的格式，用於微調預訓練模型，提高模型的分類能力和性能。
2. 文本映射：為了讓任務的標籤更接近模型微調過程中訓練過的標籤，在推理過程中使用一種基於 GloVe 模型的類別映射方式，將需要預測的目標類別用預訓練過程中的維基百科類別代替，從而提高模型預測準確率。

下面將對這兩個方法進行進一步介紹。

4.1 模型微調

本文中的實驗主要使用統一的多選項模型架構 (UniMC)、基於自然語言推理 (TE-Wiki) 及基於自訓練 (self-training) 的方法對預訓練模型 RoBERTa、BERT、ALBERT、DeBERTa 進行微調，這些方法和模型結構在章節2中有詳細介紹，因此本章節不再贅述。

本章節的將重點介紹訓練資料的獲取及各模型的輸入。



4.1.1 開放域訓練資料的獲取

開放域資料 (open domain data) 指的是與特定任務或主題無關的通用性資料，它們通常包含了大量的主題領域和語言表達形式，並具有高度的多樣性和複雜性。維基百科是一項由全球志願者共同維護的大型免費網絡百科全書，涵蓋了許多主題領域，包括科學、文化、社會、歷史、地理、藝術、體育等方面的條目，因此，維基百科是一個非常有價值的開放域資料源。

本研究中訓練資料的獲取參考了 TE-Wiki 論文，將維基百科網頁上的文本和其所屬類別，整理成訓練資料。與 TE-Wiki 不同的是，本研究將使用多個類別對預訓練模型進行微調，構建包含 n 個類別的訓練資料，而不是只有正負樣本兩個類別。如圖4.1所示，本研究中選取了維基百科概述頁面的 700 個頂級類別作為根節點，在刪除了以 “List of” 開頭的 26 個類目後，對每一個根節點，使用深度優先搜索 (DFS) 演算法，設置搜索深度為 2，來查找其子類別，並將這些類別和子類別構建成類別樹。上述做法中，設置深度為 2 的目的是確保所找到的子類別與根節點具有強關聯性。本研究中將該類別樹中所有直屬於根結點的文章蒐集起來，將每一篇文章對應的根結點作為其正例樣本，並從類別樹上其餘根結點中隨機選取 $n - 1$ 個非所屬類節點的類別，以作為文章的 $n - 1$ 個負例樣本，如圖4.2所示，若 $n = 3$ ，Text 1 指數的類別為 c_1 ，則將 c_1 設置為該筆資料的正例標籤；而負例標籤則是從 c_2 到 c_{674} 中隨機選取的兩個類別。選擇直屬於根結點的文章的目的是為了獲得只屬於一個類別的文本，構建答案唯一的單類別訓練資料。樣本構建的過程如算法1。

通過上述方式，我們獲得 974,923 篇文章及其所屬的 674 個類別。

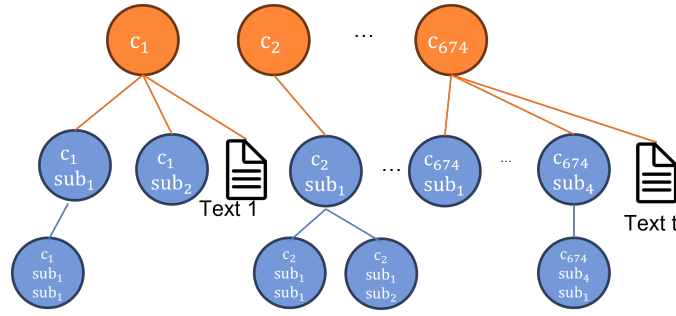


Figure 4.1: 類別樹構建示意圖

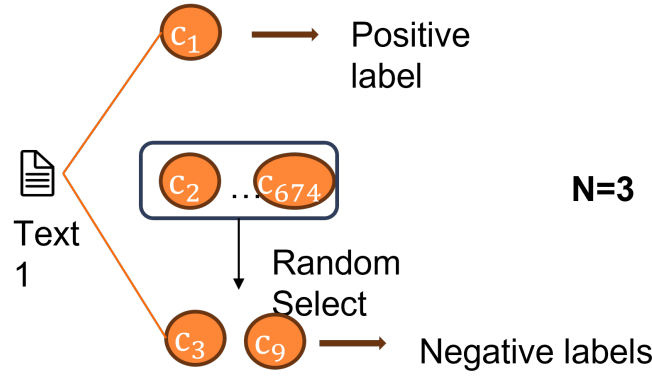


Figure 4.2: 正負例樣本選擇示意圖

Algorithm 1: Training Data Collection

input : Top-level category set S , Wikipedia subcategory graph \mathcal{G} , Wikipedia articles X , max search depth $r = 2$;

output: M

```

1 Initialize  $d(x, c) = \infty$  for any article  $x \in X$  and  $c \in S \cdot M = \{\}$ ;
2 for  $c$  in  $S$  do
3    $T = \text{DFS}(c, \mathcal{G}, r)$ ;
4   for  $t$  in  $T.\text{nodes}$  do
5     for  $x$  in  $t.\text{articles}$  do
6        $d(x, c) = \min\{d(x, c), 1 + \text{depth}(t)\}$ 
7 for  $x$  in  $X$  do
8   if  $\min_{c \in S} d(x, c) < \infty$  then
9      $P = \text{argmin}_{c \in S} d(x, c)$ ;
10    for  $c$  in  $P$  do
11      if  $c.\text{len}() == 1$  then
12        Add  $(x, c, 1)$  to  $M$ ;
13        for  $i$  in  $[1, n - 1]$  do
14          Sample  $c'$  from
15           $c' \in S - P, c' \neq P \cup \{c'1, c'2, \dots, c'n - 2\}$ ;
          Add  $(x, c', 0)$  to  $M$ ;

```



4.1.2 模型輸入格式處理

獲得文章的正負例樣本之後，本研究將這些樣例根據不同模型的輸入需求，整理為不同的輸入格式，由於預訓練模型可接收的文本長度有限，本研究截取每篇文章的前 266 個 tokens 作為當前樣例的文本，為了更好地進行模型訓練和性能評估，該研究在整理樣本時充分考慮到了每篇文章可能屬於不同的類別，通過整合正負例樣本，每篇文章都被標記為一個具有確定類別的元組。

具體而言，對於每篇文章，可以得到一個元組 $(text, class_list = [class_1, class_2, \dots, class_n], ground_truth)$ 。其中 $text$ 是截取后的文章片段。 $[class_1, class_2, \dots, class_n]$ 表示該 $text$ 對應的正負樣本類別的集合，這裏放的是將所有類別打亂之後的結果。 $ground_truth$ 指的是正樣本對應的類別，即文章的正確分類結果。

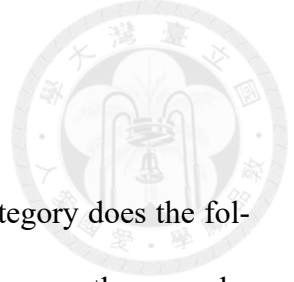
對於需要用到提示詞的模型，除了 Self-training 我們保留了其論文中使用到的提示詞 “This example is”，其餘模型本文統一設置提示詞為 “Which category does the following text belong to?”。

對於使用 prompt 的生成式模型，本文將訓練資料整理成以下輸入格式：

$$"[class_i], " \text{ for } class_i \in class_list [prompt] [text] \quad (4.1)$$

舉例來說，若有一個元組，所包含的元素內容分別如下：

- $text$: "Keep an eye on your credit card issuers – they may be about to raise your rates."
- $class_list$: ["Business", "Sci/Tech", "Sports", "World"]
- $ground_truth$: "Business"



則在前置處理中需把輸入格式整理成：

- [”Business”], [”Sci/Tech”], [”Sports”], [”World”], [”Which category does the following text belong to? ”][”Keep an eye on your credit card issuers – they may be about to raise your rates.”]

