# The Web Can Be Your Oyster for Improving Language Models

**Junyi Li**[1,3,5], **Tianyi Tang**[1], **Wayne Xin Zhao**[1,5*], **Jingyuan Wang**[4] ,
**Jian-Yun Nie**[3] and **Ji-Rong Wen**[1,2,5]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China [3]DIRO, Université de Montréal
[4]School of Computer Science and Engineering, Beihang University
[5]Beijing Key Laboratory of Big Data Management and Analysis Methods
{lijunyi,steven_tang}@ruc.edu.cn batmanfly@gmail.com

## Abstract

Pretrained language models (PLMs) encode a large amount of world knowledge. However, as such knowledge is frozen at the time of model training, the models become *static* and *limited* by the training data at that time. In order to further improve the capacity of PLMs for knowledge-intensive tasks, we consider augmenting PLMs with the large-scale web using search engine. Unlike previous augmentation sources (*e.g.,* Wikipedia data dump), the web provides broader, more comprehensive and constantly updated information. In this paper, we present a web-augmented PLM – UNIWEB, which is trained over 16 knowledge-intensive tasks in a unified text-to-text format. Instead of simply using the retrieved contents from web, our approach has made two major improvements. Firstly, we propose an adaptive *search engine assisted learning* method that can self-evaluate the confidence level of PLM's predictions, and adaptively determine when to refer to the web for more data, which can avoid useless or noisy augmentation from web. Secondly, we design a pretraining task, *i.e., continual knowledge learning*, based on salient spans prediction, to reduce the discrepancy between the encoded and retrieved knowledge. Experiments on a wide range of knowledge-intensive tasks show that our model significantly outperforms previous retrieval-augmented methods. Our code and data can be accessed at this link https://github.com/RUCAIBox/UniWeb

## 1 Introduction

With large-scale neural networks, pretrained language models (PLMs) (Brown et al., 2020; Zhao et al., 2023) can encode a large amount of world knowledge, showing phenomenal capability in knowledge-intensive tasks such as fact checking and open-domain question answering (QA). However, this capacity is naturally limited by the information contained in pretraining or finetuning datasets (usually fixed once collected), which are neither *up-to-date* nor *complete* (Komeili et al., 2021; Ji et al., 2022). Although model scaling (Brown et al., 2020; Chowdhery et al., 2022; Thoppilan et al., 2022) is a viable way to improve the knowledge capacity of PLMs, it still uses *static* pretraining datasets, and also leads to significantly larger computational costs with increased model sizes. As a result, the outdated or incomplete knowledge encoded by PLMs may lead to hallucination or incorrect generations even though the results look plausible (Ji et al., 2022).

| | |
|---|---|
| **Question** | Which popular **Korean show** was **recently** green lit for a new season? |
| **Answer** | Squid Game |
| **Wikipedia** | There are no results for the question. |
| **Web** | [...] Netflix announce Sunday that the wildly popular South Korean show is green lit for a second season. "And now, Gi-hun returns" "The Front Man returns. Season 2 is coming." "Squid Game" is a fictional drama from South Korea in which contestants who are desperately in need of money play deadly children's games to win cash prizes. [...] |
| URL: | https://www.cnn.com/2022/06/12/media/squid-game-season-2/index.html |
| **T5 w/o Web** | The Walking Dead ✗ |
| **T5 w/ Web** | Squid Game ✓ |

Table 1: An example showing that the web covers both more comprehensive (*e.g.,* Korean show) and up-to-date (*e.g.,* recently) information than Wikipedia. Based on the latest news returned by Google Search, T5-LARGE can answer the question correctly.

Recently, by drawing the idea from semi-parametric approaches (Zhao et al., 2022; Guu et al., 2020; Lewis et al., 2020b; Borgeaud et al., 2022), retrieval-augmented approaches have been proposed to equip PLMs with the ability to directly access an external database. As a major knowledge resource, Wikipedia has been widely used in previous work. While being highly accurate and well-structured, Wikipedia only covers *limited* information, both in scope and in time. Besides, even for

[*] Corresponding author

*Findings of the Association for Computational Linguistics: ACL 2023*, pages 728–746
July 9-14, 2023 ©2023 Association for Computational Linguistics

the topics that Wikipedia covers, grounding PLMs' decisions on a single source of knowledge may create biases (Wagner et al., 2016). Considering these issues, it is time to look beyond Wikipedia (or similar single-source databases) and access more *broader*, *in-depth*, and *up-to-date* knowledge from more sources. Inspired by (Komeili et al., 2021; Piktus et al., 2021), we select the *web* as the retrieval resource for enlarging the knowledge capacity of PLMs. To motivate our approach, Table 1 presents a sample question that T5 successfully answers with the support of the web (providing the latest news), but not Wikipedia. As we can see, timely and relevant supporting evidence is the key to solve such tasks for PLMs.

