Toolformer :語言模型可以自學使用工具

Timo Schick Jane Dwivedi-Yu Roberto Dessì † Roberta Raileanu
Maria Lomeli Luke Zettlemoyer Nicola Cancedda Thomas Scialom
元人工智能研究 † Universitat Pompeu Fabra

抽象的

語言模型 (LM) 表現出非凡的能力 ,可以僅通過幾個示例或文本指令來解決新任務 ,尤其是在規模上 。矛盾的是 ,它們還在基本功能上苦苦掙扎 ,例如算術或事實查找 ,而在這些功能中 ,更簡單 、更小的模型更勝一籌 。在本文中 ,我們展示了 LM 可以自學使用外部工具

通過簡單的 API 實現兩全其美 。我們介紹了 Toolformer ,這是一個經過訓練的模型 ,可以決定調用哪些 API 、何時調用它們 、傳遞哪些參數 ,以及如何最好地將結果納入未來的代幣預測 。這是以自我監督的方式完成的 ,只需要對每個 API 進行少量演示 。我們整合了一系列工具 ,包括計算器 、問答系統 、搜索引擎 、翻譯系統和日曆 。 Toolformer 在各種下游任務中實現了顯著改進的零樣本性能 ,通常可以與更大的模型競爭 ,而不會犧牲其核心語言建模能力 。

The New England Journal of Medicine 是[QA(“Who is the publisher of The New England Journal of Medicine?”) → Massachusetts Medical Society] MMS的註冊商標。

在 1400 名參與者中 ,有 400 名（或[計算器(400 / 1400) → 0.29] 29%)通過了測試。

這個名字來源於“la tortuga” ,西班牙語是[MT(“tortuga”) → turtle]烏龜的意思。

布朗法案是加利福尼亞州的法律[WikiSearch(“布朗法案”) → 拉爾夫·布朗法案是加利福尼亞州立法機關的一項法案 ,該法案保障公眾出席和參加當地立法機構會議的權利。] ,要求立法機構 ,像市議會一樣 ,向公眾開放他們的會議。

圖 1 :Toolformer 的示例性預測 。該模型自主決定調用不同的 API（從上到下 :問答系統 、計算器 、機器翻譯系統和維基百科搜索引擎)以獲取對完成一段文本有用的信息。

1 簡介

大型語言模型在各種自然語言處理任務上取得了令人印象深刻的零樣本和少樣本結果（Brown 等人， 2020 年；Chowd hery 等人， 2022 年， ia)並展示了幾種新興能力（Wei 等人， 2022 年）).然而 ,所有這些模型都有一些固有的局限性

最多只能通過進一步縮放來部分解決 。這些局限性包括無法獲取有關近期事件的最新信息（Komeili等人， 2022 年)和產生幻覺的相關傾向（Maynez 等人， 2020 年； Ji 等人，2022年） ,難以理解理解低資源語言（Lin et al., 2021） ,缺乏執行精確計算的數學技能（Patel et al., 2021)以及不了解時間的進程（Dhingra et al., 2022） 。

克服當今語言模型的這些限制的一個簡單方法是讓它們能夠使用外部工具 ,例如搜索引擎 、計算器或日曆 。然而 ,現有的方法要么依賴於大量的人工註釋（Komeili 等人， 2022 年； Thoppilan 等人， 2022 年） ,要么將工具的使用限制在特定任務的設置中（例如， Gao 等人， 2022 年； Parisi 等人） al., 2022) ,阻礙了在 LM 中更廣泛地採用工具使用。

因此 ,我們提出了 Toolformer ,一種學習以新穎方式使用工具的模型 ,它滿足以下需求 :

·應該以自我監督的方式學習工具的使用 ,而不需要大量的人工註釋 。這是重要的

圖 2 :我們方法的關鍵步驟 ,以問答工具為例 :給定輸入文本 x ,我們首先ck i 。然後我們執行這些 API 調用和採樣位置 i 和相應的 API 調用候選 c 過濾掉所有不會減少下一個令牌損失Li的調 我, c2我，。..,
用 。所有剩餘的 API 調用與原始文本交錯 ,產生新文本 x

這不僅是因為與此類註釋相關的成本 ,還因為人類認為有用的內容可能與模型認為有用的內容不同。

· LM 不應失去任何通用性 ,並且應該能夠自行決定何時

以及如何使用哪個工具 。與現有方法相比 ,這可以更全面地使用與特定任務無關的工具。

我們實現這些目標的方法基於最近的想法 ,即使用具有上下文學習的大型 LM （Brown 等人， 2020 年)從頭開始生成整個數據集（Schick 和 Schütze， 2021b； Honovich 等人， 2022 年； Wang 等人） al., 2022) :給出了一些關於如何使用 API 的人工編寫示例 ,我們讓 LM用潛在的 API 調用來註釋一個巨大的語言建模數據集 。然後 ,我們使用自監督損失來確定這些 API 調用中的哪些真正幫助模型預測未來的標記 。最後 ,我們在 LM 認為有用的 API 調用上微調 LM 本身 。如圖 1 所示 ,通過這種簡單的方法 ,LM 可以學習控制各種工具 ,並自行選擇何時使用哪種工具以及如何使用。

由於我們的方法與所使用的數據集無關 ,因此我們可以將其應用於最初用於預訓練模型的完全相同的數據集。

這確保了模型不會失去任何通用性和語言建模能力。

我們對各種不同的下游任務進行了實驗 ,證明在學習使用工具後 ,基於預訓練 GPT-J 模型（Wang 和 Komat suzaki， 2021)且具有 6.7B 參數的 Toolformer 實現了更強的零-shot 結果 ,明顯優於更大的 GPT-3 模型（Brown 等人， 2020)和各種任務的其他幾個基線。

2 方法

我們的目標是讓語言模型M具備通過 API 調用使用不同工具的能力。

我們要求每個 API 的輸入和輸出都可以表示為文本序列 。這允許將API 調用無縫插入到任何給定的文本中 ,使用特殊標記來標記每個此類調用的開始和結束。

我們將每個 API 調用表示為元組c = (ac, ic) ,其中ac是 API 的名稱， ic是對應的輸入 。給定具有相應結果r 的API 調用c ,我們將不包括和包括其結果的 API 調用的線性化序列分別表示為：

e(c) = <API> ac(ic) </API> e(c, r) =
<API> ac(ic) → r </API>

其中“<API>” 、 “</API>”和“→”是特殊標記 。1插入文本序列的線性化 API 調用的一些示例如圖1所示。 x |C|}純文本 ,我們首先轉換將這個數據集轉換為一個數據集C * ,並通過 API 調用進行

給定數據集C = {x ,. . . , 擴充 。這分三個步驟完成 ,如圖 2 所示 :首先 ,我們利用M的上下文學習能力對大量潛在的API 調用進行採樣 ,然後我們執行這些 API 調用 ,最後檢查獲得的響應是否有助於預測未來的代幣 ,這被用作過濾標準 。過濾後 ,我們合併對不同工具的 API 調用 ,得到擴充數據集C *

和微調

在實踐中 ,我們使用標記序列“ [” 、 “]”和“->”分別表示“<API>” 、 “</API>”和“→” 。這使我們的方法能夠在不修改現有 LM 詞彙表的情況下工作 。出於可讀性的原因 ,我們在本節中仍然將它們稱為“<API>” 、 “</API>”和“→” 。

您的任務是將對問答 API 的調用添加到一段文本中。這些問題應該可以幫助您獲得完成文本所需的信息。您可以通過編寫“[QA(question)]”來調用 API，其中“question”是您要問的問題。以下是 API 調用的一些示例：

輸入：喬·拜登出生於賓夕法尼亞州的斯克蘭頓。

輸出：Joe Biden 出生於[QA(Where was Joe Biden born?)] Scranton, [QA(In which state is Scranton?)] Pennsylvania。

輸入：可口可樂，或可口可樂，是可口可樂公司生產的一種碳酸軟飲料。

輸出：可口可樂，或[QA（“可口可樂的其他名稱是什麼？”）]可口可樂，是由[QA（“誰生產可口可樂？”）]可口可樂製造的碳酸軟飲料公司。

輸入：x

輸出：

圖 3：用於為問答工具生成 API 調用的示例性提示 P(x)。

M本身在這個數據集上。下面更詳細地描述了這些步驟中的每一個。

採樣 API 調用對於每個 API，我們編寫一個提示P(x)，鼓勵 LM 註釋一個示例x = x1, 。 。 。, xn與 API 調用。

圖 3 顯示了這樣一個問題提示的例子，一個回答工具；所有使用的提示都顯示在附錄A.2 中。設pM(zn+1 | z1, . . . , zn)是M分配給標記zn+1作為序列 zn 的延續的概率。我們首先採樣最多k 個候選位置z1, . . . ,通過計算進行 API 調用的 tions，對於每個i ∈ {1, . . . , n}, 概率

$$p_i = p_M(\text{<API>} \mid P(x), x_{1:i-1})$$

M分配給在位置 i 開始 API 調用。給定採樣閾值τs，我們保留所有位置I = {i | pi > τs};如果有超過k 個這樣的位置，我們只保留前 k 個。

對於每個位置i ∈ I，我們然後通過從給定i , . . . , i序列[P(x), x1, . . . , xi−1, <API>]作為前綴和 </API> 作為序列結束標記。 2