對於 TE-Wiki 模型，每個樣例將被整理成：

$$[CLS][text][SEP][class_i][SEP] \text{ ” for } class_i \in class_list \quad (4.2)$$

對於上文所舉例子，使用 TE-Wiki 需整理出四筆輸入：

- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP] [”Business”][SEP]
- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP] [”Sci/Tech”][SEP]
- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP] [”Sports”][SEP]
- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP] [”World”][SEP]

對於使用 Self-training 方法的模型，每個樣例將被整理成：

$$[CLS][text][SEP][prompt + class_i][SEP] \text{ ” for } class_i \in class_list \quad (4.3)$$

對於上文所舉例子，使用 Self-training 較 TE-Wiki 多了提示詞，其整理後結果如下：



- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP][”This example is Business”] [SEP]
- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP][”This example is Sci/Tech”] [SEP]
- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP][”This example is Sports”] [SEP]
- [CLS] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP][”This example is World”] [SEP]

對於使用統一多選項格式的模型，每個樣例都將被整理成：

$$[\text{CLS}] \text{ “}([O - MASK_i] [\text{class}_i] \text{ for } i \in n) \text{” } [\text{SEP}] [\text{prompt}] [\text{SEP}] [\text{Text}] [\text{SEP}] \quad (4.4)$$

對於上文所舉例子，UniMC 所接收的輸入格式為：

- [CLS] [$O - MASK_0$] [”Business”] [$O - MASK_1$] [”Sci/Tech”] [$O - MASK_2$] [”Sports”] [$O - MASK_3$] [”World”][SEP] [”Which category does the following text belong to? ”] [SEP] [”Keep an eye on your credit card issuers – they may be about to raise your rates.”][SEP]



4.2 類別映射

在進行模型推理（inference）之前，我們為每個目標類別選擇與其最相似的維基百科類別，用於構建替代詞列表，在模型推理過程中使用替代詞列表中的詞語代替目標類別進行預測。使用類別映射可以將預訓練模型在預訓練過程中學習到的知識、經驗和模式，運用在新的任務上，有助於預訓練模型的知識遷移。具體來說，這個過程包括以下步驟：

1. 使用 GloVe 模型，計算目標類別和維基百科類別的詞向量。
2. 根據維基百科類別詞向量和目標類別詞向量之間的余弦相似度將維基百科類別映射為目標類別的替代詞列表。
3. 在模型推理過程中使用替代詞代替目標類別進行預測。
4. 在後置處理中根據類別映射詞典將模型輸出對應至目標類別。

4.2.1 類別映射前置處理

使用 GloVe 模型計算目標類別與維基百科類別詞向量之前，需要對待輸入模型的類別進行字串的預處理，假設待輸入的類別字符串為 s ，其包含單詞 w_1, w_2, \dots, w_n ，預處理具體方式如下：

1. 去除特殊標點符號，得到字符串 s' 。
2. 對於字符串 s' 中的每個單詞 w_i ，計算其詞向量 v_i 。

3. 求字符串 s' 中所有單詞的詞向量的平均值，得到字符串向量 v_s 。

$$v_s = \frac{1}{n} \sum_{i=1}^n v_i \quad (4.5)$$



4.2.2 類別映射

假設維基百科的類別集合為: $L_w = \{w_1, w_2, \dots, w_{n_w}\}$ ，假設目標類別集合為: $L_z = \{z_1, z_2, \dots, z_{n_z}\}$ ，使用映射函數 f 表示將維基百科類別加入到某目標類別的替代詞列表候選項映射函數。

對於每一個維基百科類別，我們根據其詞向量，求出它和每一個目標類別詞向量的餘弦相似度，得到維基百科類別和目標類別的相似度矩陣 $S \in \mathbb{R}^{W \times Z}$ ， $S =$

$$\begin{bmatrix} s(w_1, z_1) & s(w_1, z_2) & \cdots & s(w_1, z_{n_z}) \\ s(w_2, z_1) & s(w_2, z_2) & \cdots & s(w_2, z_{n_z}) \\ \vdots & \vdots & \ddots & \vdots \\ s(w_i, z_1) & s(w_i, z_2) & \cdots & s(w_i, z_j) \end{bmatrix}$$

其中 $S_{i,j}$ 表示第 i 個維基百科類別和第 j 個目標類別之間的余弦相似度。對於每一個維基百科類別，我們選擇與之最相似的目標類別 k ，將該維基百科類別加入到與其最相似的目標類別的替代詞列表 M_k 中，映射函數可以表示為：

$$f(i) = k = \arg \max_j S_{i,j} \quad (4.6)$$

在詞向量空間上將維基百科類別按目標類別歸類示例如圖4.3，在該示例圖中， w_1, w_2, \dots, w_8 為待分類的維基百科類別，對於 w_1, w_2, w_3 ，和它們最相似的目標類別均為 c_1 ，則將它們加入到 c_1 的替代詞列表中。

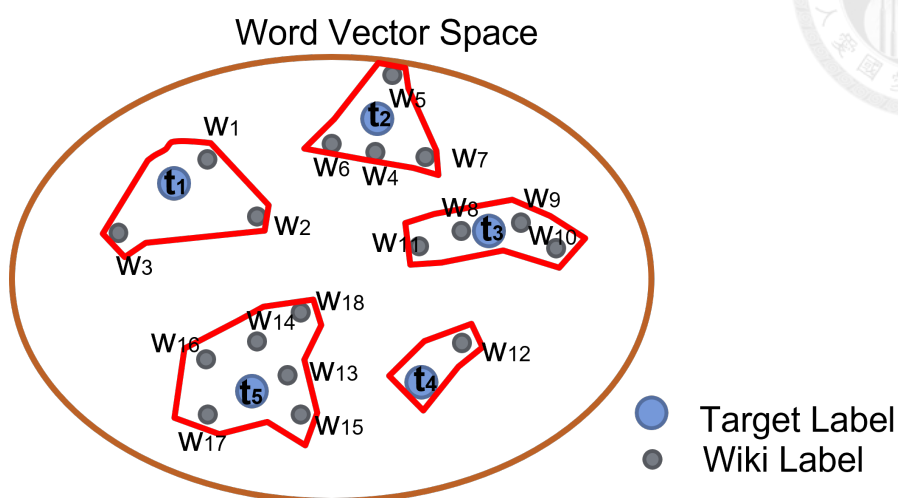


Figure 4.3: 在詞向量空間上將維基百科類別按目標類別歸類示意圖

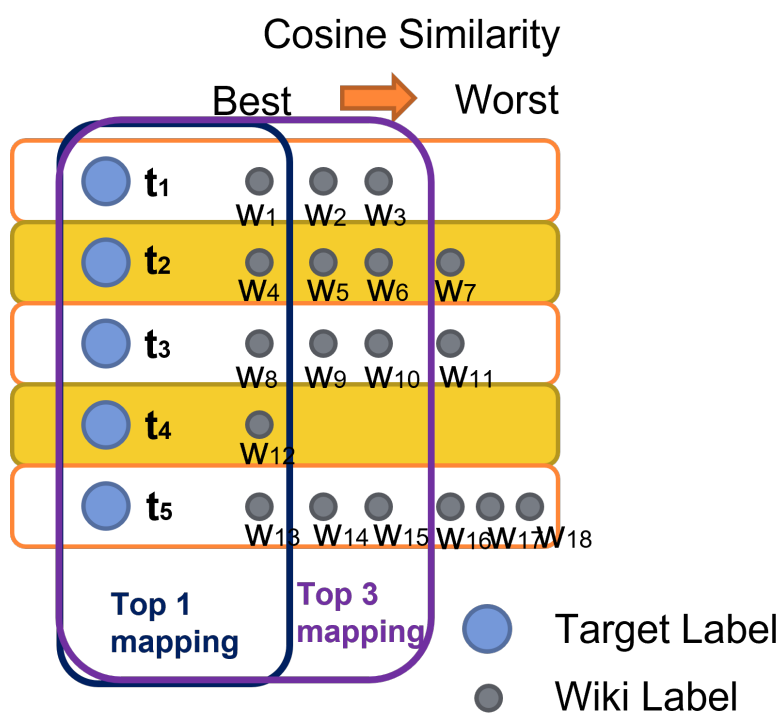


Figure 4.4: 將維基百科類別映射為目標類別候選項示意圖

如圖4.4所示，將每一個維基百科類別都映射為目標類別的替代詞列表的候選項後，本文將每個目標類別的替代詞列表中的候選項按照其與目標類別的相似度進行排序，再根據實際需求，選擇前 k 個候選詞，即與目標類別最像的前 k 個維基百科類別，作為最終替代詞列表中的選項。舉例來說，假設有一個目標資料集的其中一個類別為“哺乳動物”，我們通過上述方式得到它的維基百科類別替代詞候選項為：“恆溫脊椎動物”，“胎生動物”，“貓”，“狗”，“豬”，“兔子”其相似度為 [0.8611, 0.8610, 0.7952, 0.7451, 0.7193, 0.6060]，需要取其前 5 個最佳替代詞，則其最終替代詞列表需要刪除候選項中的最後一個類別，得到的最終替代詞列表為“恆溫脊椎動物”，“胎生動物”，“貓”，“狗”，“豬”。