In this paper, we aim to capitalize on the web as a source of up-to-date and comprehensive knowledge to solve a wide range of knowledge-intensive tasks. Unlike previous web-augmented studies (Nakano et al., 2021; Menick et al., 2022) that mostly focus on single tasks, we seek to develop a unified framework to integrate the use of the web in PLMs for multi-task solving. Although the idea of leveraging the web for improving PLMs is appealing, it is non-trivial to develop an effective solution. First, PLMs do not always need external evidence for task solving, especially considering the fact that the web contains noisy, biased, or harmful information (Luccioni and Viviano, 2021). Simply retrieving knowledge without considering the example difficulty and PLMs' own capabilities may steer models towards unexpected outputs. Second, PLMs are usually pretrained at an earlier time on a limited corpus, leading to a discrepancy between the encoded knowledge and the retrieved knowledge (*i.e.,* web contents). Therefore, we need more principled approaches to properly integrating the new knowledge into PLMs.

To address the above issues, we present a web-augmented PLMs, UNIWEB, to improve the capacity in knowledge-intensive tasks. Instead of using neural network-based retriever, we employ a commercial search engine (*i.e.,* Google Search) to obtain high-quality and comprehensive retrieval results from the web. Based on this idea, we make two major technical contributions. First, we propose a *search engine assisted learning* method that can selectively query the web only when PLM is unconfident in its predictions. For this purpose, we design a self-evaluation mechanism to estimate the confidence level of PLMs on the task examples. Secondly, to reduce the discrepancy between the encoded and retrieved knowledge, we design a pretraining task, *continual knowledge learning*, to integrate the retrieved knowledge into PLMs by predicting the salient masked spans in web documents. To train the UNIWEB model, we convert different knowledge-intensive tasks into a unified text-to-text format, and conduct supervised multitask training over 16 tasks across seven categories.

To the best of our knowledge, our model is the first unified web-augmented PLM for a wide range of knowledge-intensive tasks. Extensive experiments show that PLMs can significantly benefit from such an approach and a single unified PLM (UNIWEB) is able to achieve (near) state-of-the-art performance on all 16 tasks.

## 2 Related Work

**Retrieval-Augmented PLMs.** Augmenting a pretrained language model with retrieval has been extensively studied in existing literature (Lewis et al., 2020b; Borgeaud et al., 2022; Izacard et al., 2022; Lee et al., 2019; Guu et al., 2020). For example, REALM (Guu et al., 2020) and RAG (Lewis et al., 2020b), incorporate a differentiable retriever into pretrained models, leading to promising results on question answering. However, these studies usually rely on a sub-optimal retriever to access a static and limited knowledge resource, *i.e.,* Wikipedia. By contrast, our model utilizes the well-developed search engine to gain broader, more in-depth, and up-to-date knowledge from the web. Several studies have also looked at how Internet can help the models, but only focus on single tasks such as question answering (Nakano et al., 2021; Menick et al., 2022) and dialogue (Komeili et al., 2021). WebGPT (Nakano et al., 2021) uses human feedback to optimize answer quality by hiring massive labelers to judge the accuracy of answers. Komeili et al. (2021) retrieves knowledge from the web for every dialogue without considering the necessity. Piktus et al. (2021) only presents an empirical study to investigate the impact of replacing Wikipedia with a large-scale web-like corpus and adopting different retrieval models. We are also aware of some related studies (Jiang et al., 2023), but we have taken a different active approach for knowledge retrieval. In this paper, we develop a unified language model for solving a wide spectrum of knowledge-intensive tasks. Our model can selectively decide whether to access the web, and continuously learn from the