2我們丟棄所有M不生成 </API> 令牌。

執行 API 調用下一步，我們執行M生成的所有 API 調用以獲得相應的結果。如何做到這一點完全取決於API 本身 例如，它可能涉及調用另一個神經網絡、執行 Python腳本或使用檢索系統對大型語料庫執行搜索。每個 API 調用ci的響應需要是單個文本序列ri 。

過濾 API 調用 令i為API 調用ci在序列x = x1,中的位置。 . . , xn,令 ri為來自 API 的響應。此外，給定一個權重序列 (wi | i ∈ N)，令

$$\text{李}(z) = -\sum_{j=i}^{n} w_{j-i} \cdot \log p_M(x_j \mid z, x_{1:j-1})$$

是M在標記xi上的加權交叉熵損失 ,. . . , xn如果模型以z 為前綴。我們比較了這種損失的兩個不同實例：

$$\text{鋰}^{+} = \text{Li}(e(c_i, r_i)) =$$

$$\text{大號-} \min(\text{Li}(\varepsilon), \text{Li}(e(c_i, \varepsilon)))$$

其中ε表示一個空序列。前者是所有代幣xi的加權損失 ,. . . , xn如果 API 調用及其結果作為前綴提供給M ；3 後者是從 (i) 根本不進行 API 調用和 (ii) 進行API 調用但不提供響應中獲得的損失中的最小值。直觀地說，與根本不接收 API 調用或僅接收其輸入相比，如果向M提供此調用的輸入和輸出，則 API 調用對 M 有幫助，這使得模型更容易預測未來的標記。給定一個過濾閾值τf ，我們因此只保留 API 調用

$$\text{大號}_i - L^{+} \geq \tau_f$$

成立，即，與不進行任何API 調用或未從中獲得任何結果相比，添加 API 調用及其結果至少減少了τf的損失。

模型微調在對所有 API 的調用進行採樣和過濾之後，我們最終合併剩餘的API 調用並將它們與原始輸入交織在一起。也就是說，對於輸入文本x = x1, . . . ,在位置i 處調用相應的 API 和結果(ci , ri) , 我們構造新的序列x xn

$$^{*} =$$

3我們提供e(ci, ri)作為前綴，而不是將其插入位置i ，因為M尚未在包含 API 調用的任何示例上進行微調，因此將其插入 x 的中間會中斷流程並且不與模式中的模式對齊預訓練語料庫，從而傷害困惑。

x1:i−1, e(ci , ri), xi:n;我們對具有多個 API 調用的文本進行類似處理。對所有x ∈ C執行此操作會導致新數據集 C *增加了 API 調用。我們使用這個新數據集來微調M，使其暴露於與在C上微調相同的內容。

此外，由於 API 調用恰好插入到那些位置，並且恰好使用那些幫助M預測未來標記的輸入，對C * 的微調使語言模型能夠完全根據自己的反饋來決定何時以及如何使用。

推論在使用我們的方法進行微調後使用M生成文本時，我們會執行常規解碼，直到M生成“→”標記，這表明它接下來需要 API 調用的響應。

此時，我們中斷解碼過程，調用適當的 API 以獲得響應，並在插入響應和 </API> 標記後繼續解碼過程。

3 工具

我們探索了各種工具來解決常規 LM 的不同缺點。我們對這些工具施加的唯一限制是 (i) 它們的輸入和輸出都可以表示為文本序列，以及 (ii) 我們可以獲得一些關於它們預期用途的演示。具體來說，我們探索了以下五個工具：問答系統、維基百科搜索引擎、計算器、日曆和機器翻譯系統。與這些工具相關的 API 的一些潛在調用和返回字符串的示例顯示在

表1. 我們在下面簡要討論所有工具；進一步的細節可以在附錄A中找到。

問答我們的第一個工具是基於另一個 LM 的問答系統，它可以回答簡單的事實性問題。具體來說，我們使用Atlas （Izacard 等人， 2022 年），這是一種針對自然問題進行微調的檢索增強LM （Kwiatkowski 等人， 2019 年）。

計算器作為第二個工具，我們使用一個可以進行簡單數值計算的計算器；我們只支持四種基本算術運算。

結果始終四捨五入到小數點後兩位。

維基百科搜索我們的第三個工具是搜索引擎，給定搜索詞，返回短文本摘自維基百科。與我們的問答工具相比，這種搜索使模型能夠獲得關於某個主題的更全面的信息，但需要它自己提取相關部分。作為我們的搜索引擎，我們使用 BM25 檢索器（ Robertson 等人， 1995 年；Baeza-Yates 等人， 1999 年)為來自 KILT 的維基百科轉儲編制索引（Petroni 等人， 2021 年）。

機器翻譯系統我們的第四個工具是基於 LM 的機器翻譯系統，它可以將短語從任何語言翻譯成英語。更具體地說，我們使用 600M 參數 NLLB (Costa-jussà et al., 2022)作為適用於 200 局域網的多語言機器翻譯模型

量表（包括低資源量表）。使用快速文本分類器(Joulin et al., 2016)自動檢測源語言，而目標語言始終設置為英語。

日曆我們的最後一個工具是日曆 API，當被查詢時，無需任何輸入即可返回當前日期。這為需要一定時間意識的預測提供了時間上下文。

4 實驗

我們調查我們的方法是否使模型能夠在沒有任何進一步監督的情況下使用工具，並自行決定何時以及如何調用哪些可用工具。為了測試這一點，我們選擇了各種下游任務，在這些任務中我們假設至少有一個所考慮的工具有用，並評估零樣本設置中的性能（第 4.2 節）。

除此之外，我們還確保我們的方法不會損害模型的核心語言建模能力；我們通過查看兩個語言建模數據集的困惑度來驗證這一點（第4.3 節）。最後，我們調查了使用工具學習的能力如何受模型大小的影響（第 4.4 節）。

4.1 實驗裝置

數據集生成在我們所有的實驗中，我們使用 CCNet （Wenzek 等人， 2020)的一個子集作為我們的語言建模數據集C和 GPT J （Wang 和 Komatsuzaki， 2021)作為我們的語言模型M。為了降低計算成本為了使用 API 調用註釋C，我們為某些 API 定義啟發式方法以獲得C的一個子集，對於這些 API 調用比對普通文本更有幫助。例如，我們只考慮包含至少三個數字的計算器工具的文本。所用啟發式的詳細信息在

接口名稱	示例輸入	示例輸出
問答	哥倫布騎士團是在哪裡成立的？	康涅狄格州紐黑文
維基百科搜索	漁線輪類型	旋轉釣魚 > 旋轉釣魚根據使用的魚竿和卷線器的類型區分飛釣和拋餌釣魚。旋轉釣魚時使用兩種類型的捲線器，開放式捲線器和封閉式捲線器。
計算器	27 + 4 * 2	35
日曆	ε	今天是 2023 年 1 月 30 日，星期一。
機器翻譯	sûreté nucléaire	核安全

表 1：使用的所有 API 的輸入和輸出示例。

應用程序接口	示例數τf = 0.5	τf = 1.0	τf = 2.0
問答	51,987	18,526	5,135
維基百科搜索	207,241	60,974	13,944
計算器	3,680	994	138
日曆	61,811	20,587	3,007
機器翻譯	3,156	1,034	229

表 2：對於我們的過濾閾值τf的不同值， C *中 API 調用的示例數量。

附錄A. 為了從 C 獲得C * ，我們執行第2節中描述的所有步驟，並額外過濾掉所有在過濾步驟中消除了所有 API 調用的示例。4 對於權重函數，我們使用

$$\text{重量} = \frac{w_t}{\sum_{s \in N} w_s} \quad \text{其中 } w_t = \max(0, 1 - 0.2 \cdot t)$$

以確保 API 調用發生在API 提供的信息對模型實際有幫助的位置附近。為每個工具單獨選擇閾值τs和τf以確保足夠多的示例；詳見附錄A。表2顯示了我們的相關統計數據

使用 API 調用擴充的最終數據集。

模型微調我們使用128 的批量大小和1 · 10−5的學習率對C *上的M進行微調，並對前 10% 的訓練進行線性預熱。
我們的微調程序的詳細信息在附錄B中給出。

基線模型在本節的其餘部分，我們主要比較以下模型：

- GPT-J：沒有任何微調的常規GPT-J 模型。
- GPT-J + CC： GPT-J 在C上微調，我們的 CCNet 子集沒有任何 API 調用。
- Toolformer：在C上微調的 GPT-J *，我們的潛艇一組通過 API 調用增強的 CCNet。
- Toolformer（禁用）：與 Toolformer 相同的模型，但 API 調用在解碼期間被禁用。5

對於大多數任務，我們還與 OPT (66B) (Zhang et al., 2022)和 GPT-36 (175B)進行了比較 (Brown et al., 2020)，這兩個模型分別比我們的其他基線模型大 10 倍和 25 倍。