本文使用維基百科類別找到最相似的零樣本文本類別，並加入其代替標籤候選列表，而不是直接通過計算零樣本文本類別與維基百科文本類別，選擇其最相似的前 k 個詞，避免了我們在選擇維基百科類別時面臨的不同零樣本文本類別高相似度指向同一維基百科類別的問題。

4.2.3 替代詞列表使用

得到最終的替代詞列表後，我們在進行推理預測時，使用這些維基百科類別代替目標類別作為輸入，讓微調過的模型對這些類別進行分類。如圖4.5所示，假設需要預測的 *text* 為“這個毛茸茸的動物愛抓老鼠”，*class_list* 為“哺乳動物”，“鳥類”，*ground_truth* 為“哺乳動物”，在推理過程中，我們使用替代詞列表中的“恆溫脊椎動物”，“胎生動物”，“貓”，“狗”，“豬”，“兔子”代替“哺乳動物”作為選項，輸入到模型，讓模型根據替代詞進行推理。

最後，在模型完成推理後，使用替代詞字典，以模型的輸出為 *value*，尋找其對應的 *key*，將 *key* 作為結果輸出。如圖4.5，在使用替代詞列表之後，模型根據

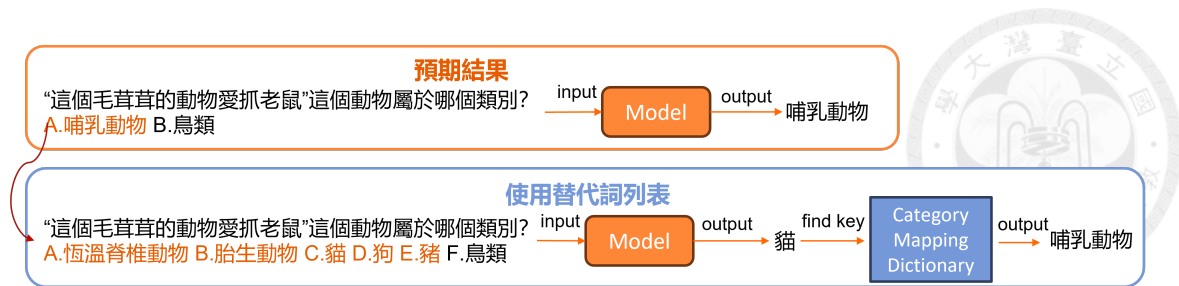


Figure 4.5: 替代詞使用示意圖

句意判斷出該動物為”貓”，我們以”貓”為 value，在替代詞列表字典中找到其 key 為”哺乳動物”，則將”哺乳動物”作為最終答案輸出。

4.2.4 替代詞篩選機制

為避免替代詞質量過差，消除替代詞中出現的非相似詞對模型判斷帶來的干擾，本文在實驗過程中還對替代詞補充加入了篩選機制，實現方法分為兩點：1. 設置閾值 2. 目標類別單詞確認。

首先，為了排除掉和目標類別相似度低的維基百科類別，本文將其相似度的閾值設置為 0.8，如果替代詞候選詞中維基百科類別與目標類別的餘弦相似度小於 0.8，則將該候選項刪除，如上述例子中，替代詞候選項”貓”、”狗”、”豬”、”兔子”與目標類別”哺乳動物”的相似度均小於 0.8，則將其從候選項中刪除。

其次，為了避免在維基百科類別中找不到相似度高的替代詞，本文加入了目標類別單詞確認的機制，該確認機制通過判斷目標類別的單詞的小寫是否包含在某個替代詞的單詞轉小寫後的子字串中，從而判斷是否需要加入該單詞，若有，則無需加入；若無，則加入該單詞。舉例來說，假設目前的目標類別為”Society & Culture”，通過閾值篩選後獲得替代詞列表 [”Science and culture”，”Popular culture”，”Mythology by culture”]，由於”Society & Culture”中，單詞”Culture”的小寫”culture”存在在”Science and culture”、”Popular culture”和”Mythology by culture”最後一個單詞”culture”的字子串裏，所以無需加入

“culture”。而“Society”不存在在三個候選替代詞的單詞的字子串中，故需要在替代詞中加入“Society”，最終的替代詞列表為：[“ Society” ,” Science and culture” ,” Popular culture” ,” Mythology by culture”]。

預訓練模型可以通過大規模開源領域的資料訓練，學習到關於文本分類的一些模式和特徵，這些知識在新的領域的文本分類任務中同樣適用。文本映射的方法可以通過對預測標籤的處理，使這些標籤與預訓練模型的標籤更加相似，從而增加模型對新任務中的標籤的理解。這樣，模型就可以更好地處理新的任務，並能更好地使用預訓練的知識和經驗。



第五章 實驗設計和結果討論

本章節將會對實驗任務設計、實驗流程與環境設定、實驗方法及結果進行詳細敘述。


5.1 實驗任務

零樣本文本分類任務的目的是在沒有訓練資料的情況下，將新出現的文本分類到已知的類別中，由於缺乏已知類別的訓練資料，無法應用傳統的監督學習方法訓練，新的文本可能會出現在各種不同的類別中，因此該任務變得更為具有挑戰性，這將要求模型對大量類別擁有豐富的知識背景。

在本論文的研究中，主要針對以下三個任務：

1. 提高模型對當前任務的感知能力。
2. 解決零樣本文本分類中缺乏標注資料導致模型無法進行學習的問題, 降低領域依賴性並提高模型泛化能力。
3. 加強知識遷移，提高分類精度。

提出了以下三個方法：

- 
1. 使用 UniMC 的模型結構，可添加 prompt 以進行任務描述，同時，將所有類別整合到一起，可以讓模型學習到更多類別與類別、類別與文本之間的關係，從而學習到如何在多個類別中選擇最佳答案。
 2. 利用公開領域的知識庫構建分類任務訓練資料，通過模型學習到的多領域先驗知識 (Prior Knowledge) 緩解領域依賴性，訓練出有強泛化能力的分類模型。
 3. 將維基百科類別映射至零樣本文本類別的替代詞列表中，將替代詞列表中的詞匯代替零樣本文本類別，讓模型進行預測，使用已訓練過的類別，促進模型所學知識在新任務上的遷移，從而提高分類精度。

本研究中的實驗將針對以上三個方法的效果進行驗證。

5.2 實驗流程及設定

本小節將介紹實驗的流程及實驗的設定。

5.2.1 實驗流程

本研究中對所有資料集採用的評估指標都是準確率 (Accuracy)，其計算方式為：

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (5.1)$$

其中，TP 代表真陽性 (True Positive) 指模型正確地將預測為正類別的樣本數量；TN 代表真陰性 (True Negative) 指模型正確地將預測為負類別的樣本數量；FP 代表假陽性 (False Positive) 指模型將負樣本錯誤地預測為正樣本的數量；FN

代表假陰性 (False Negative) 指模型將正樣本錯誤地預測為負樣本的數量。

針對本研究中提出的第一個方法，實驗 1 將採用章節 2.3 中提到的基於自然語言推理及未標記資料自訓練的方式，使用章節 2.2.4 中介紹到的預訓練語言模型作為骨幹網路 (backbone)，對比其實驗效果，並選擇效果最好的模型作為後續的基準模型。

針對本研究中提出的第二個方法，實驗 2 將採用章節 4.1 中提到的方法，將維基百科的網頁文本內容整理成訓練資料集合，用於微調 UniMC 模型。其中實驗 2.1 用於比較使用維基百科訓練資料微調前後的模型效果；實驗 2.2 用於探究訓練資料類別數量 n 的設置對實驗結果的影響。

針對本研究中提出的第三個方法，實驗 3 將採用章節 4.2 中提到的方法，使用維基百科訓練資料中的類別代替目標類別進行模型推理。實驗 3.1 比較使用前後的效果，並探究替代詞列表長度 k 的值對實驗結果的影響；實驗 3.2 目的是觀察篩選機制使用後的效果。

