Practical Natural Language Processing 推薦序

Practical NLP 直接把焦點放在一個被忽略的族群：業界的從業人員與商業主管！雖然坊間有許多書籍專門探討基礎的 ML 演算法，但這本書揭露真實的系統結構：從電子商務 app 到虛擬助理。本書描繪了現代生產系統的真實情況，不僅教導深度學習，也傳授經驗法則及處理線，這些做法定義了部署 NLP 系統的（真實）先進方法。作者們會將鏡頭拉遠，教導如何提出問題，同時也會大膽地將鏡頭拉近，聚焦於髒汙的細節，包括如何處理雜亂的資料，以及維持即時系統。對渴望實際建構和部署 NLP 的專業人士而言，本書具備不可估量的價值。

—*Zachary Lipton*，*Carnegie Mellon* 大學助理教授，*Amazon AI* 科學家，
Dive into Deep Learning 的作者

本書彌合自然語言處理（NLP）的研究和實際應用之間的鴻溝，它介紹許多運用 NLP 的熱門領域，包括醫療保健、電子商務與金融，以簡明且易懂的方式完成核心任務。整體而言，這本很棒的手冊將教你如何在你的行業中充分利用當前的 NLP。

—*Sebastian Ruder*，*Google DeepMind* 研究科學家

市面上的電腦科學書籍有兩種：非學術人員難以親近，卻可以讓你深度了解一個領域的學術教科書，以及列舉特定問題解決方案的「配方書」，但它們不提供協助讀者到處使用配方的技術基礎。本書同時提供這兩個區塊的優點，它既全面，且平易近人。可以幫讀者建立堅實的自然語言處理基礎…如果你想要在 NLP 中從零進步到一，看這本書就對了！

—*Marc Najork*，*Google AI* 研究工程總監，
ACM & IEEE 研究員

坊間有許多討論編程技巧的教科書、研究論文和書籍，但沒有一本書告訴你如何從零開始建構端對端 NLP 系統。很高與看到這本實際應用 NLP 的書籍，它填補了這個有迫切需求的空缺。作者精心、深思熟慮且清楚地介紹 NLP 的每一個層面，它們都是在建構大型實用系統時必須注意的地方。這本書也納入大量的範例，以及各種應用領域和產業鏈。任何一位有抱負的 NLP 工程師、想要運用語言技術來創業的企業家，以及想要看到發明出來的東西被真正的用戶使用的學術研究人員都需要這本書。

—Monojit，Microsoft Research India 首席研究員，印度理工學院克勒格布爾校區，阿育王大學，海得拉巴國際資訊科技研究所的兼職教授

本書彌合了理論與實務之間的空白，它不但解釋底層的概念，也關注跨越各種產業鏈的實際部署。書中有許多經歷實戰且千錘百煉的實務建議，無論是調整開源程式庫的參數、設定建立模型所需的資料處理線，還是進行優化以快速執行推斷。這是 *NLP app* 工程師必讀的書籍。

—Vinayak Hegde，Microsoft For Startups 常駐 CTO

本書展示如何將 NLP 付諸實踐。它填補了 NLP 理論與實務工程之間的空白。作者們完成一項偉大的工作——將製作高品質機器學習系統所需的深奧設計藝術以及架構簡化。真希望我在職涯早期就能夠接觸這本書，這樣我就可以避免許多錯誤了…對每一位想要開發穩健的、高性能的 NLP 系統的人來說，我相信這本書都是必不可少的讀物。

—Siddharth Sharma，Facebook 機器學習工程師

我覺得這本書不僅是 NLP 從業者必備的，對研究社群而言也是一本寶貴的參考書，可讓他們了解真正的 app 的問題空間。我很喜歡這本書，希望它可以成為一個長期的專案，加入最新的 NLP 應用趨勢！

—Mengting Wan，Airbnb 資料科學家（ML & NLP），Microsoft 研究員

引言

近年來，自然語言處理（NLP）領域已經發生了翻天覆地的變化，無論是在方法論上，還是它所支持的應用。方法論方面的進步包括新的文件表示法，以及新的語言合成新技術。隨之而來的是新的應用，從開放式對話系統，到使用自然語言來建立可解釋模型。最後，這些進步讓 NLP 在相關領域站穩腳跟，例如電腦視覺與推薦系統。在 Amazon、三星與美國國家自然科學基金會的支持之下，我的實驗室正在進行其中的一些專案。

準備使用 NLP 技術的從業者也想要隨著 NLP 擴展至這些令人期待的新領域。我在加州大學聖地牙哥分校任教的資料科學課程（CSE 258）中看到越來越多學生在進行以 NLP 為主的專案，該堂課通常是計算機科學系中學生最多的一堂。對工程師、產品經理、科學家、學生和希望用自然語言資料來建構 app 的愛好者而言，NLP 正迅速成為必備的技能。一方面，現在我們比過往任何時刻都更容易使用新的 NLP 與機器學習工具和程式庫來建立自然語言模型。但另一方面，教導 NLP 的資源必須考慮日益成長且多樣化的對象。對最近才開始採用 NLP 的機構，或是初次使用自然語言資料的學生來說更是如此。

在過去幾年裡，我很開心可以和 Bodhisattwa Majumder 一起開發令人期待的 NLP app 和進行交流，因此，很高興聽到他（與 Sowmya Vajjala、Anuj Gupta、Harshit Surana）想要寫一本關於 NLP 的書籍。他們都有廣泛的 NLP 擴展經驗，無論是在初創企業的早期階段、MIT Media Lab、Microsoft Research 還是 Google AI 裡面。

我很開心聽到他們在書中採取端對端方法，讓這本書很適合一系列的情境，也可以幫助讀者在建構 NLP app 時，面對各種錯綜複雜的選項。他們對於現代 NLP 應用（例如聊天機器人）以及跨學科主題（例如電子商務和零售）的關注更是讓我充滿期待。這些主題對產業主管和研究人員來說特別實用，也是目前的教科書很少討論的重要主題。這本書不僅是探索自然語言處理領域的首要資源，也是讓經驗豐富的從業人員探索既有領域的最新發展的指南。

— *Julian McAuley*
加州大學聖地牙哥分校，
計算機科學與工程教授

前言

自然語言處理（NLP）是結合計算機科學、人工智慧與語言學的領域，其目的是建構可以處理和了解人類語言的系統。自 NLP 在 1950 年代出現一直到最近，它都一直是學術界與研究實驗室的領域，需要長期的正規教育和培訓。但 NLP 在過去十年來的突破，讓它在零售、醫療保健、金融、行銷、人力資源，以及許多領域中的使用量越來越多，這種趨勢有很多驅動因素：

- 在業界無處不在，而且容易使用的 NLP 工具、技術與 API。現在是建構快速的 NLP 解決方案的最佳時機。
- 由於出現更具解釋性且更通用的方法，即使是複雜的 NLP 任務的基本性能也有所改善，那些任務包括開放領域對話任務、問題回答，這些都是之前無法做到的。
- 有越來越多機構（包括 Google、Microsoft 與 Amazon）投入大量資金在更具互動性的消費產品上，在這些產品中，語言是主要的交流媒介。
- 有越來越多開源資料組，以及使用它們時的標準性能數據可用，它們是這場革命的催化劑，防止專用的資料組被少數機構與個人壟斷，因而阻礙 NLP 的發展。
- NLP 已經可以處理英語等主要語言之外的語言了，即使是未被普遍數位化的語言也有資料組和專屬模型，進而導致現在每個人都可以在智慧手機使用近乎完美的自動機器翻譯工具。