4.2 下游任務

我們評估各種下游任務的所有模型。在所有情況下，我們都考慮提示零樣本設置 即指示模型以自然語言解決每個任務，但我們不提供任何上下文示例。這與之前關於工具使用的工作（例如， Gao 等人， 2022 年； Parisi 等人， 2022 年)形成對比，在這些工作中，模型提供了特定於數據集的示例，說明如何使用工具來解決具體任務。我們選擇更具挑戰性的零樣本設置，因為我們有興趣了解 Toolformer 是否恰好適用於用戶未事先指定應以何種方式使用哪些工具來解決特定問題的情況。

我們使用標準的貪婪解碼，但對 Toolformer 進行了一次修改：我們讓模型不僅在<API>最有可能時開始 API 調用

4雖然這種過濾改變了訓練示例的分佈，但我們假設其餘示例足夠接近原始分佈，因此M 的語言建模能力不受影響。這一假設在第 4.3 節中得到了經驗驗證。

5這是通過手動設置概率來實現的<API> 標記為 0。

6我們使用未在任何指令上微調的原始davinci變體。

標記 ,但只要它是k個最有可能的標記之一 。對於k = 1 ,這對應於常規的貪心解碼 ;我們改為使用k = 10來增加模型的配置 ,以使用它可以訪問的 API 。同時 ,我們每個輸入最多只調用一個 API ,以確保模型不會陷入循環 ,在循環中不斷調用 API 而不產生任何實際輸出 。第 5 節探討了這些修改的效果。

4.2.1喇嘛

我們在 LAMA 基準（Petroni 等人， 2019 年)的 SQuAD、Google RE 和 T-REx 子集上評估我們的模型 。對於這些子集中的每一個 ,任務是完成一個帶有缺失事實（例如 ,日期或地點)的簡短陳述 。由於 LAMA 最初設計用於評估掩碼語言模型（例如，Devlin 等人， 2019） ,我們過濾掉掩碼標記不是最終標記的示例 ,以便剩餘的示例可以以從左到 - 的方式處理正確的時尚 。為了考慮到不同的標記化和不形成需要單個詞的模型而增加的複雜性，我們使用比精確匹配稍微寬鬆的評估標準 ,並簡單地檢查正確的詞是否在模型預測的前五個詞中 。由於 LAMA 基於直接從維基百科獲得的陳述 ,我們阻止 Tool former 使用維基百科搜索 API 以避免給予它不公平的優勢。

所有模型的結果見表3。
所有不使用工具的 GPT-J 模型實現相似
表現 。至關重要的是 ,Toolformer 明顯優於這些基線模型 ,分別比最佳基線提高了 11.7 、5.2 和 18.6 點 。它也明顯優於 OPT (66B) 和GPT-3 (175B) ,儘管這兩個模型都大得多 。這是因為模型在幾乎所有情況下(98.1%)都獨立決定向問答工具詢問所需信息 ;僅在極少數示例中 ,它使用不同的工具 (0.7%) 或根本不使用任何工具 (1.2%)。

4.2.2 數學數據集

我們在 ASDiv （Miao 等人， 2020 年） 、 SVAMP （Patel 等人， 2021 年)和MAWPS 基準（Koncel-Kedziorski 等人， 2016 年)上測試數學推理能力 。我們再次考慮到我們通過使用更寬鬆的評估標准在零樣本設置中測試所有模型的事實 :由於所需的輸出始終是一個數字 ,我們只需檢查第一個

模型	SQuAD	Google-RE	T-REx
GPT-J	17.8	4.9	31.9
GPT-J+CC	19.2	5.6	33.2
工具成型機（禁用）	22.1	6.3	34.9
工具成型機	33.8	11.5	53.5
選擇 (66B)	21.6	2.9	30.1
GPT-3 (175B)	26.8	7.0	39.8

表 3 :LAMA 子集的結果。 Toolformer 對大多數示例使用問答工具 ,明顯優於所有相同大小的基線 ,並取得與 GPT-3 (175B) 競爭的結果。

模型	ASDiv	SVAMP	MAWPS
GPT-J	7.5	5.2	9.9
GPT-J+CC	9.6	5.0	9.3
工具成型機（禁用）	14.8	6.3	15.0
工具成型機	40.4	29.4	44.0
選擇 (66B)	6.0	4.9	7.9
GPT-3 (175B)	14.0	10.0	19.8

表 4 :需要數學推理的各種基準測試的結果。 Toolformer 在大多數示例中使用了計算器工具 ,甚至明顯優於 OPT (66B) 和 GPT-3 (175B)。

模型預測的數字.7
表4顯示了所有基準測試的結果 。雖然GPT-J 和 GPT-J + CC 的性能大致相同 ,但即使禁用 API 調用 ,Toolformer 也能獲得更強的結果 。我們推測這是
因為該模型在許多 API 調用示例及其結果上進行了微調 ,從而提高了其自身的數學能力 。儘管如此 ,允許模型進行 API 調用可以使所有任務的性能提高一倍以上 ,並且明顯優於更大的 OPT 和 GPT-3 模型 。這是因為在所有基準測試中 ,對於所有示例中的 97.9% ,模型決定向計算器工具尋求幫助。

4.2.3 問答

我們查看 Web 問題（Berant 等人， 2013 年） 、自然問題（Kwiatkowski 等人， 2019 年)和TriviaQA （Joshi 等人， 2017 年） ,這三個問題是Brown 等人考慮的回答數據集。 （2020） 。
為了進行評估 ,我們檢查模型預測的前 20 個單詞是否包含正確答案 ,而不是要求完全匹配 。對於 Tool former ,我們禁用問答工具作為

7 一個例外情況是 ,如果模型的預測包含一個等式（例如，“正確答案是 5+3=8”） ,在這種情況下 ,我們將“=”符號後的第一個數字視為其預測。

模型	WebQS	NQ	TriviaQA
GPT-J	18.5	12.8	43.9
GPT-J+CC	18.4	12.2	45.6
工具成型機（禁用）	18.9	12.6	46.7
工具成型機	26.3	17.7	48.8
選擇 (66B)	18.6	11.4	45.7
GPT-3 (175B)	29.0	22.6	65.9

表 5 :各種問答數據集的結果。
對於大多數示例使用維基百科搜索工具 ,Toolformer 明顯優於相同大小的基線 ,但不及 GPT-3 (175B)。

這將使解決任務變得微不足道 ,特別是考慮到底層 QA 系統已針對自然問題進行了微調。

結果如表 5 所示 。Toolformer 再次明顯優於所有其他基於 GPT-J 的模型 ,這次主要依靠維基百科搜索 API (99.3%) 來查找相關信息 。然而 ,Toolformer 仍然落後於更大的 GPT-3 (175B) 模型 。這可能是由於我們搜索引擎的簡單性（在許多情況下 ,它返回的結果顯然不是給定查詢的良好匹配)和 Toolformer 無法與其交互 ,例如 ,通過重新構建其查詢 ,如果結果沒有幫助或通過瀏覽多個頂級結果 。我們相信添加此功能是未來工作的一個令人興奮的方向。

4.2.4 多語言問答

我們在 MLQA （Lewis 等人， 2019 年）（一種多語言問答基準)上評估 Toolformer 和所有基線模型 。每個問題的上下文段落都以英文提供 ,而問題可以是阿拉伯文 、德文 、西班牙文 、印地文 、越南文或簡體中文 。為了解決任務 ,模型需要能夠理解段落和問題 ,因此將問題翻譯成英文可能會有所幫助 。我們的評估指標是模型的生成次數（上限為 10 個單詞)包含正確答案的百分比。

結果如表 6 所示 。使用 API 調用持續提高 Toolformer 對所有語言的性能 ,表明它已經學會使用機器翻譯工具 。根據語言的不同 ,該工具用於所有示例的 63.8% 到 94.9%；唯一的例外是印地語 ,只有 7.3% 的情況使用機器翻譯工具 。然而 ,工具

模型	Es	De	Hi	Vi	Zh	Ar
GPT-J	15.2	16.5	1.3	8.2	18.2	8.2
GPT-J + CC	15.7	14.9	0.5	8.3	13.7	4.6
Toolformer（禁用）	19.8	11.9	1.4	2.1	0.6	1.7
Toolformer	20.6	13.5	6.3	11.3	16.0	3.1
選擇 (66B)	0.3	0.1	1.1	0.2	0.7	0.1
GPT-3 (175B)	0.1					
GPT-J（全英文）	24.3	27.0	23.9	23.3	23.1	23.6
GPT-3（全英文）	26.1	24.9	23.6	24.0		

模型	3.4	1.1	0.1	1.7	17.7
Toolformer	24.7	27.2			

表 6 :西班牙語 (Es) 、德語 (De) 、印地語 (Hi) 、越南語 (Vi) 、中文 (Zh) 和阿拉伯語 (Ar) 的 MLQA 結果 。雖然使用機器翻譯工具翻譯問題對所有語言都有幫助 ,但在 CCNet 上進一步預訓練會降低性能 ;因此 ,Toolformer 並不總是優於 GPT-J 。最後兩行對應於給定上下文和英語問題的模型。