為探究維基百科微調與類別映射這兩個方法相互之間的關係，本文設置實驗組 4 以對兩者對模型的效果進行消融分析。使用上述三個方法最終確定的最佳模型，本文分別在實驗 5.1 和實驗 5.2 中將其與微調前的原始模型 UniMC、目前的最佳模型 Self-training 進行效果上的比較。

5.2.2 實驗設定

本研究中使用國家高速電腦中心台灣雲 (Taiwan computing cloud, TWCC) 的 PyTorch 開發型容器 (型號: cm.xsuper) 進行實驗，採用的實驗環境設定如表 5.1 所示。



Table 5.1: 實驗環境

項目	參數
CPU	: Intel(R) Xeon(R) Gold 6154 (八核心)
GPU	Nvidia Tesla V100 * 2
RAM	128 GB
OS	Ubuntu 20.04 LTS

Method	Backbone	Fine-tune dataset	Inference batch size
Prompt	GPT-3.5	—	-
TE-Wiki	BERT-base	Wikipedia, 3,387.028k examples	16
Self-training	DeBERTa-large	Unlabeled data from target dataset	16
UniMC	ALBERT-xxlarge	14 datasets for different tasks, 309.27k examples	16

Table 5.2: 各模型參數

5.3 實驗結果

本章節將對實驗結果進行分析。

5.3.1 實驗 1：零樣本文本分類模型性能對比實驗

該實驗設置了多個不同結構的模型，用以橫向評估各種模型在零樣本文本分類任務上的性能，其中上游模型包括生成式模型 GPT-3.5 的 text-davinci-003 三個 MLM (Masked Language Model) 模型 BERT-base、ALBERT-xxlarge、DeBERTa-large，方法包括使用 prompt、使用統一多選項方式微調 (UniMC)、使用維基百科資料並用自然語言推理的方式微調 (TE-Wiki)、使用自訓練的方式並用自然語言推理的方式進行微調 (Self-training)，各模型參數如表 5.2。

上述的模型中，GPT-3.5 未使用過訓練資料進行微調，TE-Wiki 使用了由維基百科網頁文本構建的 3,387,028 筆資料進行微調，Self-training 使用當前資料集的未標注資料進行微調，UniMC 則使用過來自 14 個資料集的 309,270 筆資料進行微調。上述模型中，除了 GPT-3.5 本文是以逐條調用 API 的方式進行推理的，其餘

Model	Backbone	Yahoo Answers	AG News	DBPedia	IMDB	Avg
Prompt tuning	GPT-3.5	45.12	65.32	N/A	82.56	64.33
TE-Wiki	BERT-base	56.54	79.62	93.10	62.02	72.82
Self-training	DeBERTa	62.04	81.40	94.52	92.52	82.62
UniMC	ALBERT-xxlarge	62.98	75.83	12.93	92.63	61.09

Table 5.3: 零樣本文本分類模型性能對比實驗結果

模型均設置推理時的 batch size 為 16。需要補充說明的是，對於 GPT-3.5 生成的答案，如果輸出的內容不包含類別，則計算生成內容和目標類別 GloVe 詞向量空間上的餘弦相似度，選擇最相似的類別作為輸出。

實驗最終結果如表5.3，顯示的數值為準確率 *%，由於 DBpedia 資料集數量較大，本研究並未用 GPT-3.5 在 DBpedia 上進行推理。

如表5.3所示，加粗部分為在某資料集上表現最佳的模型的準確率，Avg 表示在各資料集上的準確率的平均值。從結果中可以看出，對生成式模型使用提示微調的方式相對基於 MLM 模型微調的方式來說結果較差；使用 Self-training 方式的模型效果最好。而基於 MLM 模型微調的方式中，UniMC 在 Yahoo 資料集上的結果略勝於使用無標籤資料進行自訓練的模型，在其它兩個資料集中，Self-training 方式的模型效果最好。

5.1可以更直觀地看到各模型的性能。儘管 UniMC 在兩個資料集上的準確率最高，但是由於其在 DBpedia 上的準確率極低，導致其平均效果不如其它模型，因此，提升模型在 DBpedia 此類類別多、內容多的資料集上的性能，將會是提升 UniMC 模型性能的關鍵點。UniMC 模型在還未加入維基百科資料進行微調時，已經能在兩個資料集達到最高的準確率，其相比於其它模型，有著很大的提升可能，故而我們在後續的實驗中繼續使用 UniMC 作為基準模型，並著力於提升其在類別多、內容豐富的資料上的效果。

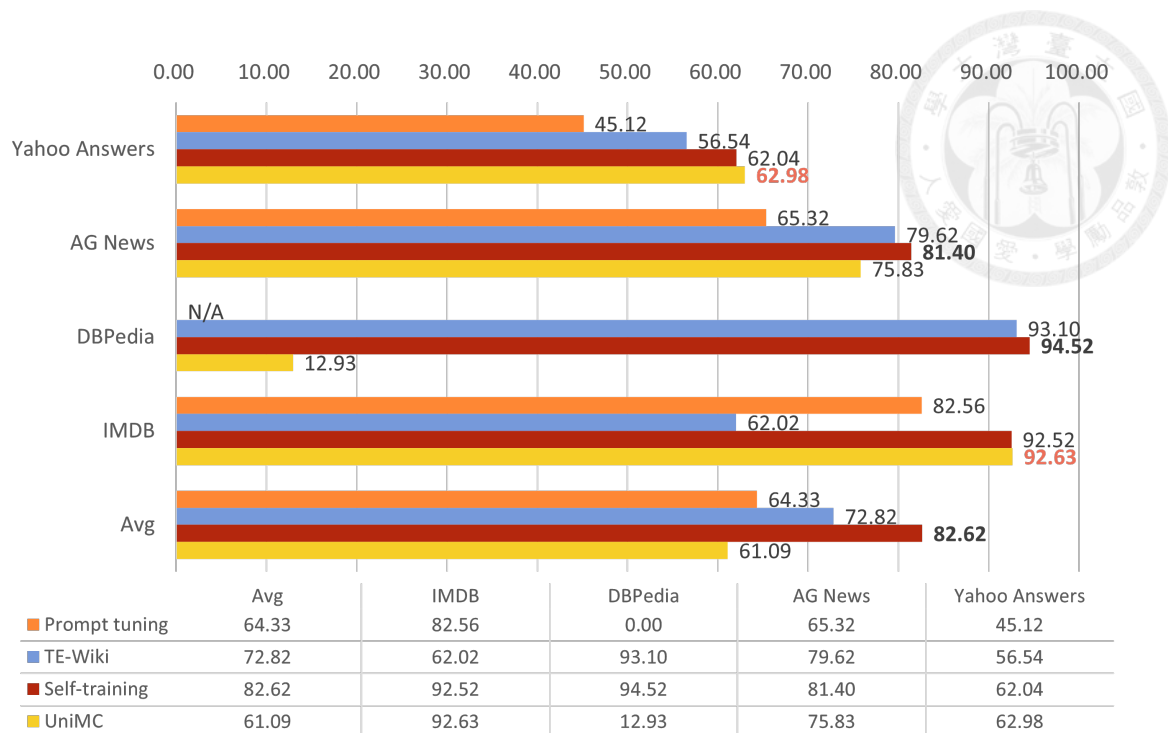


Figure 5.1: 零樣本文本分類模型性能對比實驗結果條形圖

	Yahoo Answers	AG News	DBpedia	IMDB	Avg
UniMC-ori	62.98	75.83	12.93	92.63	61.09
UniMC-5 classes	64.05	76.29	68.02	92.29	75.16

Table 5.4: 使用維基百科資料微調模型前後效果比對實驗

5.3.2 實驗 2.1：使用維基百科資料微調模型前後效果比對實驗

為探究使用維基百科構建訓練資料，並用於微調模型，其對 UniMC 在零樣本文本分類任務上是否能帶來效果上的提升，設置實驗 2.1 對使用維基百科微調模型前後進行結果對比。訓練過程中實驗參數設置如下：batch size 為 4，學習率為 $2e-5$ ，設置 early stopping，patience 值為 5，每 500 個 steps 保存一次 checkpoint，優化器為 AdamW。訓練資料處理如下：選擇訓練資料中 9,749 筆資料作為驗證集，占比 0.01；其餘 965,174 筆資料，作為訓練集，占比 0.99。類別數量 n 設置為 5。

使用維基百科資料對 UniMC 模型進行微調前後結果如表 5.4。其中 UniMC-ori 表示未進行微調之前的結果，UniMC-5 classes 為使用類別數量為 5 的資料集對