隨著 NLP 越來越普遍，有越來越多 NLP 系統建構者想要從這個主題的有限經驗和理論知識中突破，本書希望從應用的角度解決這個需求。本書的主旨是引導讀者在商業環境中建立、迭代和擴展 NLP 系統，並且為不同的產業鏈定製它們。

著作動機

現在已經有很多熱門的 NLP 書籍了，其中有些是教科書，側重理論，有些則是透過大量的範例程式來介紹 NLP 概念，有些專門介紹特定的 NLP 或機器學習程式庫，並提供使用這些程式庫來解決各種 NLP 問題的「操作」指南。那麼，為什麼你需要另一本 NLP 書籍？

我們曾經在頂尖的大學與技術公司建構與擴展 NLP 解決方案超過十年，在指導同事和其他工程師時，我們發現業界的 NLP 實踐法與新人（尤其是剛開始製作 NLP 的工程師）的技術之間有不小的差距。這些差距在我們為 NLP 業界專家舉辦的研討會裡面更是明顯，我們在那裡發現商業與工程主管也有這些差距。

大部分的線上課程與書籍都以玩具用例與流行的資料組（通常是大型的、乾淨的、具備良好定義的）來探討 NLP 問題。雖然這種做法可以傳授普通的 NLP 方法，但我們認為它無法提供足夠的基礎來解決真正的新問題，以及開發具體的解決方案。據我們所知，在建構真正的 app 時經常遇到的一些問題都不是用現有的資源來處理的，那些問題包括收集資料、處理有雜訊的資料與訊號、漸進開發解決方案，還有將解決方案當成更大規模的 app 的一部分來部署時可能出現的問題。我們也看到，大部分的場景都沒有開發 NLP 系統的最佳實踐法。我們認為應該用一本書來填補這個空白，這就是這本書誕生的原因！

哲學

我們想要提供一個具備整體性的實際觀點來協助讀者在更大型的生產環境之中成功地建構真正的 NLP 解決方案。因此，大部分的章節都會展示相關 Git 版本庫裡面的程式碼。本書也提供廣泛的參考文獻，讓讀者可以更深入地研究。本書將從一個簡單的解決方案開始，採取業界常見的最簡可行產品（minimum viable product，MVP）方法來逐步建構更複雜的解決方案。我們也會根據我們的經驗與教訓提供一些建議。在可能的情況下，每一章都會討論該主題的最新技術，大部分的章節都有真實用例的案例研究。

想像一下，你要在你的機構中建構聊天機器人或原文分類系統。起初，你可能只有少量資料，或完全沒有資料可以使用，此時適合採取基本的解決方案，例如規則式系統，或傳統的機器學習。但是隨著資料的累積，你或許會開始使用更精密的 NLP 技術（通常是資料密集型的），包括深度學習。這個旅途的每一步都有數十種方法可選，本書將協助你走出這個迷宮。

範圍

本書將全面介紹如何建構真正的 NLP app。我們將介紹典型的 NLP 專案的完整生命周期——從資料收集到監控模型，其中有些步驟適合任何一種 ML 處理線，有些只用於 NLP。我們也會介紹特定任務的案例研究與領域專屬的指南，說明如何從零開始建構 NLP 系統。我們特別加入豐富的任務，包括原文分類、問題回答、資訊提取，以及對話系統。類似地，我們也提供如何在不同的領域執行這些工作的方法，包括電子商務、醫療保健、社交媒體及金融等領域。因為我們探討的主題與情境有一定的深度和廣度，所以我們不會一步一步地解釋程式碼和所有概念。對於實作的細節，我們有詳細的原始碼 notebook。本書的程式段落包含核心的邏輯，通常會跳過初階步驟，例如設定程式庫或匯入程式包，因為它們都可以在相關的 notebook 裡面找到。為了介紹如此寬廣的概念，我們提供超過 450 個廣泛的參考資料來深入研究這些主題。本書是一本日常食譜，可在你建構任何 NLP 系統時提供實用的觀點，也可以當成跳板，在你的領域中擴展 NLP 的應用。

誰該看這本書

本書適合正在為真正的用例建構 NLP app 的所有人，包括軟體開發者、測試員、機器學習工程師、資料工程師、MLOps 工程師、NLP 工程師、資料科學家、產品經理、人力資源主管、VP、CXO，及新公司創辦人，此外，也包括涉及資料建立和標注程序的所有人——簡而言之，就是在產業中，以任何形式參與 NLP 系統的建構的每一個人。雖然並非所有章節都適合每一位角色，但我們會盡量清楚地解釋，不使用難解的術語，讓讀者更直觀地理解。我們相信對每一位想要全面了解如何建構 NLP app 的人而言，每一章都有一些東西可以引起興趣。

有些章節不需要太多編程經驗就可以理解，而且程式的部分可以視需要跳過。例如，沒有任何編程經驗的讀者可以了解第 1 章與第 9 章的前兩節，或第 11 章的「資料科學流程」與「在你的機構中成功發展 AI」這兩節。當你閱讀本書時，你會在各章裡面發現更多這種小節。但是，為了讓本書、其 notebook 及參考文獻提供最大的益處，我們希望讀者具備下列的背景知識：

- 具備中等水準的 Python 編程能力。比如說，了解 Python 的功能，例如串列生成式、知道怎麼撰寫函式與類別，以及使用既有的程式庫。熟悉軟體開發周期（SDLC）的各個層面，例如設計、開發、測試、DevOps 等。

- 具備基本的機器學習知識，包括熟悉常用的機器學習演算法，例如羅吉斯迴歸、決策樹，還有在 Python 中利用既有的程式庫（例如 scikit-learn）來使用演算法。
- 了解 NLP 的基礎知識有幫助，但不是必要的。知道原文分類以及專名個體識別等任務的概念也有幫助。

你會學到什麼

我們的讀者主要包括為各種產業鏈建構真正的 NLP 系統的工程師與科學家。常見的職稱包括：軟體工程師、NLP 工程師、ML 工程師及資料科學家。本書應該也可以幫助產品經理和工程主管，但可能無法幫助頂尖的 NLP 研究者，因為我們並未探討 NLP 概念的深層理論和技術細節。透過這本書，你將：

- 了解在 NLP 領域中，廣泛的問題陳述方式、任務與解決方案。
- 知道如何實作與評估各種 NLP 應用程式，並且在過程中，使用機器學習和深度學習方法。
- 根據各種商業問題與產業鏈來微調 NLP 解決方案。
- 為特定的任務、資料組，和 NLP 產品階段評估各種演算法與做法。
- 規劃 NLP 產品周期，並且藉著遵守 NLP 系統的發布、部署與 DevOps 來製作軟體解決方案。
- 從商業與產品主管的角度了解 NLP 的最佳實踐法、機會和路線圖。