前者並不總是優於香草GPT-J 。這主要是因為對於某些語言， CCNet 上的微調會降低性能 ;這可能是由於與GPT-J 的原始預訓練數據相比分佈發生了變化。

OPT 和 GPT-3 在所有語言中的表現都出奇地弱 ,這主要是因為它們未能用英語提供答案 ,儘管按照規定這樣做。 GPT-J 沒有遇到這個問題的一個潛在原因是它接受了比 OPT 和 GPT-3 更多的多語言數據訓練 ,包括 EuroParl 語料庫（Koehn， 2005 年； Gao等人， 2020 年） 。作為上限 ,我們還在MLQA 的變體上評估 GPT-J 和 GPT-3 ,其中上下文和問題均以英語提供 。在此設置中 ,GPT-3 的性能優於所有其他模型 ,支持我們的假設 ,即它在 MLQA 上的性能不佳是由於任務的多語言方面。

4.2.5 時間數據集

為了研究日曆 API 的實用程序 ,我們評估了TEMPLAMA上的所有模型（Dhingra 等人， 2022 年)和一個我們稱為 DATESET 的新數據集。
TEMPLAMA 是一個從 Wikidata 構建的數據集 ,包含關於隨時間變化的事實的完形填空查詢（例如 ,“Cristiano Ronaldo plays for ___”)以及 2010 年和 2020 年之間的正確答案 。DATESET ,在附錄D中描述 ,也是通過一系列模板生成的 ,但使用隨機日期/持續時間的組合填充（例如 ,“ 30 天前是星期幾？”） 。至關重要的是 ,需要知道當前日期才能回答這些問題。

模型	TEMPLAMA數據集	
GPT-J	13.7	3.9
GPT-J+CC	12.9	2.9
工具成型機（禁用）	12.7	5.9
工具成型機	16.3	27.3
選擇 (66B)	14.5	1.3
GPT-3 (175B)	15.5	0.8

表 7 ：時間數據集的結果。 Toolformer 優於所有基線，但不使用TEMPLAMA 的日曆工具。

對於這兩項任務，我們使用與原始 LAMA 數據集相同的評估。

表7中顯示的結果表明，工具模型優於TEM PLAMA 和 DATESET 的所有基線。然而，更仔細的檢查表明，TEMPLAMA的改進不能歸功於日曆工具，日曆工具僅用於所有示例的 0.2%，而主要歸功於Toolformer 調用最多的維基百科搜索和問答工具。這是有道理的，因為TEMPLAMA中的命名實體通常非常具體和罕見，即使只知道確切的日期也無濟於事。該數據集的最佳操作過程　首先查詢 calen dar API 以獲取當前日期，然後使用該日期查詢問答系統　不僅被我們限制每個示例最多使用一個 API 調用所禁止，而且鑑於其訓練數據中的所有 API 調用都是獨立採樣的，因此Toolformer也很難學習。

另一方面，對於DATESET，與其他模型相比，Toolformer 的顯著改進可以完全歸功於日曆工具，它使用了所有示例的 54.8%。

4.3 語言建模

除了驗證各種下游任務的改進性能外，我們還希望通過 API 調用的微調確保Toolformer 的語言建模性能不會降低。為此，我們在兩個語言建模數據集上評估了我們的模型：WikiText （Merity 等人， 2017 年)和來自 CCNet （Wenzek 等人， 2020 年)的 10,000 個隨機選擇的文檔子集，這些文檔在訓練期間未使用。表 8 顯示了各種模型的困惑。正如人們所預料的那樣，在 CCNet 上進行微調會導致在不同的 CC Net 子集上略微提高性能，但會略微降低 WikiText 上的性能，大概是因為原始預

模型	維基文本	CCNet
GPT-J	9.9	10.6
GPT-J+CC	10.3	10.5
工具成型機（禁用）	10.3	10.5

表 8 ：WikiText 和我們的 CCNet 驗證子集上不同模型的困惑。在沒有任何 API 調用的情況下，添加 API 調用不會帶來語言建模的困惑。

GPT-J 的訓練數據比我們隨機選擇的 CCNet 子集更類似於維基文本。

然而，最重要的是，在推理時禁用 API 調用時，與C上的訓練相比， C * （我們用 API 調用註釋的數據集)訓練不會導致困惑度增加。 8

4.4 比例定律

我們調查了當我們改變 LM 的大小時，尋求外部工具幫助的能力如何影響性能。為此，我們不僅將我們的方法應用於 GPT-J，還應用於 GPT-2 家族的四個較小模型（Radford 等人， 2019），參數分別為 124M、355M、775M 和 1.6B .我們只使用三種工具的一個子集來做到這一點：問答系統、計算器和維基百科搜索引擎。除此之外，我們遵循第 4.1節中描述的實驗設置。

圖4顯示，利用提供的工具的能力僅在大約 775M 參數時出現：較小的模型在使用和不使用工具的情況下實現相似的性能。一個例外是主要用於 QA 基準的維基百科搜索引擎；我們假設這是因為 API 相對易於使用。隨著規模的增長，模型在無需 API 調用的情況下解決任務的能力會越來越好，同時它們充分利用提供的 API 的能力也會提高。

因此，即使對於我們最大的模型，使用和不使用 API 調用的預測之間仍然存在很大差距。

5 分析

解碼策略我們研究了第 4.2 節中介紹的修改後的解碼策略的效果，而不是總是生成

8我們不評估啟用 API 調用的 Toolformer 的困惑度，因為計算給定x1的令牌xt的概率pM(xt | x1, . . . , xt－1) , . . . , xt－1將需要邊緣化模型在位置 t 可能進行的所有潛在 API 調用，這是棘手的。

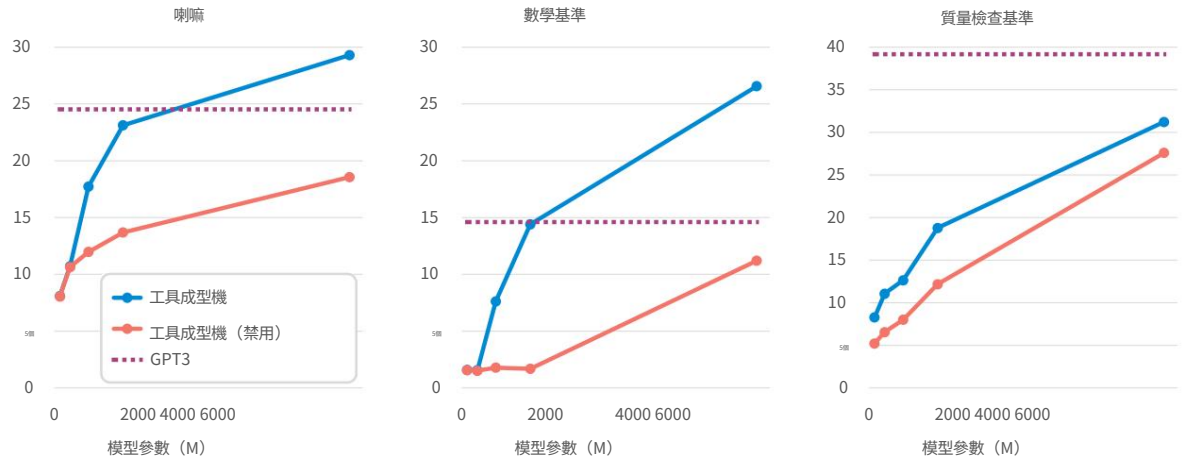

圖 4 :LAMA 的平均性能 ,我們的數學基準和我們針對不同大小的 GPT-2 模型的 QA 基準以及使用我們的方法微調的 GPT-J ,無論是否有 API 調用 。雖然 API 調用對最小的模型沒有幫助 ,但較大的模型學習如何充分利用它們 。即使對於更大的模型 ,使用和不使用 API 調用的模型預測之間的差距仍然很大。

最有可能的標記 ,如果它是k個最有可能的標記之一 ,我們將生成<API>標記 。表9顯示了不同 k 值在 LAMA 的 T-REx 子集和 WebQS 上的性能 。正如預期的那樣 ,增加k會導致模型對更多示例進行 API調用 從k = 1 （即常規貪婪解碼)的 40.3% 和 8.5%到k = 10的 98.1%和 100% 。而對於 T-REx ,貪婪解碼的性能已經有了明顯的提高 ,在 WebQS 上 ,我們的模型才開始進行大量的 API 調用 ,因為我們

	霸王龍						WebQS					
k	所有	AC	NC	%	所有	AC	NC	%				
0	34.9	–	34.9	0.0	1	47.8	53.0	44.3	18.9	–	18.9	0.0
10	98.1	98.1	22.5	100.0	10	98.1	98.1	26.4	19.3	17.1	19.9	8.5

表 9 :對於解碼期間使用的不同 k 值 ,Toolformer 在 LAMA 的 T-REx 子集和 WebQS 上的結果 。顯示的數字是整體性能 (All) 、模型決定調用 API 的子集 (AC) 和所有剩餘示例 (NC) 的性能 ,以及模型決定調用 API 的示例的百分比原料藥 (%)。

稍微增加k 。有趣的是 ,對於k = 1 ,模型在某種程度上進行了校準 :它決定

調用 API 以獲取在不進行 API 調用的情況下會表現特別糟糕的示例 。這可以從以下事實看出 :決定不進行 API 調用的示例（44.3 和19.9)的性能高於如果根本不進行 API 調用（34.9 和 18.9)的平均性能。然而 ,對於較高的 k 值 ,此校準會丟失。