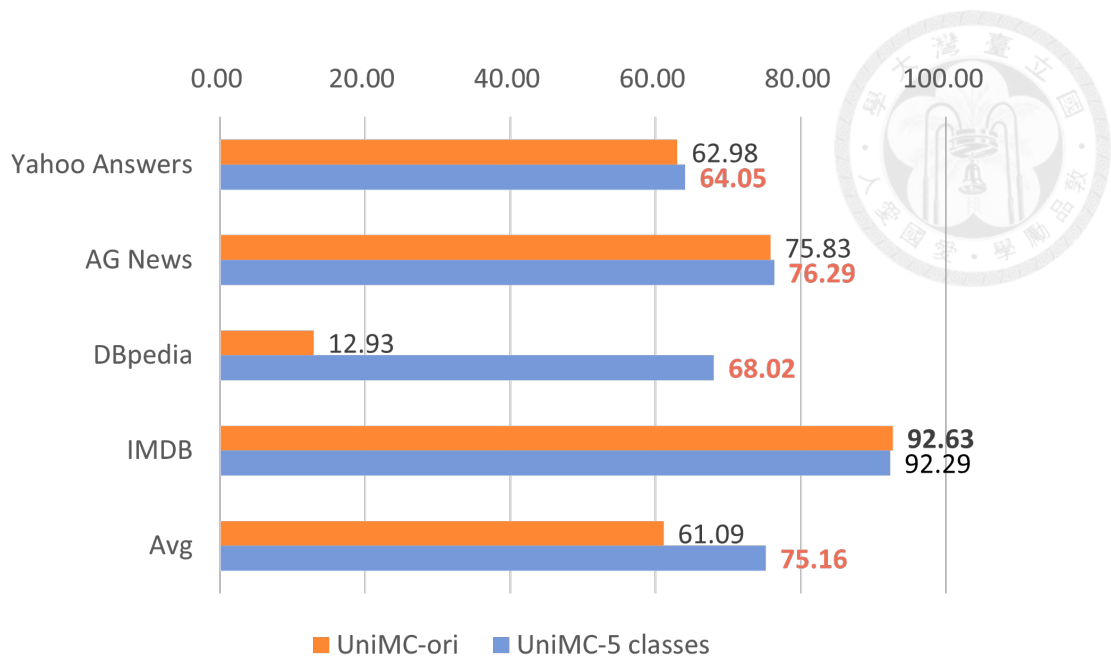


Figure 5.2: 零樣本文本分類模型性能對比實驗結果條形圖

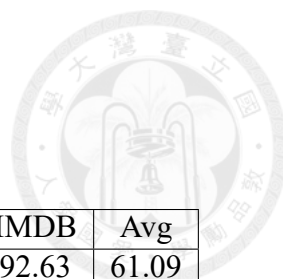
UniMC 模型進行微調的結果。實驗結果條形圖如5.2，可以直觀地觀察到微調過後的模型除了在 IMDB 資料集上的準確率略微下降，其餘三個資料集上結果均有所提升，其中在 DBpedia 上的準確率由 12.93% 提升至 68.02%，其效果最為明顯。

實驗 2.1 使用維基百科資料對 UniMC 進行微調後，平均準確率提升了 14.07%，說明該方法可行。

5.3.3 實驗 2.2：訓練資料類別數量探究實驗

為探究訓練資料的最佳類別數量為多少，類別數量是否對模型結果有影響，本實驗中設置對照組 UniMC-5 classes、UniMC-10 classes、UniMC-20 classes、UniMC-30 classes、UniMC-40 classes、UniMC-50 classes 分別表示類別 n 總數為 5、10、20、30、40、50 的訓練資料，這些訓練資料都使用單類別文本資料，即只有一個正面樣例，負面樣例標籤分別為 4、9、19、29、39、49 個。

實驗 2.2 結果如表5.5，結果條形圖如5.3。從實驗結果中並未找出類別數量與模型性能的明顯相關關係，但根據實驗結果，我們篩選出最佳模型為使用 40 個類



	Yahoo Answers	AG News	DBpedia	IMDB	Avg
UniMC-ori	62.98	75.83	12.93	92.63	61.09
UniMC-5 classes	64.05	76.29	68.02	92.29	75.16
UniMC-10 classes	61.02	54.24	86.07	66.13	66.87
UniMC-20 classes	63.33	76.93	87.89	87.72	78.97
UniMC-30 classes	64.49	66.87	91.48	86.24	77.27
UniMC-40 classes	65.12	78.45	93.79	89.20	81.64
UniMC-50 classes	64.66	73.72	88.78	85.45	78.15