你也會學到如何針對不同的產業鏈（例如醫療保健、金融與零售）調整解決方案，此外，你將了解在各種產業鏈可能遇到的注意事項。

本書架構

本書分成四大部分，圖 P-1 描述書中的各章。與其他章不相連的獨立章是最容易在閱讀時跳過的。

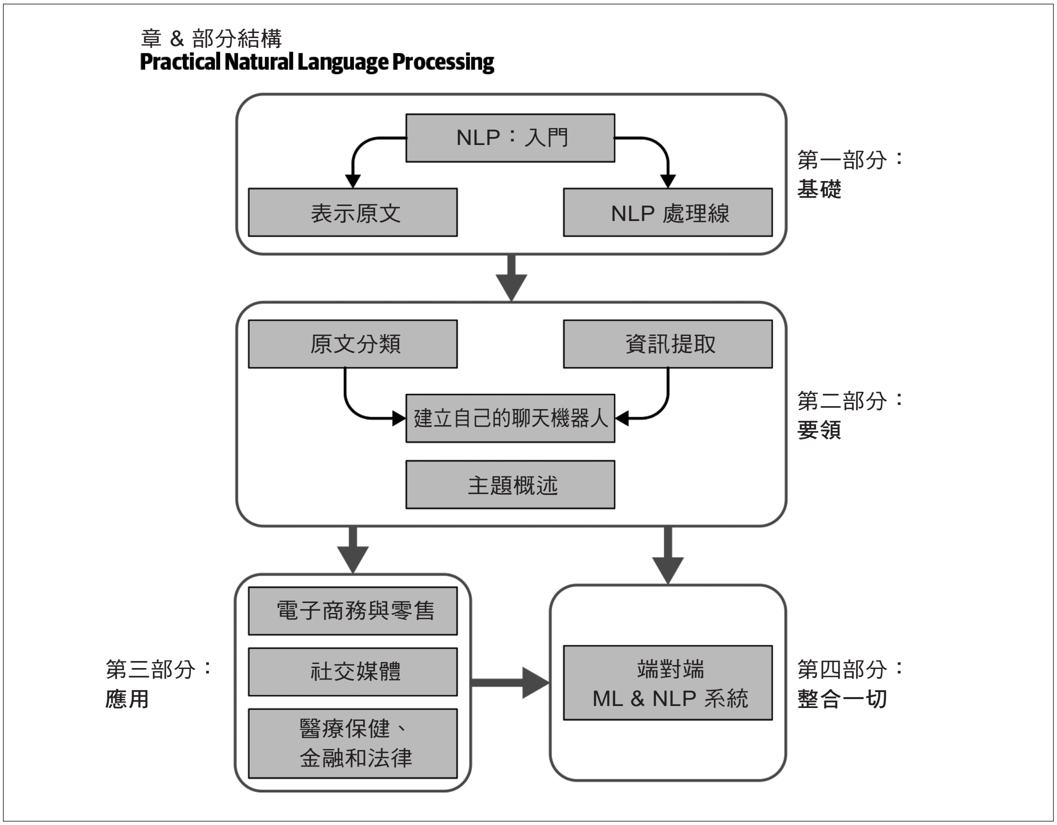

圖 P-1　本書章節架構

第一部分，**基礎**，是本書其餘部分的基石，它概要介紹 NLP（第 1 章）、討論在建構 NLP 系統時的典型資料處理和模型建立處理線（第 2 章），並介紹在 NLP 內表示文字資料的各種方式（第 3 章）。

第二部分，**要領**，關注最常見的 NLP 應用，特別注重真實世界的用例。在可能的情況下，我們會展示眼前問題的多種解決方案，並示範如何在不同的選項之間進行選擇。我們介紹的應用包括原文分類（第 4 章）、資訊提取（第 5 章）與聊天機器人的建構（第 6 章）。我們也會介紹其他的應用，例如搜尋、主題建模、原文摘要生成，以及機器翻譯，並討論實際的用例（第 7 章）。

第三部分，**應用**（第 8–10 章）特別關注大量使用 NLP 的三個產業鏈，並詳細討論這些領域的具體問題，以及如何使用 NLP 來解決這些問題。

最後，第四部分（第 11 章）藉著處理 NLP 系統的實務端對端部署所涉及的問題，來整合學過的所有內容。

如何閱讀本書

這本書的讀法取決於讀者的角色和目的。對鑽研 NLP 的資料科學家或工程師而言，我們建議先閱讀第 1–6 章，再關注有興趣的特定領域或次級問題。對領導者而言，我們建議關注第 1、2 和 11 章，這類讀者或許也可以閱讀第 3–7 章的案例研究，了解關於從零開始建構 NLP app 的流程的概念。產品主管應該深入研究相關章節的參考文獻，以及第 11 章。

NLP 在各種領域裡面的應用方式可能與第 3–7 章討論的一般問題的應用方式不同。這就是我們更關注電子商務、社交媒體、醫療保健、金融和法律等特定領域的原因。如果你的興趣或工作使你進入這些領域，你可以深入研究這些章節，及其參考文獻。

本書編排方式

本書採取下列編排方式：

斜體（*Italic*）

代表新術語、URL、email 地址、檔名，與副檔名。

定寬體（`Constant width`）

在長程式中使用，或是在文章中代表變數、函式名稱、資料庫、資料型態、環境變數、陳述式、關鍵字等程式元素。

定寬粗體（**`Constant width bold`**）

代表應由使用者親自輸入的命令或其他文字。

等寬斜體（*`Constant width italic`*）

代表使用者所提供的值，或由上下文決定的值。

第一章

NLP：入門

語言並非只是單字，它是一種文化、
一種傳統、統一群體的機制，
創造一個群體的完整歷史，
這些事物都會在語言中呈現。

—*Noam Chomsky*

想像一位虛構的人物，John Doe。他是一家快速發展的科技初創公司的首席技術官。在忙碌的一天裡，John 醒來，與他的數位助理對話：

John：「今天天氣怎樣？」

數位助理：「今天室外溫度 37 度，不會下雨。」

John：「我有什麼行程？」

數位助理：「下午 4 點有個戰略會議，下午 5:30 有個全體會議。根據今天的交通狀況，建議在上午 8:15 之前出發，前往辦公室。」

John 在穿衣服時，詢問助手他的穿搭風格：

John：「我今天要怎麼穿？」

數位助理：「白色應該不錯。」

你可能用過 Amazon Alexa、Google Home 或 Apple Siri 等智慧助理來做類似的事情。我們不是用程式語言和這些助理交談的，而是用我們的自然語言——大家用來溝通的語言。自古以來，這種自然語言就是人類交流的主要媒介。但是電腦只會處理二進制資料，也就是 0 和 1。雖然我們可以用二進制來表示語言資料，但怎麼讓機器了解這些語言？這就是自然語言處理（NLP）的用武之地了，它是計算機科學的一個領域，專門處理分析、建模和了解人類語言。每一種涉及人類語言的智慧型 app 背後都有一些 NLP。本書將解釋什麼是 NLP，以及如何使用 NLP 來建構和擴展智慧型 app。由於 NLP 問題的開放性，一個特定的問題可以用幾十種替代方案來解決，本書將協助你在迷宮般的選項中找出方向，並建議你如何根據問題做出最好的選擇。

在深入研究如何為各種應用場景實作 NLP 解決方案之前，本章的目的是快速介紹什麼是 NLP。我們會先概述真實場景中的許多 NLP 應用，再介紹建構各種 NLP 應用基礎的各種任務。接下來，我們會從 NLP 的角度來了解語言，並討論為何 NLP 是個難題。在此之後，我們將簡介經驗法則、機器學習與深度學習，再介紹一些常用的 NLP 演算法。然後，我們會演練一個 NLP 應用。在本章的最後，我們將概述本書的後續章節。圖 1-1 是用各種 NLP 任務與應用來整理的章節概要。

圖 1-1　NLP 任務與應用

我們先來看一下經常在日常生活中看到，而且將 NLP 當成主要元件來使用的一些應用程式。

在真實世界中的 NLP

NLP 是我們在日常生活中使用的廣泛軟體的重要成分。在這一節，我們將介紹一些關鍵的應用，並且看一下在各種不同的 NLP 應用程式中常見的任務。本節進一步解釋圖 1-1 中的應用，本書其餘的部分會更詳細的解釋它們。