第四個例子中的 “Fast train success”沒有給出任何相關信息 ,但仍然減少了困惑 。然而 ,未過濾的 API 調用中的一些噪聲實際上是有用的 ,因為它迫使在C *上微調的模型不總是盲目地遵循它所做的每個調用的結果。

數據質量我們定性分析了一些使用我們的方法針對不同 API 生成的 API 調用 。表10顯示了一些來自 CCNet 的文本示例 ,這些文本增加了 API 調用 ,以及用作過濾標準的+相應分數L — — L ,以及模型進上下文中是否直觀有用. + 不提供任何對預測未來令牌有用的信息的 API 調用。

通常

有一些例外 ,例如 ,API 調用

6 相關工作

語言模型預訓練有多種方法可以在預訓練期間使用某種形式的附加文本信息來增強語言模型 ,包括各種形式的元數據（Keskar 等人， 2019 年） 、 HTML 標籤（Aghajanyan 等人， 2021 年） 、維基百科標記（Schick et al., 2022)，或從信息檢索系統中獲得的相關文本(Guu et al., 2020; Borgeaud et al., 2021; Izacard et al., 2022) 。對於所有這些

例子	大筆¡ — L +	有用
Flodden Window（一座獻給米德爾頓弓箭手的戰爭紀念碑)位於米德爾頓一級保護建築聖倫納德教堂 ,據說是英國最古老的戰爭紀念碑。 <API> WikiSearch(War memorial Flodden) → Battle of Flodden > Commemoration > The stained glass Flodden Window in Middleton Parish Church [. . .] 由 Richard Assheton 爵士建造 ,以紀念弗洛登戰役和參加戰鬥的米德爾頓弓箭手。 </API>米德爾頓的理查德·阿謝頓爵士（聖倫納德的建造者)被授予爵士頭銜 [. . .]	5.49	✓
注意 :WL 將於週五開放， <API> Calendar() →今天是 2017 年 3 月 9 日星期四。 </API> 3 月 10 日和 3 月 19 日星期日的正常營業時間。	2.11	✓
尼羅河的大概長度為<API> QA(What is the approximate length of the Nile?) → 6,853 km </API> 6,853 公里 ,白尼羅河是它的主要源頭。	2.08	✓
如果金星的大氣層與地球相似 ,那麼您會預計金星的平均溫度為499 K (1.74 x 287) 而不是 735 K ,即<API> Calculator(735 / 499) → 1.47 </API> 1.47 (735 / 499) 比應有的溫度高幾倍。	1.59	✓
你在這裡： 主頁 / 精選 / 趕上通往成功的快車！ <API> WikiSearch(Fast train success) → Fast Train > 它也在 1971 年 6 月 26 日在加拿大 CHUM 單曲榜上達到了第 23 位的頂峰 。這首單曲的成功確立了 Myles Goodwyn 作為樂隊的主要詞曲作者，並使其成為可能為 April Wine 錄製第二張專輯。 </API>不要等待數周和[. . .]	0.92	
Os Melhores Escolas em Jersey 2020 <API> MT(Os Melhores Escolas em Jersey) →最佳 澤西島的學校</API>在此頁面上 ,您可以搜索澤西島的大學 、學院和商學院	0.70	✓
欣賞<API> Calendar()中的這些圖片→今天是 2013 年 4 月 19 日 ,星期五。 </API> 尋找復活節彩蛋。	0.33	✓
85 名患者 (23%) 活著住院並住進了醫院病房 。其中， <API>計算器(85 / 23) → 3.70 </API> 65% 有心髒病因 [. . .]	—0.02	
但是 ,嘿 ,在<API> Calendar()之後→今天是 2011 年 6 月 25 日 ,星期六。 </API>迪斯尼樂園的消防演習慘敗 ,我認為可以肯定地說 Chewey 不會讓任何人死於火災。	—0.41	
我最後一次和<API> QA 在一起（我最後一次和誰在一起？） →最後一次</API> 他我問他喜歡我什麼 ,他說有一天他會告訴我。	—1.23	

+表 10 :不同工具的 API 調用示例 ,按 L — — L 標準的值排序 。高值通常對應於直觀地用於預測未來令牌的 API 調用 。 用作過濾

方法 ,總是提供額外的信息 ,不管它是否有幫助。相比之下 ,Toolformer 自己學習明確要求正確的信息。

工具使用有幾種方法旨在讓 LM能夠使用外部工具 ,例如搜索引擎（Komeili 等人， 2022 年； Thoppilan 等人， 2022 年； Lazaridou 等人， 2022 年； Shuster 等人， 2022 年； Yao 等人等人， 2022 年） 、網絡瀏覽器（Nakano 等人， 2021 年） 、計算器（Cobbe 等人， 2021 年； Thoppilan等人， 2022 年） 、翻譯系統（Thoppilan 等人， 2022 年)和 Python 解釋器（Gao 等人等人， 2022 年） 。

這些模型學習使用工具的方式大致可以分為兩種方法 :要么依賴大量的人工監督（Komeili 等人， 2022 年； Nakano 等人， 2021 年； Thoppilan 等人， 2022 年） ,要么他們工作通過在針對特定任務量身定制的少量設置中提示語言模型 ,其中先驗地知道哪些工具需要使用(Gao et al., 2022; Lazaridou et al., 2022; Yao et al., 2022) 。相比之下， Toolformer 的自我監督性質使其能夠學習如何以及何時

使用工具而不需要特定的提示來顯示如何使用工具的特定任務示例 。也許與我們的工作最密切相關的是 TALM （Parisi 等人， 2022 年） ,這是一種使用類似的自我監督目標來教模型使用計算器和搜索引擎的方法 ,但僅在模型針對下游任務進行了微調。

自舉使用自我訓練和自舉技術來改進模型的想法已經在各種情況下進行了研究 ,包括詞義消歧（Yarowsky， 1995） 、關係提取（Brin， 1999； Agichtein 和 Gravano， 2000） 、解析（McClosky等人， 2006 年； Reichart 和 Rappoport， 2007 年） ,序列生成（He 等人， 2020 年） ,少量文本分類

化(Schick and Schütze, 2021a)和檢索(Izacard and Grave, 2021)到推理(Zelikman et al., 2022) 。本著與這些方法類似的精神， Toolformer 在應用基於困惑度的過濾步驟後根據自己的預測進行訓練。

7 限制

雖然我們的方法使 LM 能夠以自我監督的方式學習如何使用各種工具，但對於可以實現的目標有一些明顯的限制
使用我們當前形式的方法 。一個這樣的限制是 Toolformer 無法使用鏈中的工具（即，使用一個工具的輸出作為另一個工具的輸入） 。這是因為每個工具的 API 調用都是獨立生成的；因此，微調數據集中沒有使用鏈接工具的示例 。我們目前的方法也不允許 LM 以交互方式使用工具 尤其是對於搜索引擎等工具，可能會返回數百個不同的結果，使 LM 能夠瀏覽這些結果或優化其搜索查詢本著與 Nakano 等人類似的精神。(2021)對於某些應用來說可能至關重要 。除此之外，我們發現使用 Toolformer 訓練的模型在決定是否調用 API時通常對輸入的確切措辭很敏感，這也許不足為奇，因為已知 LM 對在零和少鏡頭設置中提供的提示非常敏感（Jiang 等人， 2020 年； Schick 和 Schütze， 2021a） 。根據工具的不同，我們的方法的樣本效率也很低；例如，處理超過一百萬個文檔只會產生幾千個對計算器 API 的有用調用示例 。這個問題的一個潛在解決方案可能是迭代應用我們的方法，類似於相關引導方法中的做法（Schick 和 Schütze， 2021a； Izacard和 Grave， 2021； Parisi 等人， 2022） 。最後，在決定是否進行 API 調用時，Tool former 目前沒有考慮進行 API 調用所產生的依賴於工具的計算成本。

8 結論

我們引入了 Toolformer，這是一種語言模型，它以自我監督的方式學習如何通過簡單的 API 調用來使用不同的工具，例如搜索引擎 、計算器和翻譯系統 。這是通過對大量採樣的API 調用進行微調來完成的，這些 API 調用是根據它們是否減少對未來代幣的困惑。 Toolformer顯著提高了 6.7B 參數 GPT-J 模型的零樣本性能，使其能夠在一系列不同的下游任務上什至優於更大的 GPT-3 模型。

參考

Armen Aghajanyan 、Dmytro Okhonko 、Mike Lewis 、Mandar Joshi 、Hu Xu 、Gargi Ghosh 和 Luke Zettle moyer。 2021. Htlm ：語言模型的超文本預訓練與提示。

Eugene Agichtein 和 Luis Gravano。 2000. Snowball ：從大型純文本集合中提取關係。
在第五屆 ACM 數字圖書館會議記錄中，DL 00，第 85–94 頁，美國紐約州紐約市 。計算機協會。

Ricardo Baeza-Yates 、Berthier Ribeiro-Neto 等。 1999.現代信息檢索，第463卷。
ACM 出版社紐約。

Jonathan Berant 、Andrew Chou 、Roy Frostig 和 Percy Liang。 2013.從問答對對 Freebase 進行語義解析 。在 2013 年自然語言處理經驗方法會議記錄中，第 1533–1544 頁，西雅圖，華盛頓，美國 。計算林學協會。