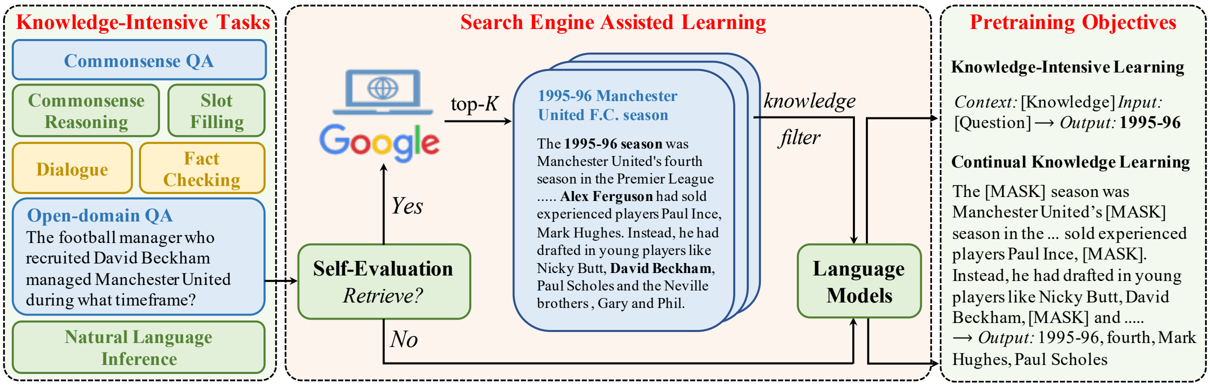

Knowledge-Intensive Tasks
Commonsense QA
Commonsense Reasoning
Slot Filling
Dialogue
Fact Checking
Open-domain QA
The football manager who recruited David Beckham managed Manchester United during what timeframe?
Natural Language Inference
Search Engine Assisted Learning
Google
top-K
Yes
Self-Evaluation
Retrieve?
No
1995-96 Manchester United F.C. season
The 1995-96 season was Manchester United's fourth season in the Premier League ..... Alex Ferguson had sold experienced players Paul Ince, Mark Hughes. Instead, he had drafted in young players like Nicky Butt, David Beckham, Paul Scholes and the Neville brothers , Gary and Phil.
knowledge
filter
Language Models
Pretraining Objectives
Knowledge-Intensive Learning
Context: [Knowledge] Input: [Question] ⟶ Output: 1995-96
Continual Knowledge Learning
The [MASK] season was Manchester United's [MASK] season in the ... sold experienced players Paul Ince, [MASK]. Instead, he had drafted in young players like Nicky Butt, David Beckham, [MASK] and .....
⟶ Output: 1995-96, fourth, Mark Hughes, Paul Scholes

Figure 1: Overview of our proposed web-augmented pretrained language model UNIWEB.

retrieved knowledge.

**Knowledge-Intensive Learning.** Recent work has shown that PLMs' parameters have implicitly stored linguistic or factual knowledge (Petroni et al., 2019; Roberts et al., 2020). However, the implicitly encoded knowledge is limited by the model's scale and training data, contradicting the dynamic nature of the world. Hence, many researchers propose to fuse relevant external knowledge from texts with the encoded knowledge of PLMs to deal with knowledge-intensive tasks such as open-domain QA (Guu et al., 2020; Lewis et al., 2020b), entity linking (Wu et al., 2019), fact verification (Liu et al., 2019b), and commonsense reasoning (Lin et al., 2020). Wikipedia has been the most widely used knowledge source for these tasks, which is still limited despite its wide coverage. Instead, we rely on the real-time web. The existing studies usually design task-specific training, architecture, and knowledge fusion method to exploit knowledge sources. In this work, we aim to develop a single unified framework that can be used for most knowledge-intensive tasks.

## 3 Task Formulation

Knowledge-intensive tasks (Yin et al., 2022) aim to leverage external knowledge resources to accomplish a broad range of tasks such as open-domain question answering and fact verification.

Following prior work (Lewis et al., 2020b; Guu et al., 2020), we employ a retrieval-augmented generation framework that consists of two components: a retriever $\mathcal{R}$ and a generator $\mathcal{G}$. Given an input text $\mathcal{X}$ such as a question, the retriever $\mathcal{R}$ learns to retrieve a set of top-$K$ passages $\mathcal{P} = \{p_1, ..., p_K\}$ from a knowledge resource. Conditioned on the input text $\mathcal{X}$ and the retrieved passages $\mathcal{P}$, the generator $\mathcal{G}$ aims to generate the output text $\mathcal{Y}$. The model is trained to maximize the joint likelihood:

$$\text{Pr}(\mathcal{Y}|\mathcal{X}) = \sum_{\mathcal{R},\mathcal{G}} \text{Pr}(\mathcal{P}|\mathcal{X})\text{Pr}(\mathcal{Y}|\mathcal{P},\mathcal{X}). \quad (1)$$

To implement the framework, previous studies usually adopt a trainable neural retriever based on a (single) knowledge resource such as Wikipedia or knowledge bases. However, such an approach can only access limited, static knowledge. In this paper, we rely on a general, off-the-shelf search engine as the retriever to access both *comprehensive* and *up-to-date* knowledge from the whole web.