核心應用：

- email 平台，例如 Gmail、Outlook 等，廣泛使用 NLP 來提供一系列的產品功能，例如垃圾郵件分類、收件箱優先順序、行事曆事件提取、自動完成等。我們將在第 4 章與第 5 章更詳細討論其中的一些功能。
- 語音助理，例如 Apple Siri、Google Assistant、Microsoft Cortana 與 Amazon Alexa 都依靠廣泛的 NLP 技術來與用戶互動，了解用戶的指令，以及做出相應的回應。第 6 章會介紹這種系統的關鍵層面，該章主要討論聊天機器人。
- 現今網際網路的基石——現代搜尋引擎（例如 Google 與 Bing）重度使用 NLP 來處理各種次級任務，例如了解指令、擴展查詢、問題回答、資訊檢索、排序和分類結果等，族繁不及備載。第 7 章會討論其中的一些次級任務。
- 現今世界越來越常使用機器翻譯系統（例如 Google Translate、Bing Microsoft Translator 與 Amazon Translate）來處理廣泛的場景與商務用例。這些服務都是 NLP 的直接應用。第 7 章會介紹機器翻譯。

其他的應用有：

- 跨領域的機構分析社交媒體源，來深入了解顧客的聲音，我們會在第 8 章討論。
- NLP 被廣泛地用來解決 Amazon 這類的電子商務平台的各種用例，包括：從產品說明提取相關資訊，以及了解用戶的評論。詳情見第 9 章。
- NLP 被進一步用來解決醫療保健、金融和法律等領域的用例。詳情見第 10 章。
- Arria [1] 等公司正使用 NLP 技術來自動產生各種領域的報告，包括天氣預報和金融服務。

- NLP 是拼寫和語法糾正工具的支柱，例如 Grammarly，以及在 Microsoft Word 與 Google Docs 裡面的拼寫檢查工具。
- *Jeopardy!* 是熱門的電視智力競賽節目。在節目中，參賽者要根據以答案形式提供的各種線索，以問題形式作出正確的回答。IBM 開發了 Watson AI 來與節目的頂級參賽者競賽，贏得 100 萬美元的頭獎，比世界冠軍更多。Watson AI 是用 NLP 技術來建構的，它是 NLP 機器人贏得世界級競賽的例子之一。
- 許多學習 / 評估工具及技術都使用 NLP，例如 Graduate Record Examination（GRE）等考試的自動評分、抄襲檢測（例如 Turnitin）、智慧教學系統，及語言學習 app（例如 Duolingo）。
- NLP 被用來建構大型的知識庫，例如 Google Knowledge Graph，它在搜尋與問題回答等應用中很實用。

以上並非詳盡的清單，許多其他的應用領域也越來越頻繁地使用 NLP，新的 NLP 應用也正在不斷出現。我們的重點是藉著討論各種類型的 NLP 問題，以及如何解決它們，來介紹這些應用的建構方法背後的哲學。為了讓你了解本書即將介紹的東西，並理解在建構這些 NLP app 時的細微差別，我們來看一些重要的 NLP 任務，它們是許多 NLP app 和產業用例的基礎。

NLP 任務

有一些基本的任務經常在各種不同的 NLP 專案中出現。由於這些任務具備重複性質和基礎性質，所以它們已經被廣泛地研究了。掌握它們可以讓你做好準備，建構橫跨產業鏈的各種 NLP app。（我們也會看到圖 1-1 之中的一些任務。）我們先簡單介紹它們：

語言建模

這種任務是根據前面的單字來預測句子接下來的單字。這個任務的目標是學習特定語言中出現一系列單字的機率。語言建模可以為各種問題建構解決方案，例如語音辨識、光學字元辨識、手寫辨識、機器翻譯和拼寫糾正。

原文分類

這種任務是根據原文的內容，將原文分類至一組已知類別。到目前為止，原文分類是最流行的 NLP 任務，它也被用來開發各種工具，包括 email 垃圾郵件識別和情緒分析。

資訊提取

顧名思義，這種任務可從原文中提取相關資訊，例如 email 裡面的行事曆事件，或社交媒體貼文中提到的人名。

資訊檢索

這種任務是在一個大集合中尋找與用戶查詢指令有關的文件。Google Search 之類的 app 是著名的資訊檢索用例。

對話代理人

這種任務是建構以人類語言交談的對話系統。Alexa、Siri 是這項任務常見的 app。

原文摘要生成

這項任務是為較長的原文創造精簡的摘要，同時保留核心內容，以及整體文章的意思。

問題回答

這種任務是建構可以自動回答以自然語言提出的問題的系統。

機器翻譯

這是將一段原文從一種語言翻譯成另一種語言的任務。Google Translate 之類的工具是這種任務常見的應用。

主題建模

這種任務的目的是找出一群文件的主題結構。主題建模是常見的原文挖掘（text-mining）工具，文學與生物資訊學等領域都使用它。

圖 1-2 是開發這些任務的解決方案的相對難易度。

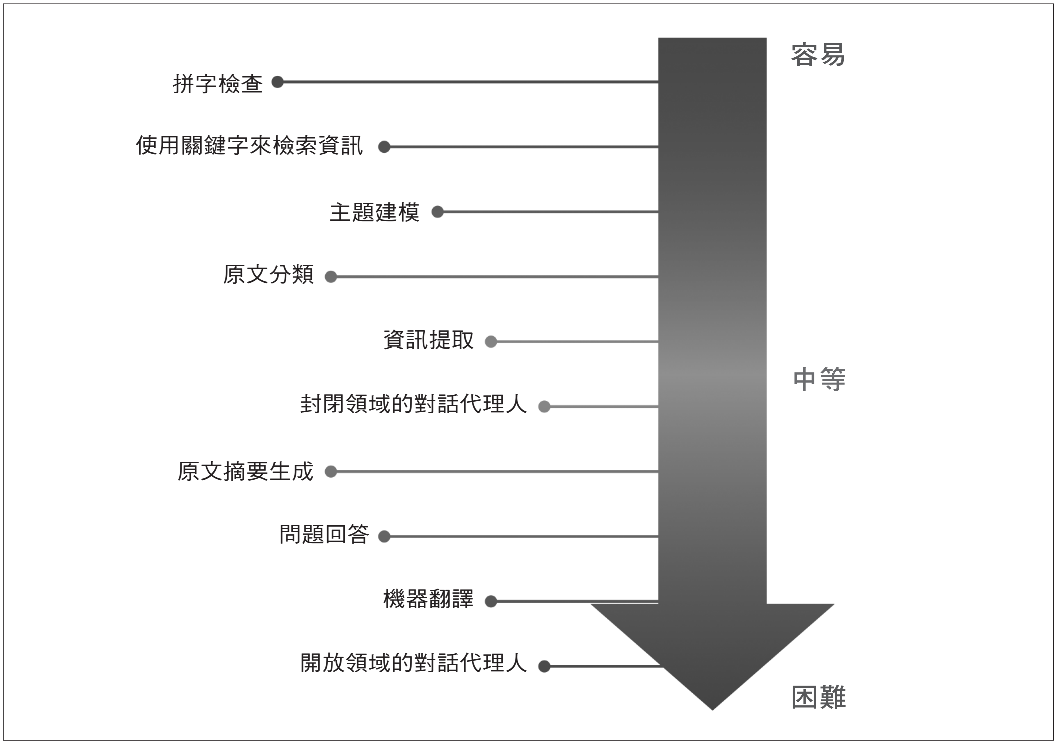

圖 1-2　NLP 任務的相對難易度

在本書的其餘章節中，我們將了解這些任務的挑戰，並學習如何為特定的用例開發解決方案（甚至圖中的困難任務）。為此，了解人類語言的本質，以及將處理語言的工作自動化是很有幫助的，接下來兩節將簡要介紹它們。