塞巴斯蒂安·博爾若 、亞瑟·門施 、喬丹·霍夫曼 、特雷弗·蔡 、伊麗莎·盧瑟福 、凱蒂·米利肯 、喬治·範·登·德里斯切 、讓-巴蒂斯特·萊斯皮奧 、博格丹·達摩克 、艾丹·克拉克 、迭戈·德·拉斯·卡薩斯、奧雷莉亞·蓋伊 、雅各布·梅尼克 、羅曼·林 、湯姆Hen nigan、Saffron Huang 、Loren Maggiore 、Chris Jones 、Albin Cassirer 、Andy Brock 、Michela Paganini 、Ge offrey Irving 、Oriol Vinyals 、Simon Osindero 、Karen Simonyan、Jack W. Rae 、Erich Elsen 和 Laurent Sifre。 2021.通過從數万億個標記中檢索來改進語言模型。

謝爾蓋·布林。 1999.從萬維網中提取模式和關係 。在萬維網和數據庫中，第 172-183 頁，柏林，海德堡。

斯普林格柏林海德堡。

湯姆·布朗,本傑明·曼恩,尼克·瑞德,梅蘭妮·蘇比亞,賈里德·D·卡普蘭,普拉富拉·達里瓦爾,阿文德·尼拉坎坦,普拉納夫·希亞姆,吉里什·薩斯特里,阿曼達·阿斯凱爾,桑蒂尼·阿加瓦爾,阿里爾·赫伯特·沃斯,格雷琴·克魯格,湯姆·亨尼漢,RewonChild,AdityaRamesh, Daniel Ziegler 、Jeffrey Wu 、Clemens Winter 、Chris Hesse 、Mark Chen 、Eric Sigler 、Mateusz Litwin 、Scott Gray 、Benjamin Chess 、Jack Clark 、Christopher Berner、Sam McCandlish 、Alec Radford 、Ilya Sutskever 和 Dario Amodei。

2020.語言模型是少數學習者。 In Advances in Neural Information Processing Systems，第 33 卷，第 1877–1901 頁 。柯倫聯合公司

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay、Noam Shazeer、Vin odkumar Prabhakaran、Emily Reif、Nan Du、Ben Hutchinson、Reiner Pope、James Bradbury、Jacob Austin、Michael Isard、Guy Gur-Ari、Pengcheng Yin、Toju Duke、Anselm Levskaya、Sanjay Ghe mawat、Sunipa Dev , Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fe dus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai、Thanumalayan Sankara narayana Pillai、Marie Pellat、Aitor Lewkowycz、Erica Moreira、Rewon Child、Oleksandr Polozov、Katherine Lee、Zongwei Zhou、Xuezhi Wang、Bren nan Saeta、Mark Diaz、Orhan Firat、Michele Catasta、Jason Wei、Kathy Meier-赫爾斯特恩、道格拉斯·埃克、傑夫·迪恩、斯拉夫·P埃特羅夫和諾亞·菲德爾。 2022.

Palm：使用路徑擴展語言建模。

Karl Cobbe、Vineet Kosaraju、Mohammad Bavarian、Mark Chen、Heewoo Jun、Lukasz Kaiser、Matthias Plappert、Jerry Tworek、Jacob Hilton、Reiichiro Nakano 等。 2021. 訓練驗證者解決數學應用題。 arXiv 預印本 arXiv:2110.14168。

Marta R Costa-jussà、James Cross、Onur Çelebi、Maha Elbayad、Kenneth Heafield、Kevin Heffernan、Elahe Kalbassi、Janice Lam、Daniel Licht、Jean Maillard 等。 2022. 不遺餘力：擴展以人為本的機器翻譯。 arXiv 預印本 arXiv:2207.04672。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。 2019. BERT：用於語言理解的深度雙向轉換器的預訓練。在計算語言學協會北美分會 2019 年會議記錄中：人類語言技術，第 1 卷（長文和短文），第 4171-4186 頁，明尼蘇達州明尼阿波利斯。計算語言學協會。

Bhuwan Dhingra、Jeremy R. Cole、Julian Martin Eisenschlos、Daniel Gillick、Jacob Eisenstein 和 William W. Cohen。 2022.作為時間知識庫的時間感知語言模型。計算語言學協會會刊，10：257-273。

Leo Gao、Stella Biderman、Sid Black、Laurence Golding、Travis Hoppe、Charles Foster、Jason Phang、Ho race He、Anish Thite、Noa Nabeshima 等人。 2020. The pile：一個 800gb 的不同文本數據集，用於語言建模。 arXiv 預印本 arXiv:2101.00027。

Luyu Gao、Aman Madaan、Shuyan Zhou、Uri Alon、Pengfei Liu、Yiming Yang、Jamie Callan 和 Graham Neubig。 2022. Pal：程序輔助語言模型。

Kelvin Guu、Kenton Lee、Zora Tung、Panupong Pasu pat 和 Ming-Wei Chang。 2020.境界：檢索增強語言模型預訓練。

Junxian He、Jiatao Gu、Jiajun Shen 和 Marc Aurelio Ranzato。 2020.重新審視神經序列生成的自我訓練。在國際學習代表大會上。

或者 Honovich、Thomas Scialom、Omer Levy 和 Timo Schick。 2022.非自然指令：在（幾乎）沒有人工的情況下調整語言模型。

Gautier Izacard 和 Edouard Grave。 2021.將讀者的知識提煉到獵犬，以便提問和回答。在國際學習代表大會上。

Gautier Izacard、Patrick Lewis、Maria Lomeli、Lucas Hosseini、Fabio Petroni、Timo Schick、Jane Dwivedi Yu、Armand Joulin、Sebastian Riedel 和 Edouard Grave。 2022. Atlas：使用檢索增強語言模型進行少量學習。

Ziwei Ji、Nayeon Lee、Rita Frieske、Tiezheng Yu、Dan Su、Yan Xu、Etsuko Ishii、Yejin Bang、Andrea Madotto 和 Pascale Fung。 2022.自然語言生成中的幻覺調查。 ACM 計算調查。

Zhengbao Jiang、Frank F. Xu、Jun Araki 和 Graham Neubig。 2020.我們怎麼知道語言模型知道什麼？計算語言學協會彙刊，8:423–438。

Mandar Joshi、Eunsol Choi、Daniel Weld 和 Luke Zettlemoyer。 2017. TriviaQA：用於閱讀理解的大規模遠程監督挑戰數據集。在計算語言學協會第 55 屆年會論文集（第 1 卷：長文），第 1601–1611 頁，加拿大溫哥華。計算林學協會。

Armand Joulin、Edouard Grave、Piotr Bojanowski、Matthijs Douze、Hérve Jégou 和 Tomas Mikolov。 2016. 快速文本。 zip：壓縮文本分類模型。 arXiv 預印本 arXiv:1612.03651。

Nitish Shirish Keskar、Bryan McCann、Lav R. Varsh ney、Caiming Xiong 和 Richard Socher。 2019. Ctrl：用於可控生成的條件轉換器語言模型。

菲利普·科恩。 2005. Europarl：統計機器翻譯的平行語料庫。在機器翻譯峰會論文集 x：論文，第 79-86 頁。

Mojtaba Komeili、Kurt Shuster 和 Jason Weston。 2022.互聯網增強對話生成。在計算語言學協會第 60 屆年會論文集（第 1 卷：長文），第 8460-8478 頁，都柏林，愛爾蘭。

計算語言學協會。

Rik Koncel-Kedziorski、Subhro Roy、Aida Amini、Nate Kushman 和 Hannaneh Hajishirzi。 2016. MAWPS：數學單詞問題存儲庫。在計算語言學協會北美分會 2016 年會議記錄中：人類語言技術，第 1152-1157 頁，加利福尼亞州聖地亞哥。計算語言學協會。

湯姆·克維亞特科夫斯基、詹妮瑪利亞·帕洛瑪基、奧利維亞·雷德菲爾德、邁克爾·柯林斯、安庫爾·帕里克、克里斯·阿爾伯蒂、丹妮爾·愛潑斯坦、伊利亞·波洛蘇欣、雅各布·德弗林、肯頓·李、克里斯蒂娜·圖塔諾娃、利昂·瓊斯、馬修·凱爾西、張明偉、Andrew M. Dai、Jakob Uszkoreit、Quoc Le 和 Slav Petrov。 2019. 自然問題：問題回答研究的基準。計算語言學協會彙刊，7:452–466。

Angeliki Lazaridou、Elena Gribovskaya、Wojciech Stokowiec 和 Nikolai Grigorev。 2022. 互聯網增強語言模型，通過少量提示進行開放域問答。 arXiv 預印本 arXiv:2203.05115。

Patrick Lewis、Barlas Oguz、Ruty Rinott、Sebastian Riedel 和 Holger Schwenk。 2019. Mlqa：評估跨語言提取問答。 arXiv 預印本 arXiv:1910.07475。

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, 王天祿, 陳碩輝, Daniel Simig, Myle Ott, Na man Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O Horo, Jeff Wang、Luke Zettle moyer、Zornitsa Kozareva、Mona Diab、Veselin Stoy anov 和 Xian Li。 2021.使用多語言語言模型進行少量學習。