Table 5.5: 訓練資料類別數量探究實驗結果

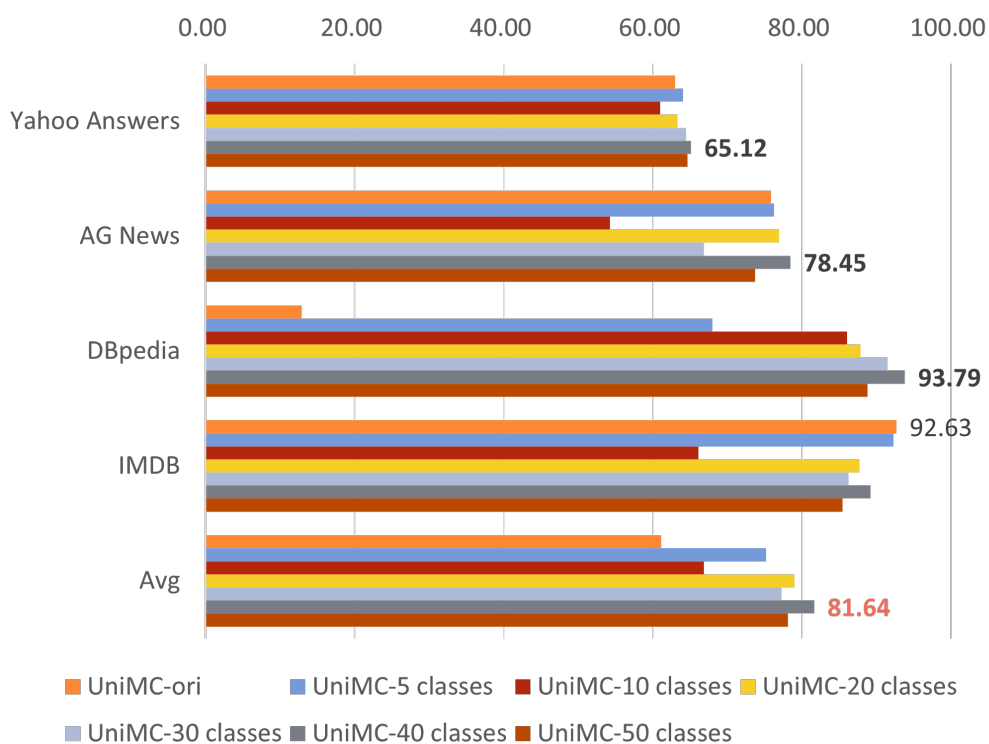


Figure 5.3: 零樣本文本分類模型性能對比實驗結果條形圖

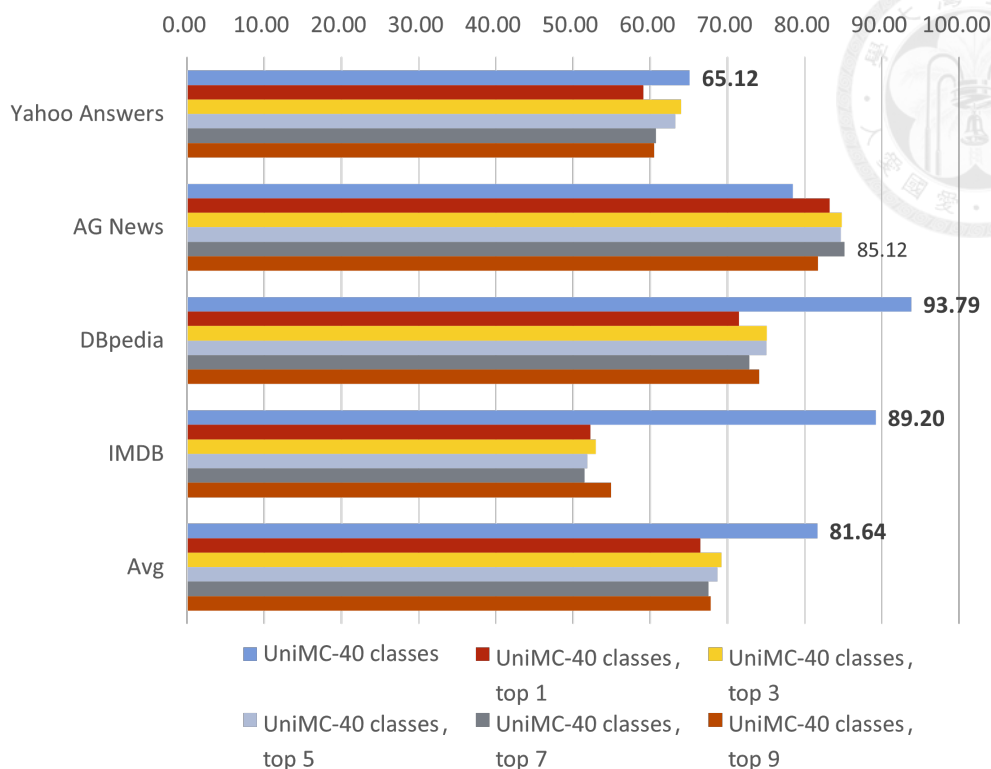


Figure 5.4: 替代詞列表效果探究實驗條形圖

	Yahoo Answer	AG News	DBpedia	IMDB	Avg
UniMC-40 classes	65.12	78.45	93.79	89.20	81.64
UniMC-40 classes, top 1	59.11	83.24	71.48	52.23	66.52
UniMC-40 classes, top 3	64.00	84.79	75.07	52.94	69.20
UniMC-40 classes, top 5	63.22	84.68	75.02	51.85	68.69
UniMC-40 classes, top 7	60.73	85.12	72.81	51.50	67.54
UniMC-40 classes, top 9	60.50	81.71	74.11	54.93	67.81

Table 5.6: 替代詞列表效果探究實驗結果

別進行微調的模型。後續實驗的模型將基於此模型。

5.3.4 實驗 3.1：替代詞列表效果探究實驗

為了探究使用替代詞及替代詞數量對模型預測結果的影響，本實驗設置選取替代詞列表前 k 分別為 1、3、5、7、9 個替代詞，在模型推理過程中替代目標類別，觀察各替代詞列表長度對實驗效果的影響。這裏用於產生目標類別及維基百科類別的 GloVe.6B，該模型是預訓練在六百萬個 tokens，包含 Gigaword5 和 Wikipedia2014 的語料庫上的模型，其輸出的向量維度為 300 維。

目標類別	Sports	Business & Finance	Entertainment & Music	Politics & Government
替代詞列表 (k=9) 及相似度	"Sports": 0.9999, "Water sports": 0.7659, "Air sports": 0.7651, "Whitewater sports": 0.7436, "Basketball": 0.5788, "Association football": 0.5695, "Baseball": 0.5571, "Olympic Games": 0.5134, "American football": 0.5097	"Finance":0.8673, "Business": 0.8342, "Industry": 0.6205, "Pharmaceutical industry": 0.5687, "Personal development": 0.5671, "Electronics companies": 0.5526, "Operations research": 0.5524, "Management": 0.5399, "Money": 0.5395	"Entertainment": 0.8611, "Music":0.8603, "Performing arts": 0.5869, "New media art": 0.5823, "Visual arts": 0.5794, "Musical groups": 0.5644, "Dance": 0.5585, "Video games": 0.5509, "Film": 0.4879	"Government": 0.8444, "Politics": 0.8334, "Political people": 0.7365, "Government agencies": 0.7239, "Public administration": 0.7052, "Politicians": 0.6373, "Criticism of religion": 0.6218, "People by legal status": 0.6036, "Social work": 0.5902

Figure 5.5: 目標類別與替代詞相似度實例

實驗 3.1 結果如表 5.6，結果條形圖如 5.4。從條形圖中可以很明顯看出除 AG News 資料集外，使用替代詞列表後模型預測結果反而明顯下降。因此，我們需進一步分析其原因。

我們從 Yahoo Answers 資料集中隨機調取四個類別，並取出其取前 9 時的替代詞列表及替代詞與目標類別的相似度，所取出的實例如圖 5.5。通過分析實力發現，在每個目標的前 9 個替代詞中，有可能存在與目標類別相似度很低的維基百科類別，有的替代詞甚至相似度低於 0.6。考慮到單純使用前 k 個替代詞反而可能在模型預測過程中帶來干擾，我們決定加入替代詞篩選機制，並設置實驗 3.2。

5.3.5 實驗 3.2：篩選機制效果實驗

實驗 3.2 的目的是為了加入篩選機制，消除不相似替代詞對模型性能的負面影響，探究篩選機制的使用效果。我們使用方法 4.2.4 中提到的替代詞篩選機制，設置閾值為 0.8，並使用目標類別確認，對替代詞數量分別取 5、7、9，使用篩選機制前後的結果進行了比較，實驗結果如圖 5.6，實驗數據如表 5.7。

加入篩選詞機制後，每個類別對應的平均替代詞長度如表 5.8，可以看到，加入替代詞篩選機制之後，替代詞列表平均長度減少至每個目標類別詞語對應 1-3

	Yahoo Answer	AG News	DBpedia	IMDB	Avg
UniMC-40 classes, top 5	63.22	84.68	75.02	51.85	68.69
UniMC-40 classes, top 5, filtering	65.70	84.34	93.69	89.20	83.23
UniMC-40 classes, top 7	60.73	85.12	72.81	51.50	67.54
UniMC-40 classes, top 7, filtering	65.43	84.52	93.69	89.20	83.21
UniMC-40 classes, top 9	60.50	81.71	74.11	54.93	67.81
UniMC-40 classes, top 9, filtering	64.30	84.52	93.69	89.20	82.93

Table 5.7: 篩選機制效果實驗結果

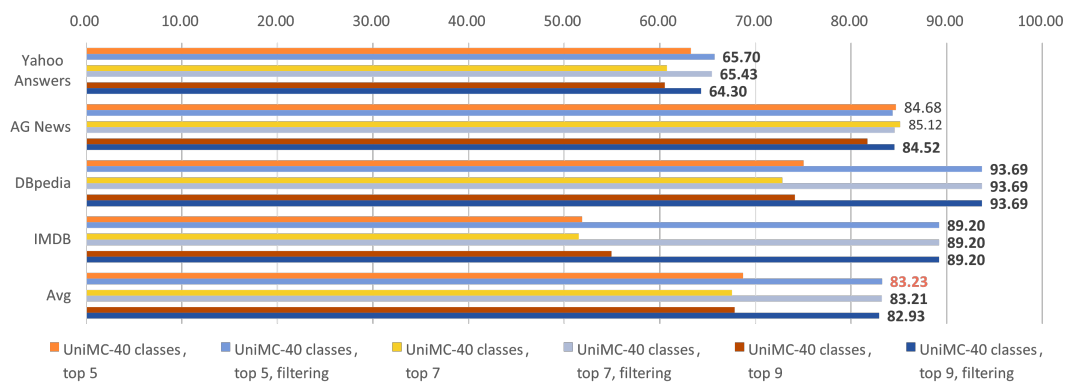


Figure 5.6: 篩選機制效果實驗結果條形圖

個維基百科類別，數量比取前 k 有所降低。加入篩選機制平均效果比未加入篩選機制高 15.12%，除了 $k = 5$ 、 $k = 7$ 時在 AG News 上，使用篩選機制比未使用略差，其餘條件下使用篩選機制效果均優於未使用篩選機制。由於在 $k = 5$ 並採用篩選機制時結果最佳，因此我們將該模型作為我們的最終模型，並命名為 UniMC-Wiki。

	AG News	Yahoo Answer	DBpedia	IMDB
Top 5, filtering mechanism	2	2.2	1.071	1
Top 7, filtering mechanism	2.5	2.6	1.071	1
Top 9, filtering mechanism	2.5	2.7	1.071	1

Table 5.8: 加入篩選機制之後每個類別對應平均替代詞列表長度

	Yahoo Answers	AG News	DBPedia	IMDB	Avg
UniMC-ori	62.98	75.83	12.93	92.63	61.09
UniMC-ori, label mapping	57.67	65.87	11.45	90.76	56.44
UniMC-40 classes	65.12	78.45	93.79	89.20	81.64
UniMC-40 classes, label mapping	65.70	84.34	93.69	89.20	83.23