什麼是語言？

語言是一種結構化的溝通系統，它包含各種成分（例如字元、單字、句子等）的複雜組合。語言學是有系統地研究語言的學科。為了研究 NLP，我們必須了解語言學的一些語言結構，本節將介紹它們，並說明它們與之前列出的 NLP 任務有什麼關係。

人類語言可以分成四個主要成分：音素（phoneme）、詞素（morpheme）與詞元（lexeme）、語法（syntax），以及語境（context）。NLP 應用程式需要關於這些元素的各種級別的知識，從語言的基本聲音（音素）到有意義地表達事情的原文（語境）。

圖 1-3 是這些語言的主要成分、它們包含哪些東西，以及哪些 NLP 應用需要該項知識。圖中有一些尚未介紹的詞（例如解析、單字 embedding 等）會在接下來的前三章介紹。

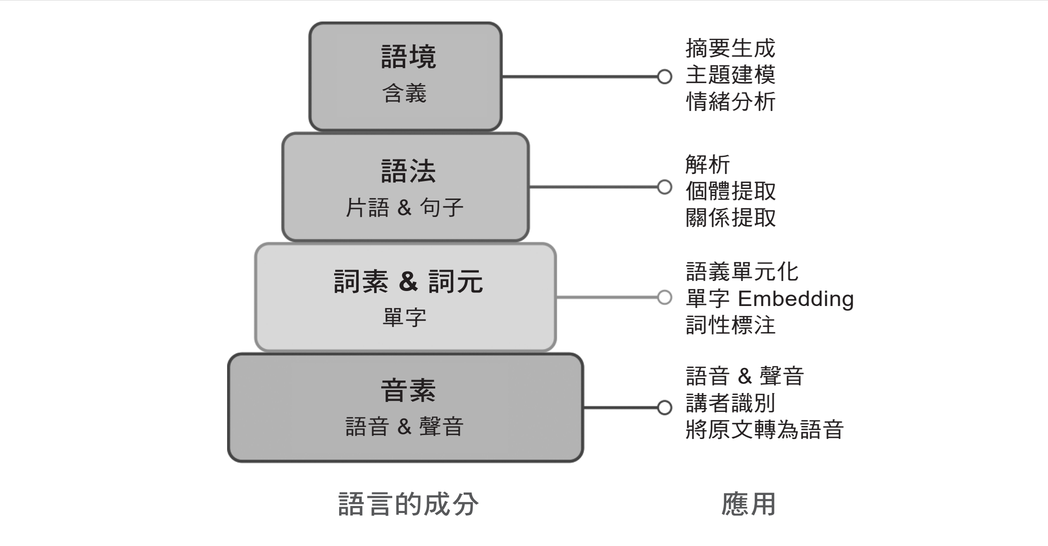

圖 1-3　語言的成分及其應用

語言的成分

我們先來了解一下這些語言成分是什麼，以及 NLP 涉及的挑戰背景。

音素

音素是語言最小的聲音單位。它們本身可能沒有任何意義，但是當它們與其他音素結合時，即可產生意義。例如，標準英語有 44 個音素，它們是單一字母或字母的組合 [2]。圖 1-4 是這些音素和範例單字。音素在牽涉「語音理解」的應用中特別重要，例如語音辨識、將語音轉換為原文，以及將原文轉換為語音。

子音音素及單字範例		母音音素及單字範例	
1. /b/ - bat	13. /s/ - sun	1. /a/ - ant	13. /oi/ - coin
2. /k/ - cat	14. /t/ - tap	2. /e/ - egg	14. /ar/ - farm
3. /d/ - dog	15. /v/ - van	3. /i/ - in	15. /or/ - for
4. /f/ - fan	16. /w/ - wig	4. /o/ - on	16. /ur/ - hurt
5. /g/ - go	17. /y/ - yes	5. /u/ - up	17. /air/ - fair
6. /h/ - hen	18. /z/ - zip	6. /ai/ - rain	18. /ear/ - dear
7. /j/ - jet	19. /sh/ - shop	7. /ee/ - feet	19. /ure/[4] - sure
8. /l/ - leg	20. /ch/ - chip	8. /igh/ - night	20. /ə/ - corn<u>er</u>（'schwa' - 不加重的母音，接近 /u/）
9. /m/ - map	21. /th/ - thin	9. /oa/ - boat	
10. /n/ - net	22. /**th**/ - then	10. /**oo**/ - boot	
11. /p/ - pen	23. /ng/ - ring	11. /oo/ - look	
12. /r/ - rat	24. /zh/[3] - vision	12. /ow/ - cow	

圖 1-4　音素與範例

詞素與詞元

詞素（morpheme）是最小的有意義語言單位，它是由音素組成的。並非所有詞素都是單字，但所有的字首（prefix）與字尾（suffix）都是詞素。例如，在單字「multimedia」中，「multi-」不是單字，而是一個字首，當它和「media」放在一起時會改變意思，所以「multi-」是個詞素。圖 1-5 是一些單字及其詞素。對「cats」與「unbreakable」這種單字而言，它們的詞素只是整個字的組成部分，至於「tumbling」與「unreliability」這種單字，將單字分解成詞素時，會產生一些變化。

unbreakable *un + break + able*	cats *cat + s*
tumbling *tumble + ing*	unreliability *un + rely + able + ity*

圖 1-5　詞素範例

詞元（lexeme）是詞素因為彼此間的意義產生的結構性變化。例如，「run」與「running」屬於同一個詞元形式。形態分析（morphological analysis）是藉著研究單字的詞素與詞元來分析單字的結構，它是許多 NLP 任務的基本元素，那些任務有語義單元化（tokenization）、詞幹提取（stemming）、學習單字 embedding、詞性標注等，接下來的章節會介紹它們。

語法

語法是在一種語言中，使用單字與子句來建構正確句子的規則。在語言學裡面，語法結構有許多表達形式。有一種常見的做法是用解析樹（parse tree）來表示句子。圖 1-6 是兩個英語句子的解析樹。

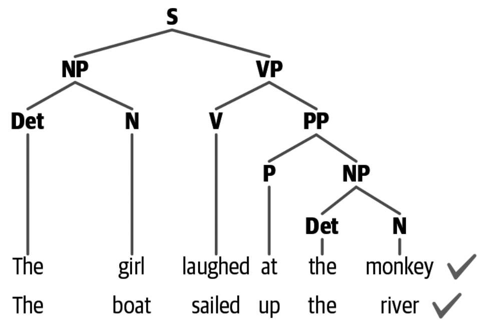

圖 1-6　兩個語法相似的句子的語法結構

它表現了語言的階層結構，最底層是單字，接下來是詞性標注，接下來是子句，最後是最高層的句子。在圖 1-6 裡面的兩個句子有相似的結構，因此有相似的語法解析樹。在這種表示法中，N 代表名詞，V 代表動詞，P 代表介詞，NP 是名詞子句，VP 是動詞子句。圖中的名詞子句是「The girl」與「The boat」，動詞子句是「laughed at the monkey」與「sailed up the river」。語法結構遵守語言的語法規則（例如句子是由一個 NP 與一個 VP 組成的），這套規則也引導語言處理工作的一些基本任務，例如解析。解析是自動建構這種樹狀結構的 NLP 任務。個體提取與關係提取是建構這種解析知識的 NLP 任務，我們將在第 5 章深入探討。注意，上述的解析結構是英語特有的，有些語言的語法有很大的差異，那種語言的處理方法也會相應改變。