Joshua Maynez、Shashi Narayan、Bernd Bohnet 和 Ryan McDonald。 2020.關於抽象摘要中的忠實性和真實性。

David McClosky、Eugene Charniak 和 Mark Johnson。 2006.有效的解析自我訓練。在 NAACL 人類語言技術會議論文集中，主要會議，第 152-159 頁，美國紐約市。計算語言學協會。

Stephen Merity、Caiming Xiong、James Bradbury 和 Richard Socher。 2017.指針哨兵混合模型。在國際學習代表大會上。

苗神雲、梁朝春和蘇克義。 2020.用於評估和開發英語數學單詞問題解決者的多元化語料庫。在計算語言學協會第 58 屆年會論文集中，第 975-984 頁，在線。計算語言學協會。

Reiichiro Nakano, 雅各布希爾頓, Suchir Balaji, Jeff Wu, 歐陽龍、克里斯蒂娜·金、克里斯托弗·黑塞、Shantanu Jain、Vineet Kosaraju、William Saunders、Xu Jiang、Karl Cobbe、Tyna Eloundou、Gretchen Krueger、Kevin Button、Matthew Knight、Benjamin Chess 和 John Schulman。 2021. Webgpt：瀏覽器通過人工反饋輔助問答。

亞倫·帕里西、姚昭和諾亞·菲德爾。 2022. Talm：工具增強語言模型。

Arkil Patel、Satwik Bhattamishra 和 Navin Goyal。 2021. NLP模型真的能解決簡單的數學應用題嗎？在計算語言學協會北美分會 2021 年會議記錄中：人類語言技術，第 2080-2094 頁，在線。

計算語言學協會。

Fabio Petroni、Aleksandra Piktus、Angela Fan、Patrick Lewis、Majid Yazdani、Nicola De Cao、James Thorne、Yacine Jernite、Vladimir Karpukhin、Jean Maillard、Vassilis Plachouras、Tim Rocktäschel 和 Sebastian Riedel。 2021. KILT：知識密集型語言任務的基準。在計算語言學協會北美分會 2021 年會議記錄中：人類語言技術，第 2523-2544 頁，在線。計算語言學協會。

Fabio Petroni、Tim Rocktäschel、Sebastian Riedel、Patrick Lewis、Anton Bakhtin、Yuxiang Wu 和 Alexander Miller。 2019.語言模型作為知識庫？在 2019 年自然語言處理經驗方法會議和第 9 屆自然語言處理國際聯合會議 (EMNLP IJCNLP) 的會議記錄中，第 2463-2473 頁，中國香港。作為計算語言學協會。

Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei、Ilya Sutskever 等。 2019. 語言模型是無監督的多任務學習者。 OpenAI 博客，1(8):9。

Roi Reichart 和 Ari Rappoport。 2007.對在小型數據集上訓練的統計解析器進行增強和域適應的自我訓練。在第 45 屆計算語言學協會年會論文集中，第 616-623 頁，捷克共和國布拉格。計算林學協會。

Stephen E Robertson、Steve Walker、Susan Jones、Micheline M Hancock-Beaulieu、Mike Gatford 等。 1995. 霍加皮在 trec-3。 Nist 特別出版物 Sp，109：109。

Timo Schick、Jane Dwivedi-Yu、Zhengbao Jiang、Fabio Petroni、Patrick Lewis、Gautier Izacard、Qingfei You、Christoforos Nalmpantis、Edouard Grave 和 Se bastian Riedel。 2022. Peer：協作語言模型。

Timo Schick 和 Hinrich Schütze。 2021a。利用完形填空問題進行少量文本分類和自然語言推理。在會議記錄中

計算語言學協會歐洲分會第 16 屆會議：正卷，第 255-269 頁，在線。計算語言學協會。

Timo Schick 和 Hinrich Schütze。 2021b。使用預訓練語言模型生成數據集。在 2021 年自然語言處理經驗方法會議論文集中，第 6943-6951 頁，在線和多米尼加共和國蓬塔卡納。

計算語言學協會。

庫爾特·舒斯特、徐靜、Mojtaba Komeili、Da Ju、埃里克·邁克爾·史密斯、斯蒂芬·羅勒、梅根·翁、Moya Chen、Kushal Arora、Joshua Lane、Morteza Behrooz、William Ngan、Spencer Poff、Naman Goyal、Arthur Szlam、Y-Lan Boureau、梅蘭妮·卡姆·巴杜爾和傑森·韋斯頓。 2022. Blenderbot 3：部署的對話代理不斷學習負責任地參與。

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, 李亞光, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, De hao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan,萊奇·曼凱瑟琳·梅爾-赫爾斯特恩梅雷迪思·林格爾·莫里斯圖爾西·多西雷內利托·德洛斯·桑托斯托朱·杜克強尼·索拉克本·澤文伯根維諾德·庫馬爾·普拉巴卡蘭馬克·迪亞茲本·哈欽森克里斯汀·奧爾森亞歷杭德拉·莫里納艾琳·霍夫曼·約翰喬什·李, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda：對話應用程序的語言模型。

Ben Wang 和 Aran Komatsuzaki。 2021. GPT J-6B：一個 60 億參數的自回歸語言模型。 https://github.com/kingoflolz/mesh-transformer-jax 。

Yizhong Wang、Yeganeh Kordi、Swaroop Mishra、Al isa Liu、Noah A. Smith、Daniel Khashabi 和 Han naneh Hajishirzi。 2022.自我指導：將語言模型與自我生成的指導相結合。

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raf fel, Barret Zoph, Sebastian Borgeaud, Dani Yo gatama, Maarten Bosma, Denny Zhou, Donald Met zler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean,和威廉·費杜斯。 2022.
大型語言模型的新興能力。

Guillaume Wenzek、Marie-Anne Lachaux、Alexis Con neau、Vishrav Chaudhary、Francisco Guzmán、Armand Joulin 和 Edouard Grave。 2020. CCNet：從中提取高質量的單語數據集

網絡抓取數據。在第十二屆語言資源和評估會議記錄中，第 4003-4012 頁，法國馬賽。歐洲語言資源協會。

Shunyu Yao、Jeffrey Zhao、Dian Yu、Nan Du、Izhak Shafran、Karthik Narasimhan 和 Yuan Cao。 2022. React：在語言模型中協同推理和行動。

大衛·亞羅夫斯基。 1995.與監督方法相媲美的無監督詞義消歧。在計算語言學協會第 33 屆年會上，第 189-196 頁，美國馬薩諸塞州劍橋市。計算語言學協會。

Eric Zelikman、Yuhuai Wu、Jesse Mu 和 Noah D. 好人。 2022.星：用推理引導推理。

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher De wan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mi haylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura、Anjali Sridhar、Tianlu Wang 和 Luke Zettlemoyer。 2022.選擇：打開預訓練的轉換器語言模型。

API 詳情

在採樣和過濾 API 調用時 ,默認情況下我們使用τs = 0.05和τf = 1.0的值 即 ,我們只在<API>標記的概率至少為 5% 的位置進行 API 調用 ,並且我們保留 API 調用如果他們將損失減少至少 1.0 。我們只保留前k = 5 個這樣的位置 ,並對一段文本中標識的每個位置採樣最多m = 5 個API 調用 。由於下面描述的啟發式過濾 ,我們只在C的一小部分上為計算器和機器翻譯系統生成 API 調用 ;為了彌補這一點 ,我們為這些工具設置τs = 0.0 、 k = 20和m = 10 。由於生成的 API 調用集仍然相對較小 ,我們另外設置τf = 0.5 。

A.1 實施問答我們使用

Izacard 等人的 Atlas 模型。 (2022)對自然問題 (Kwiatkowski et al., 2019)作為我們的問答系統進行了微調 。為了創建C * ,我們使用 Atlas large ,使我們能夠高效地處理數百萬個 API 調用 ;在推理過程中 ,我們[illegible]。

計算器我們的計算器基於簡單的Python 腳本 ,僅支持運算符“+” 、 “-” 、 “*”和“/” 。對於語法無效的方程式 ,它不會返回任何結果 。對於採樣API 調用 ,我們將啟發式過濾器應用於我們的CCNet 子集 ,並且僅處理以下文檔 :(i)在 100 個標記的窗口內至少包含三個數字 ,其中一個數字是對其他兩個 ,(ii) 包含序列“=” 、“等於” 、 “等於” 、“總計” 、 “平均值”中的一個 ,後跟一個數字 ,或 (iii) 包含至少三個數字 ;對於只符合最後一個標準的文本 ,我們只保留 1% 的隨機子集。

Calendar用於創建我們的數據集C * ,我們假設在這種情況下的日曆日期應該是文件的日期

創建 。我們通過從 URL 中提取日期（如果存在)來對此進行近似 。我們過濾掉無法提取日期的文本 ,留下大約18% 的文檔。

機器翻譯對於訓練和推理 ,我們使用 600M 參數 NLLB (Costa jussà et al., 2022)作為我們的機器翻譯 (MT)模型。使用 fastText 分類器自動檢測源語言（Joulin 等人， 2016) ,而目標語言始終設置為英語 。由於大部分 CCNet 數據集都是英文的 ,我們在生成 API 調用之前過濾掉僅包含英文文本的部分 。更具體地說 ,我們只保留那些包含非英語語言文本塊的段落 ,其前後是英文文本 。我們使用大小為 10 個標記的文本塊 。確定