Table 5.9: 維基百科微調與類別映射消融實驗結果

5.3.6 實驗 4：維基百科微調與類別映射消融實驗

實驗 4 使用消融實驗分析維基百科微調與類別映射兩個方法對模型性能的影響，以及兩個方法之間的相互影響。該實驗中分別設置以下四組對照組：“UniMC-ori”、“UniMC-ori, label mapping”、“UniMC-40 classes”、“UniMC-40 classes, label mapping”，其中“UniMC-ori”代表使用未經維基百科資料微調的模型，inference 目標類別上；“UniMC-ori, label mapping”代表使用未經維基百科資料微調的模型，inference 在類別映射後的替代詞列表中的維基百科類別上；“UniMC-40 classes”代表使用經維基百科資料微調的模型，inference 目標類別上；“UniMC-40 classes, label mapping”代表使用經維基百科資料微調的模型，inference 在類別映射後的替代詞列表中的維基百科類別上。其中經維基百科資料微調的模型，微調時的維基百科類別 n 取值為 40；類別映射的方法中取前 $k = 5$ 個最相似的詞，並加入篩選詞機制。

實驗 4 結果如表 5.9，結果條形圖如圖 5.7。

由實驗組一和實驗組三、四的比較可見，就算未使用類別映射，使用維基百科資料微調也能夠提升模型預測的準確率，該方法對模型效能的提升無需前提；實驗組一分別和實驗組二、實驗組四進行相互對比，可發現只有在使用了維基百科資料微調的前提下，使用類別映射才能對模型提升起正向作用，否則會使得模型性能下降。對比所有實驗組的平均準確率可以發現同時進行微調及類別映射能達到最佳效果。

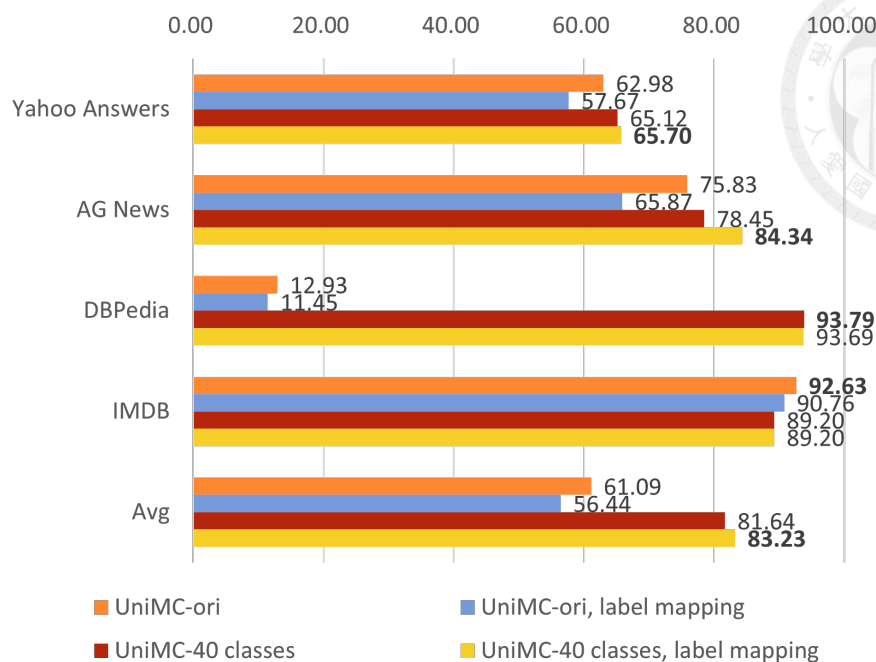


Figure 5.7: 維基百科微調與類別映射消融實驗結果條形圖

	Yahoo Answer	AG News	DBpedia	IMDB	Avg
UniMC-ori	62.98	75.83	12.93	92.63	61.09
UniMC-WiKi	65.70	84.34	93.69	89.20	83.23

Table 5.10: 研究方法使用前後模型性能對比實驗結果

5.3.7 實驗 5.1：研究方法使用前後模型性能對比實驗

實驗 5.1 使用最終模型 UniMC-Wiki 與微調前的模型 UniMC-ori 進行了比較，實驗結果如表 5.10，條形圖如圖 5.8。可以看出，在使用本研究方法後，除了在 IMDB 資料集上的結果略微下降外，在其餘資料集上的性能均大幅提升，其中在 DBpedia 上，使用本研究方法準確率比未使用高 80.76%。在整體性能上，相比起原模型，UniMC-Wiki 平均準確率提升了 22.14%。

為了探究本研究方法為何對情感分類任務較差，本研究中調取了最終模型在 IMDB 資料集上的替代詞列表，發現替代詞列表中的標籤為“positive”和“negative”，與原目標類別一致，這說明維基百科類別中無法找到和情感類別標籤“positive”和“negative”相似度較高的詞，由於維基百科為主題相關資料集，在

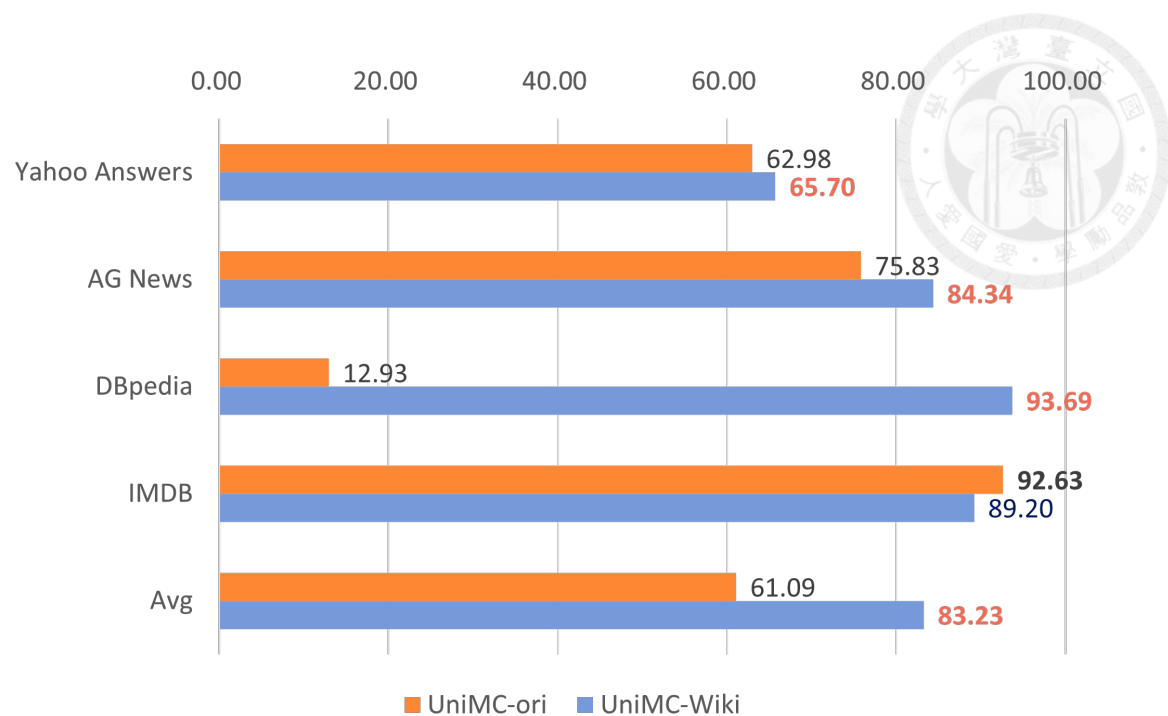


Figure 5.8: 研究方法使用前後模型性能對比實驗結果條形圖

	Yahoo Answer	AG News	DBpedia	IMDB	Avg
Self-training	62.04	81.40	94.52	92.52	82.62
UniMC-WiKi	65.70	84.34	93.69	89.20	83.23

Table 5.11: UniMC-WiKi 與最佳模型性能對比實驗結果

情感相關任務上仍有一定局限性，因此對 IMDB 資料集未起作用。

5.3.8 實驗 5.2：UniMC-WiKi 與最佳模型性能對比實驗

實驗 5.2 對 UniMC-Wiki 及目前最佳模型 Self-training 之間的性能進行了對比，實驗設置模型推理時 batch size 為 16，實驗結果如表 5.11，條形圖如圖 5.9。