語境

語境就是將語言的各種部分組合起來，來傳達特定的意思。語境包括長期參考（long-term references）、世界知識、常識，以及單字與子句的字面意義。這意味著一個句子可能因為背景的不同而改變，因為單字與子句有時有很多種意思。語境通常是由語義（semantic）和語用（pragmatic）組成的。語意是在沒有外部情境的情況下，單字與句子的直接意義。語用則包含對話時的世界知識與外部情境，讓我們可以推斷隱含的意義。諷刺偵測、摘要生成與主題建模等複雜的 NLP 任務都是大量使用語境的任務。

語言學是針對語言進行的研究，因此它本身就是個廣闊的領域，我們只介紹一些基本的概念來說明語言知識在 NLP 中的作用。不同的 NLP 任務需要不同程度的語言元素建構知識，感興趣的讀者可以參考 Emily Bender [3, 4] 探討 NLP 語言學基礎的書籍，來進一步研究。知道語言的基本元素有哪些之後，我們來看看為什麼電腦很難理解語言，以及為何 NLP 具有挑戰性。

為何 NLP 具有挑戰性？

為什麼 NLP 是個有挑戰性的問題領域？人類語言的模糊性和創造性只是讓 NLP 領域的門檻很高的其中兩個特性而已。本節將從語言的模糊性開始談起，詳細地探討每一種特性。

模糊性

模糊性代表意義的不確定性。大部分的人類語言在本質上都是模棱兩可的。看看這段句子：「I made her duck.」這個句子有很多種意思。第一種是：我為她煮了一隻鴨。第二種是：我讓她彎腰來閃避一個物體（此外還有其他的意思，留給讀者思考）。這句話的模糊性來自「made」這個字的使用，這句話的意思取決於它的背景，如果它出現在親子故事書裡面，它應該是第一個意思，但是如果它出現在體育領域書籍裡面，它應該是第二個意思。我們看到的例子是一個直接句（direct sentence）。

如果句子涉及象徵語言（figurative language），也就是習慣用語（idiom），模糊性還會提升。例如，「He is as good as John Doe.」試著回答「How good is he?」答案依 John Doe 有多好而定。圖 1-7 是個語言模糊性的例子。

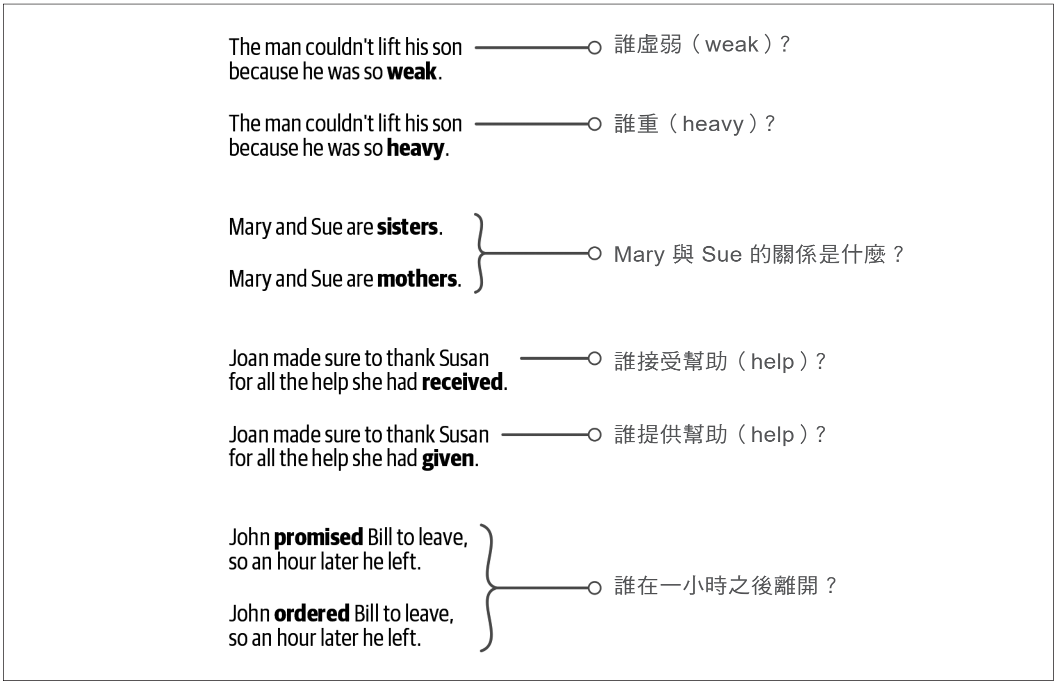

圖 1-7　語言模糊性案例，來自 Winograd Schema Challenge

這些例子來自 Winograd Schema Challenge [5]，書名來自史丹福大學的 Terry Winograd 教授。在這種模式中，成對的句子裡面只有少數單字不同，但是這些句子的意思往往因為這種小差異而天差地別。雖然人類很容易解決這些例子的模糊性，但大部分的 NLP 技術都無法解決它們。考慮圖中的成對句子與它們旁邊的問題，經過一些思考，我們應該可以知道一個單字的變化如何改變答案。你可以做另一個實驗：在現成的 NLP 系統（例如 Google Translate）中嘗試各種例子，看看這種模糊性如何影響（或不影響）系統的輸出。

常識

「常識」是人類語言的關鍵層面之一。它是大部分的人都知道的事實集合。在任何對話中，我們都假設這些事實是眾所週知的，因此不會特別說出它們，但它們與句子的意思有關。例如這兩個句子：「man bit dog」與「dog bit man」，我們都知道第一個句子不太可能發生，但第二個非常有可能，為什麼？因為我們都「知道」人極不可能咬

狗，此外，狗會咬人是大家都知道的事情。我們需要這種知識才能說第一句不太可能發生，而第二句可能發生。注意，這兩句話都沒有提到這個常識。人類一向使用常識來理解和處理任何語言。上面的兩個句子的語法很相似，但是電腦很難區分兩者，因為它缺乏人類所擁有的常識。在 NLP 中，將人類的所有常識植入計算模型是一項關鍵的挑戰。

創造性

語言並不是只根據規則，它也有創造性層面，任何語言都有各種風格、方言、流派和變體，詩歌就是一個很好的語言創造性案例。讓電腦理解創造性不但在 NLP 裡面是個難題，在一般的 AI 中也是如此。

跨語言的多樣性

對世界上大多數的語言來說，任何兩種語言的詞彙之間都沒有直接的對映關係。所以我們很難將一種語言的 NLP 解決方案移植到另一種語言。能夠處理一種語言的解決方案可能完全無法處理另一種語言。這意味著我們只能從「建構一個處理所有語言的解決方案」和「單獨為各種語言建構解決方案」之中選擇一種，第一種做法在概念上非常困難，但另一種既費力且費時。

以上的所有問題都使得 NLP 成為一個具有挑戰性但又值得研究的領域。在了解如何用 NLP 處理其中一些挑戰之前，我們來了解一下常見的 NLP 問題解決方法。在更深入了解 NLP 的各種做法之前，我們先來看一下機器學習與深度學習和 NLP 有什麼關係。

機器學習、深度學習與 NLP：概要

大致上說，人工智慧（AI）是電腦科學的一個分支，它的目的是建構一個系統來執行需要人類智慧的任務，有時它也稱為「機器智慧」。AI 的基礎是在 1950 年 Dartmouth College 的一個研討會上奠定的 [6]。最初，AI 都是以邏輯、經驗法則和規則系統構成的。機器學習（ML）是 AI 的分支，其目的是開發可以從大量案例中學習，進而自動執行任務的演算法，因此不需要人工創造規則。深度學習（DL）是機器學習的分支，它採用人工神經網路結構。ML、DL 與 NLP 都是 AI 的次級領域，圖 1-8 是它們之間的關係。