中間文本塊是否使用不同於英語的語言 ,我們再次使用置信度大於 0.8 的 fastText 分類器 。我們還過濾掉任何僅包含數字或特殊符號的文本塊 。這種過濾機制使我們能夠通過將 API 調用生成集中在MT 工具可能有用的地方來更有效地生成數據 。在生成 MT API 調用之後 ,我們還從我們的訓練集中刪除了 MT 工具的輸入出現在 API 調用之後而不是之前的那些 。雖然在數據生成期間模型可以提前生成 API 調用 ,但這在推理時是不可能的 ,因此我們希望阻止模型在這種情況下調用 API 。

A.2 提示下面，

我們列出了用於對所考慮的每個工具進行 API 調用示例的提示。

問答我們對問答工具使用以下提示 :您的任務是將對問答 API 的調用添加到一段文本中。

這些問題應該可以幫助您獲得完成文本所需的信息 。您可以通過編寫“[QA(question)]”來調用 API ,其中“question”是您要問的問題 。以下是 API 調用的一些示例： 輸入 :Joe Biden 出生於賓夕法尼亞州的斯克蘭頓。

輸出 :Joe Biden 出生於 [QA(Where was Joe Biden born?)] Scranton, [QA(Scranton is In which state is?)]

賓夕法尼亞州。

輸入 :可口可樂 ,或可口可樂 ,是可口可樂公司生產的一種碳酸軟飲料。

輸出 :可口可樂 ,或 [QA （“可口可樂的其他名稱是什麼？”）] 可口可樂 ,是由 [QA （“誰生產可口可樂？”）] 可口可樂製造的碳酸軟飲料公司。

輸入 :x
輸出：

計算器我們對計算器使用以下提示：

您的任務是將對計算器 API 的調用添加到一段文本中。

這些電話應該可以幫助您獲得完成文本所需的信息。您可以通過編寫“[Calculator(expression)]”來調用 API，其中“expression”是要計算的表達式。以下是 API 調用的一些示例：

輸入：下一項的數字是 18 + 12 x 3 = 54。

輸出：下一項的數字是 18 + 12 x 3 = [Calculator(18 + 12 * 3)] 54。

輸入：人口為 658,893 人。
這是全國平均水平 5,763,868 人的 11.4%。

輸出：人口為 658,893 人。
這是 [計算器(658,893 / 11.4%)] 5,763,868 人的全國平均水平的 11.4%。

輸入：總共進行了 252 場資格賽，打進了 723 個進球（平均每場比賽進球 2.87 個）。這比去年的 2169 個進球少了三倍。

輸出：總共進行了 252 場資格賽，打進 723 球（平均每場比賽 [計算器（723 / 252）] 2.87）。這比 [Calculator(723 - 20)] 去年的 703 個進球多了 20 個進球。

輸入：我 1994 年去了巴黎，一直呆到 2011 年，總共 17 年。

輸出：我 1994 年去了巴黎，一直呆到 2011 年，所以總共 [Calculator(2011 - 1994)] 17 年。

輸入：據此，我們有 4 * 30 分鐘 = 120 分鐘。

輸出：由此，我們有 4 * 30 分鐘 = [計算器(4 * 30)] 120 分鐘。

輸入：x
輸出：

維基百科搜索我們使用維基百科搜索工具的以下提示：

你的任務是完成一段給定的文本。您可以使用維基百科搜索 API 來查找信息。您可以通過寫“[WikiSearch(term)]”來做到這一點，其中“term”是您要查找的搜索詞。以下是 API 調用的一些示例：

輸入：加納國旗上的顏色含義如下：紅色代表烈士的鮮血，綠色代表森林，金色代表礦藏。

輸出：加納國旗上的顏色具有以下含義：紅色代表 [WikiSearch(Ghana flag red meaning)] 烈士的鮮血，綠色代表森林，金色代表礦藏。

輸入：但是納米材料生產過程中有哪些風險？一些納米材料可能會引起各種肺損傷。

輸出：但是納米材料生產過程中的風險是什麼？

[WikiSearch(nanomaterial production risks)] 一些納米材料可能會引起各種肺損傷。

輸入：二甲雙胍是 2 型糖尿病和肥胖患者的一線藥物。

輸出：二甲雙胍是 [WikiSearch(二甲雙胍一線藥物)] 2 型糖尿病和肥胖患者的一線藥物。

輸入：x
輸出：

機器翻譯我們對機器翻譯工具使用如下提示：

您的任務是使用機器翻譯 API 完成一段給定的文本。

您可以通過編寫“[MT(text)]”來實現，其中 text 是要翻譯的文本成英文。
這裡有些例子：

輸入：他出版了一本書：O homem suprimido（“The Suppressed Man”）
輸出：他出版了一本書：O homem suprimido [MT(O homem suprimido)]（“被壓抑的人”）

輸入：在 Morris de Jonge 的 Jeschuah, der klassische jüdische Mann 中，描述了一位猶太作家

輸出：在Morris de Jonge的Jeschuah, der klassische jüdische Mann [MT(der klassische jüdische Mann)]中，描述了一位猶太作家

輸入：南京高淳縣住宅和城市建設局城 市 新區設計 參考平面 高淳是省會南京的七個區之一

輸出： [MT(南京高淳縣住宅和城市建設局城市新區設計計)]參考平面高淳是省會南京的七個區之一

輸入：x
輸出：

日曆我們使用日曆工具的提示如下：

您的任務是將對日曆 API 的調用添加到一段文本中。 API 調用應該可以幫助您獲得完成文本所需的信息。您可以通過編寫“[Calendar()]”來調用 API 以下是 API 調用的一些示例：

輸入：今天是第一個星期五
年。
輸出：今天是第一個 [Calendar()]
一年中的星期五。

輸入 :美國總統是喬·拜登。

輸出 :美國總統是 [Calendar()] Joe Biden。

輸入 :星期幾是星期三。

輸出 :當前星期幾是 [Calendar()] 星期三。

輸入 :從現在到聖誕節還有 30 天。

輸出 :從現在到聖誕節的天數是 [Calendar()] 30。

輸入 :商店週末不開門 ,所以今天關門了。

輸出 :商店在周末從不營業 ,所以今天 [Calendar()] 它關門了。

輸入 :x
輸出：

B 模具加工培訓

每個 API 我們最多使用 25k 個示例 。最大序列長度 1,024。有效批量大小為 128 。所有模型均使用 DeepSpeed 的 ZeRO-3 進行訓練（Rasley等人 ,2020 年） 。我們使用了 8 個帶 BF16 的 NVIDIA A100 40GB GPU 。最多訓練 2k 步 ,我們在 CCNet 的小型開發集上評估 PPL ,每 500 步包含 1,000 個示例。

我們選擇表現最好的檢查點。

C 零射擊提示

C.1 LAMA 和TEMPLAMA

對於 LAMA 和TEMPLAMA ,給定輸入文本x ,我們使用以下提示 :請完成以下文本 ,使其與事實相符 :x。

C.2 數學基準

對於所有數學基準 ,給定上下文x和問題q ,我們的提示是：xq 答案是。

C.3 問答

對於所有問答數據集 ,包括DATESET ,我們只需在問題前加上Answer the following question: 。如果問題尚未以問號結尾 ,我們會附加一個問號。

C.4 多語言問答

對於 MLQA ,給定上下文x和問題q ,我們的提示是 :你的任務是

模板	尺寸
{ago was, are there until} {past_date, future_date} 多少天？	400
(current_date – past_date) {days, weeks, months, years} 之前是哪{星期幾 、月幾 、月 、年}？	800
在 (future_date – current_date) 天是哪{星期幾 、月幾 、月 、年} ？	800
{過去的日期 ,未來的日期}是一周中的哪一天？	400
什麼{星期幾 ,月幾 ,月 ,年} {是 ,是}它 {前天 ,是星期三 ,今天 ,明天 ,後天}？	4,000
今年的{星期幾 、月份日期}{是 、過去}是什麼假期？	1,800
今年有多少{天 、週 、月 、年}假期？	1,200
全部的	9,400

表 11 :用於創建隨機選擇 current_date 的DATESET 的模板 。對於每個當前日期 ,生成一個隨機的過去日期和未來日期並用於填充每個模板（如果相關） 。在涉及假期的模板中使用了美國的聯邦假期（例如 ,感恩節） 。

根據以下段落回答問題： x 現在用英語回答以下問題：
q.

D日期設定

DATESET是通過首先隨機選擇 500 個“當前日期”創建的。對於每個當前日期 ,在四年範圍內隨機選擇另一個相對過去/未來的日期 ,這兩個日期用於填充表11中的查詢模板 。使用第一個模板的此類查詢的一個示例是， “ 2020 年 8 月 14 日是多少天前？”如果被調用 ,日曆工具將返回假定的當前日期（例如，“今天是2020 年 11 月 20 日 ,星期日”） 。