最終的實驗結果顯示，UniMC-Wiki 模型在 Yahoo Answers 和 AG News 資料集表現較 Self-training 好，而 Self-training 模型在另外兩個資料集上更佳。UniMC-Wiki 模型平均準確率略微高於 Self-training (0.61%)，達到了與最佳模型相當的效果。

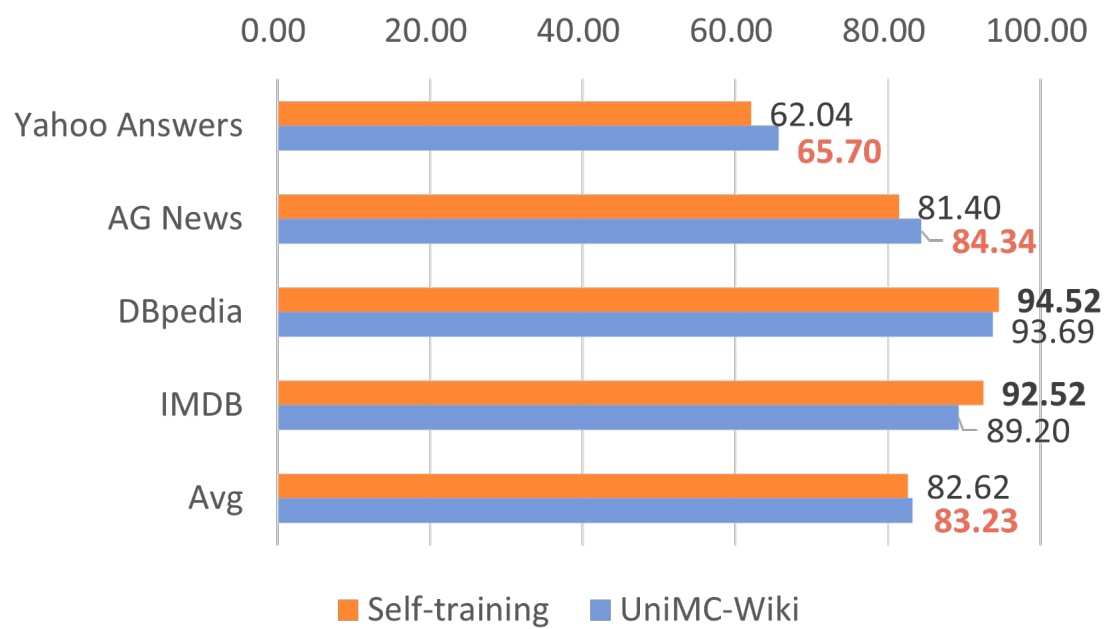


Figure 5.9: UniMC-WiKi 與最佳模型性能對比實驗結果條形圖





第六章 結論和未來工作

本章節介紹結論及未來工作。

6.1 結論

為了解決零樣本分類任務數據缺乏和領域依賴性的問題，本文提出了三種解決方法。首先，使用既包含用於描述任務的 prompt，又能夠將所有類別整合到一起讓模型學習的 UniMC 結構對模型進行訓練，從而使模型具有更好的任務和類別感知能力。其次，利用公開領域的知識庫維基百科構建分類任務訓練資料，通過微調模型學習多領域的先驗知識，提高模型的泛化能力。最後，通過將維基百科類別映射到零樣本文本類別的替代詞列表中，使用代替詞列表中的詞語代替零樣本文本類別進行分類，促進了模型所學知識在新任務上的遷移，從而進一步提高分類精度。

為此，本文設計了五組實驗以對提出的方法進行驗證。實驗組一中，通過和大量的模型進行對比實驗，證明使用 UniMC 模型，在兩個資料集上相比起僅使用自然語言推理方法的模型效果好，因而證實了 UniMC 結構的可行性，其效能提升關鍵點在於 DBpedia。實驗組二中，通過使用公開領域的知識構建成文本分類訓練資料對 UniMC 進行微調，使得 UniMC 在不同主題分類任務得到了提升，說明使用公開域的知識來訓練模型，能夠一定程度地提高模型的泛化能力。實驗組三

中證明，使用類別映射，並加入替代詞篩選機制的方法能夠提升模型預測的準確率。實驗組四證實類別映射方法起作用須有維基百科資料微調模型的前提。實驗組五顯示相較原始模型，使用本研究中的方法，平均準確率可提升 22.14 %；本研究中的方法達到了目前最佳模型相當的效果。

此外，實驗中發現使用維基百科類別代替零樣本文本類別的方式，僅適用於主題分類任務，對於情感分析任務沒有太大的幫助，我們分析這是由於維基百科的類別難以匹配到情感分類類別，使用維基百科文本整理成訓練資料微調而成的模型對情感相關任務具有一定局限性。

儘管本研究相比目前最佳模型，結果上的提升並不大，但值得一提的是，由於本研究的方法訓練出來的模型可直接應用於各個資料集，而 Self-training 方法需要根據每個資料集使用未標記資料進行訓練，本研究的模型具備“一次訓練，多任務通用”的特點，因而本研究的内容仍然具有很大的參考價值。

6.2 未來工作

基於以上研究，本文認為未來工作的一個方向是進一步探索和改進方法二、三。目前的實驗結果顯示，本研究方法在主題分類任務上取得了良好的效果，但對情感分析任務的幫助有所局限。因此，可以進一步研究如何尋找更適合情感分類的知識源，並探索如何以更有效的方式將這些知識引入模型中。

此外，本文在研究中發現，對於 UniMC 結構中的提示詞的設計也會很大程度地影響實驗結果，然而該模型的提示詞是由研究者自定義設計的，我們或許可以進一步改進 UniMC 的模型結構，設計一種可以在訓練過程中同時對提示詞進行調整，自動產生提示詞的方法，從而提高模型的性能和準確率。

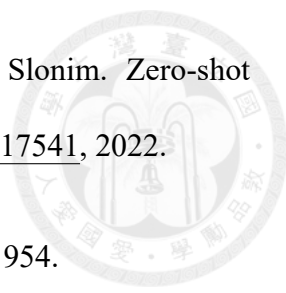
總之，未來的工作可以著重於改進和擴展所提出的方法，進一步提高零樣本分類模型的性能。同時，可以探索其他創新的解決方案，以應對數據缺乏和領域依賴性的挑戰，從而促進文本分類技術在實際應用中的進一步發展。



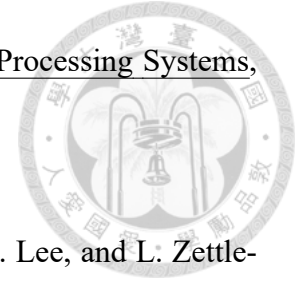


參考文獻

- [1] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. Advances in neural information processing systems, 13, 2000.
- [2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.
- [3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.
- [4] 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.
- [5] H. Ding, J. Yang, Y. Deng, H. Zhang, and D. Roth. Towards open-domain topic classification. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations, pages 90–98, 2022.
- [6] S. T. Dumais et al. Latent semantic analysis. Annu. Rev. Inf. Sci. Technol., 38(1):188–230, 2004.

- 
- [7] A. Gera, A. Halfon, E. Shnarch, Y. Perlitz, L. Ein-Dor, and N. Slonim. Zero-shot text classification with self-training. arXiv preprint arXiv:2210.17541, 2022.
- [8] Z. S. Harris. Distributional structure. Word, 10(2-3):146–162, 1954.
- [9] P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654, 2020.
- [10] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
- [11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942, 2019.
- [12] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):1995, 1995.
- [13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.
- [14] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development, 1(4):309–317, 1957.
- [15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
- [16] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instruc-

tions with human feedback. Advances in Neural Information Processing Systems, 35:27730–27744, 2022.



- [17] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.
- [18] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.
- [20] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620, 1975.
- [21] D. Svozil, V. Kvasnicka, and J. Pospichal. Introduction to multi-layer feed-forward neural networks. Chemometrics and intelligent laboratory systems, 39(1):43–62, 1997.
- [22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.
- [23] P. Yang, J. Wang, R. Gan, X. Zhu, L. Zhang, Z. Wu, X. Gao, J. Zhang, and T. Sakai. Zero-shot learners for natural language understanding via a unified multiple choice perspective. arXiv preprint arXiv:2210.08590, 2022.
- [24] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet:

Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32, 2019.

- [25] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. Advances in neural information processing systems, 28, 2015.