在圖中，雖然 NLP、ML 與 DL 之間有一些重疊的區域，但它們仍然完全不相同的研究領域。與 AI 的其他早期工作一樣，早期的 NLP 應用也採用規則與經驗法則。但是在過去數十年裡，NLP 應用程式的開發已經被 ML 的方法深深地影響，最近也經常有人使用 DL 來建構 NLP 應用程式。所以，本節將簡單地介紹 ML 與 DL。

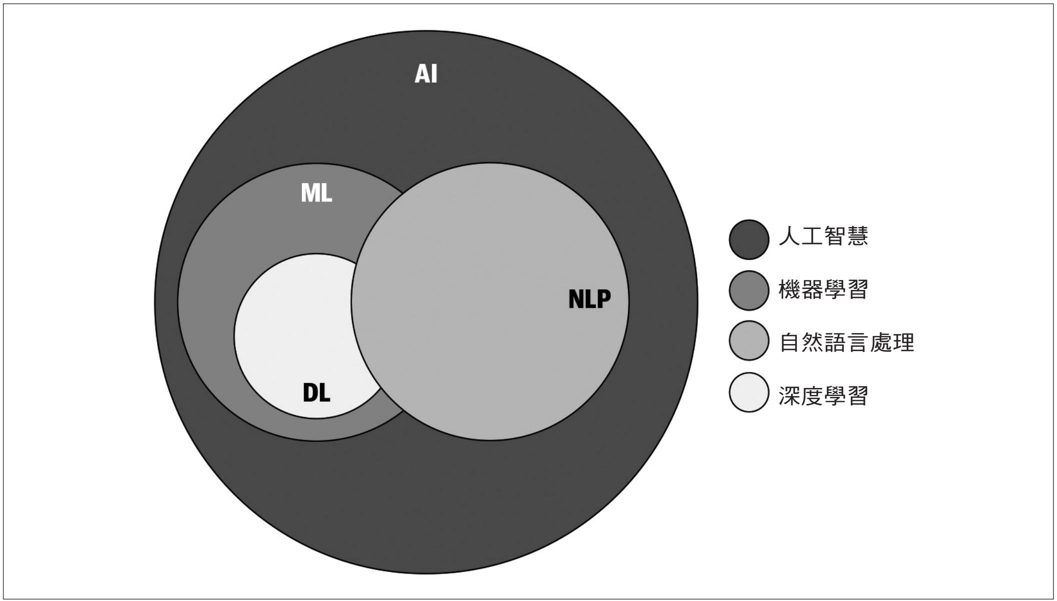

圖 1-8　NLP、ML 與 DL 之間的關係

ML 的目標是在沒有明確指引的情況下從案例（稱為「訓練資料」）「學習」如何執行任務。這通常是藉著建立訓練資料的數值形式（稱為「特徵」），並使用這種形式來學習案例的模式來完成的。機器學習演算法可以分成三種主要模式：監督學習、無監督學習，及強化學習。監督學習的目標是使用許多輸入 / 輸出形式的範例來學習將輸入對映至輸出的函數，那些輸入 / 輸出稱為**訓練資料**，輸出則被具體稱為**標籤**（*label*）或**基準真相**（*ground truth*）。與語言有關的監督學習問題包括將 email 訊息分類為垃圾郵件和非垃圾郵件，根據這兩個類別的幾千個案例。這是在 NLP 中常見的場景，這本書將不斷展示監督學習的例子，尤其是在第 4 章。

無監督學習是根據輸入資料，在沒有任何參考輸出的情況下尋找隱含模式的機器學習方法，也就是說，相較於監督學習，無監督學習使用的是大量的無標籤資料。在 NLP 中，這種任務的其中一種案例就是在不了解有哪些主題的情況下，在大量文字資料中找出可能的主題。這種任務稱為**主題建模**（*topic modeling*），我們將在第 7 章討論它。

真實世界的 NLP 專案經常採用半監督學習，這種方法使用一小組有標籤的資料，以及大量無標籤的資料。半監督學習使用這兩種資料組來學習手頭的任務。最後一種，但也很重要的是，強化學習處理的是「缺少大量有標籤或無標籤資料，並且採用試誤法」的學習任務。這種學習是在自足（self-contained）的環境之中完成的，而且是透過環境促成的回饋（獎勵或懲罰）來改善的。這種學習方式在應用 NLP 中還很罕見，它在電腦遊戲（例如圍棋或西洋棋）、自動駕駛汽車的設計中，以及在機器人技術中比較常見。

深度學習是機器學習的一個分支，它採用人工神經網路結構。神經網路的概念來自人腦的神經，以及它們彼此的互動方式。在過去的十年裡，深度學習的神經結構已經成功地改善各種智慧型應用的性能了，例如圖像與語音辨識和機器翻譯，這導致業界採用深度學習解決方案的數量激增，包括 NLP 應用程式。

本書將討論如何使用這些方法來開發各種 NLP 應用程式。我們接著來討論解決 NLP 問題的各種做法。

NLP 的方法

解決 NLP 問題的方法通常分成三類：經驗法則、機器學習，及深度學習。本節只是各種方法的介紹 —— 如果你無法完全理解這些概念，不用擔心，本書的其餘部分將更詳細地探討它們。我們先來討論經驗法則式 NLP。

採用經驗法則的 NLP

與其他早期的人工智慧系統相似的是，早期的 NLP 系統也是試著為眼前的任務構建規則來建立的。為了制定將要納入程式中的規則，開發者必須具備該領域的專業知識。這種系統也需要字典與同義詞詞典之類的資源，通常要用一段時間來編譯它們，並將它們數位化。以詞典為主的情緒分析（lexicon-based sentiment analysis）就是使用這種資源來設計規則來解決 NLP 問題的一種例子。它使用原文中的肯定詞和否定詞的數量來判斷原文的情緒。第 4 章會簡單介紹這種技術。

除了字典與同義詞詞典之外，也有人建構出更精密的知識庫來協助處理一般的 NLP 問題，尤其是以規則為主的 NLP。其中一個例子是 Wordnet [7]，它是個資料庫，裡面有單字以及單字之間的語義關係，這種關係包括同義詞、下位詞（hyponym）與分體詞（meronym）。同義詞代表不同的單字有相似的意義。下位詞代表 is-type-of（一種 …）關係。例如，棒球、相撲和網球都是體育的下位詞。分體詞代表 is-part-of（… 的一部分）關係。例如，手與腿都是身體的分體詞。這些資訊在建構語言的規則式系統時都很有用。圖 1-9 是單字之間的這些關係，它是用 Wordnet 繪成的。

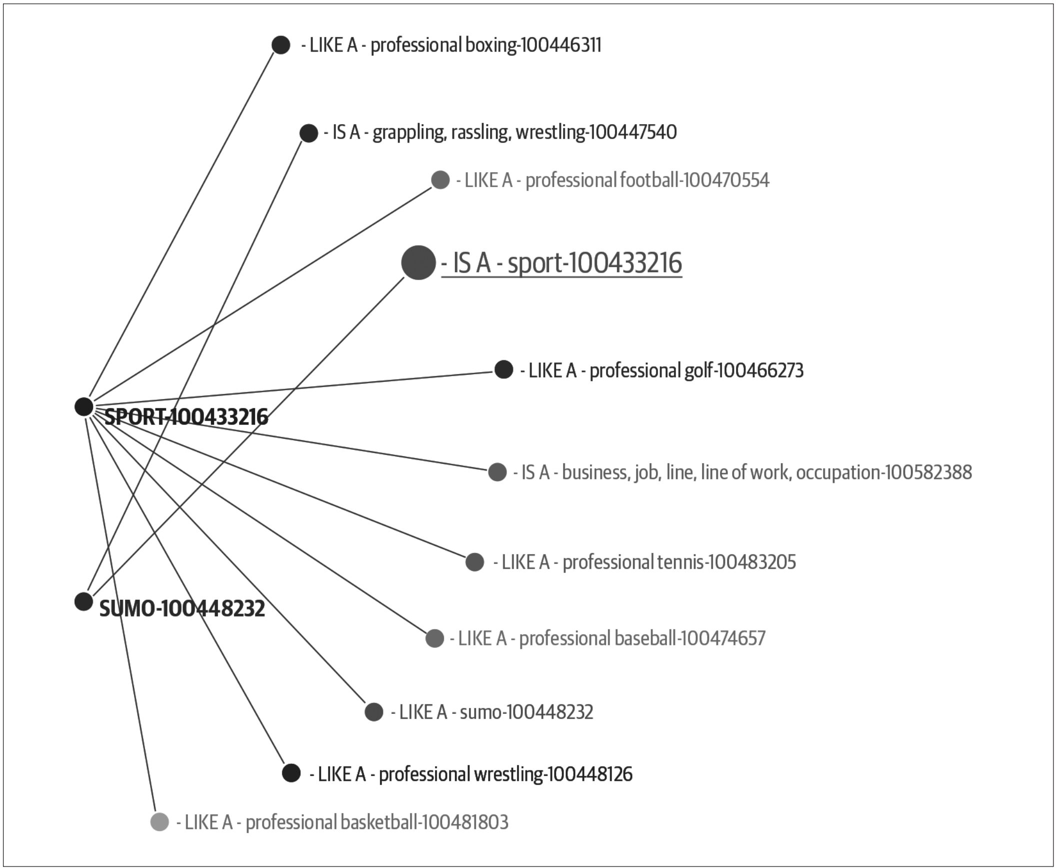

圖 1-9　單字「sport」的 Wordnet 圖 [8]

最近，Open Mind Common Sense [9] 之類的知識庫也納入世界常識知識，可以協助建構這種規則式的系統。雖然我們目前看到的辭彙資源大都是基於單字級別的資訊，但規則式系統不只採用單字，也可以使用其他形式的資訊。接著會介紹其中的一些。

正規表達式（regex）是很適合用來分析原文，和建構規則式系統的工具。regex 是一組字元或模式，其用途是比對及找出原文中的次級字串。例如，‘^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})$’可用來找出一段原文內的所有 email ID。regex 也很適合用來將領域知識整合至 NLP 系統，例如，假設我們透過聊天室或 email 接收顧客投訴，想要建構系統來自動識別他們投訴的產品，我們有一系列的產品代號對映至特定的品牌名稱，此時可以用 regex 來輕鬆地比對它們。

regex 是建構規則式系統的常見手段。StanfordCoreNLP 這類的 NLP 軟體具備 TokensRegex [10]，它是定義正規表達式的框架，可以識別原文中的模式，並使用匹配的原文來建立規則。regex 用於確定性（deterministic）比對——也就是說，它要嘛匹配，要嘛不匹配，機率性 regex 是它的分支，藉著加入匹配的機率來處理這種限制。感興趣的讀者可以研究 pregex [11] 之類的程式庫。

上下文無關文法（Context-free grammar，CFG）是一種形式文法（formal grammar），其用途是建構自然語言模型。CFG 是 Noam Chomsky 教授發明的，他是著名的語言學家和科學家。CFG 可用來描述比較複雜且階層式的資訊，它們可能是 regex 無法描述的。Earley 解析器 [12] 可以解析各種 CFG。JAPE（Java Annotation Patterns Engine）之類的語法語言可用來模擬更複雜的規則 [13]。JAPE 有 regex 以及 CFG 的功能，可在規則式 NLP 系統中使用，例如 GATE（General Architecture for Text Engineering）[14]。GATE 的用途是從封閉、定義良好、覆蓋範圍的準確性與完整性比較重要的領域中提取原文。例如，有人使用 JAPE 與 GATE 從臨床報告中提取心律調節器植入程序的資訊 [15]。圖 1-10 的 GATE 介面是一個規則式系統的案例，在圖中的原文中，有幾個突出顯示的資訊種類。

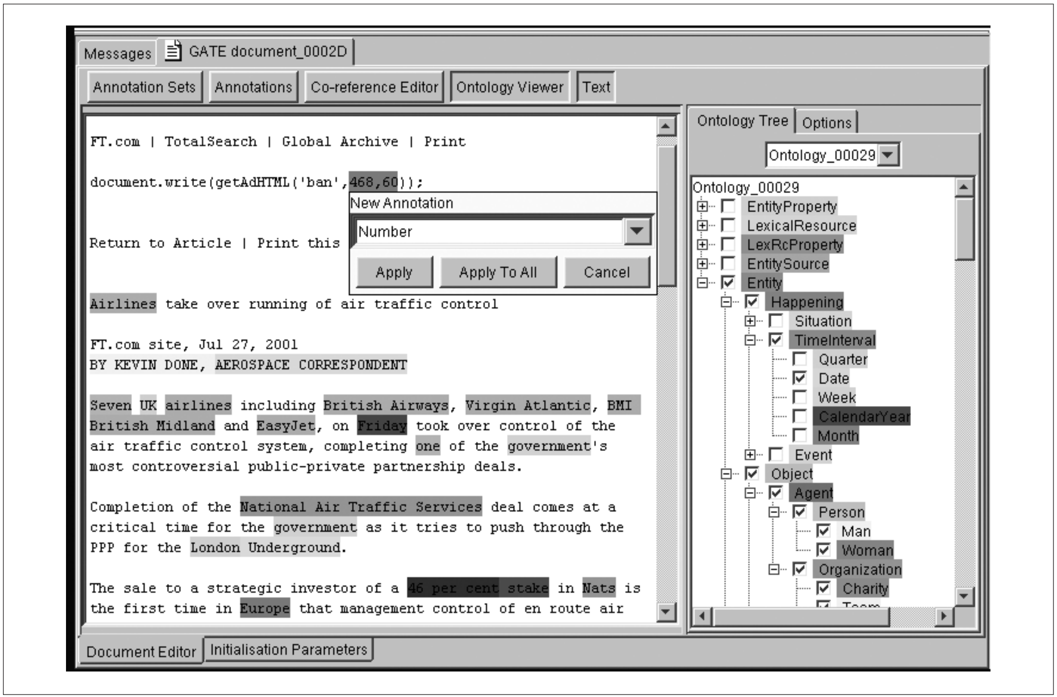

圖 1-10　GATE 工具

即使是現在，在 NLP 專案的完整生命周期中，規則與經驗法則仍然扮演重要的角色。一方面，它們是建構第一版的 NLP 系統的好方法。簡單地說，規則和經驗法則可幫助你快速建構模型的第一個版本，進而更理解眼前的問題。我們將在第 4 章與第 11 章深入討論這個部分。規則與經驗法則在機器學習式 NLP 系統也很實用。在專案生命周期頻譜的另一端，規則與經驗法則被用來填補系統中的空白。任何一種運用統計學、機器學習或深度學習技術來建構的 NLP 系統都會出錯，有些錯誤的代價很高，例如，醫療保健系統在察看病人的醫療紀錄之後，錯誤地不建議進行一項關鍵的檢查，這個錯誤甚至可能要人命。規則與經驗法則很適合在生產系統中填補這種空白。接下來，我們把焦點轉到 NLP 的機器學習技術。