## 4 Approach

Our proposed web-augmented PLM, UNIWEB, is depicted in Figure 1. We first transform knowledge-intensive tasks into a unified text-to-text paradigm and consider the web as a general form of knowledge source. Based on the retrieved knowledge, we further design two training objectives to build our model. In the next sections, we will describe our method in detail.

### 4.1 Knowledge-Intensive Tasks Unification

Previous retrieval-augmented approaches usually adopt diverse architectures and different types of knowledge resources (Yin et al., 2022). Instead, we aim to leverage the general knowledge source (*i.e.*, the web) to develop a unified framework that can fulfill various (or most) knowledge-intensive tasks. Specifically, we unify 16 typical knowledge-intensive tasks across 7 task families, including fact checking, slot filling, dialogue, open-domain question answering, commonsense question answering, commonsense reasoning, and natural language inference. We convert these tasks as a general text-to-text transformation for training a unified PLM.

These tasks are mainly from the studies (Petroni et al., 2020; Piktus et al., 2021), in which the original tasks of fact checking, slot filling, dialogue, and open-domain QA are designed specifically based on the retrieved knowledge from Wikipedia, while other tasks of commonsense QA, commonsense reasoning, and natural language inference focus on some more specific commonsense knowledge, going beyond Wikipedia. We consider these knowledge-intensive tasks as typical NLP tasks to show that the large-scale web can be specially useful for satisfying diverse information needs. More details about each task can be found in Appendix A.

## 4.2 Web-based Knowledge Retrieval

Unlike prior work that retrieves documents from offline corpora such as Wikipedia (Guu et al., 2020; Lewis et al., 2020b), we propose to retrieve *comprehensive* and *up-to-date* information from the online web through a general-purpose search engine. Although it is intuitive to extend the retrieval-augmented framework with the web as the knowledge resource, it is non-trivial to effectively leverage the knowledge found on the web. The documents on the web have inconsistent quality, and contain noisy, biased, or even harmful contents (Luccioni and Viviano, 2021). Low-quality content may steer PLMs towards seemingly plausible but factually incorrect outputs (Ji et al., 2022). On the other hand, compared to a local neural retriever, black-box search engines can only be accessed through queries, which is less controllable and not easy to filter out noisy contents from the search results. In addition, PLMs do not always need external knowledge for task solving, especially for easy tasks. Therefore, we should request for more knowledge only when needed.

### 4.2.1 PLM Knowledge Evaluation

To address the above challenges, it is essential to evaluate PLMs' own capabilities in a task and the necessity to refer to external knowledge. In our approach, we consider a non-trivial question before retrieval: *does a PLM need to retrieve knowledge for a specific task instance?* For this purpose, we investigate whether or not PLMs can correctly answer questions without using external evidence. According to the recent study (Kadavath et al., 2022), PLMs can self-evaluate the confidence level of their generation results (*e.g., True* or *False*). Hence, we propose to utilize the self-evaluation mechanism to determine whether it is necessary to access additional web information.

**Self-Evaluation.** Specifically, we hypothesize that when a model "knows" the true output (*i.e.,* confident about its output) for a specific input, sampling the outputs many times would result in an output distribution with small entropy. Following Kadavath et al. (2022), we sample $n$ ($n$ = 200) different outputs for each input and estimate the entropy of the output distribution as follows:

$$\begin{aligned} H(\hat{\mathcal{Y}}|\mathcal{X}) &= \mathbb{E}[-\log \Pr(\hat{\mathcal{Y}}|\mathcal{X})] \quad (2) \\ &= \mathbb{E}\left[-\sum_{w_i \in \hat{\mathcal{Y}}} \log \Pr(w_i|\mathcal{X}, w_{<i})\right], \end{aligned}$$

where $\hat{\mathcal{Y}} = \langle w_1, ..., w_i, ..., w_m \rangle$ is the output text generated by the model $\mathcal{G}$. Then, we set an entropy threshold $\eta$. If $H(\hat{\mathcal{Y}}|\mathcal{X})$ is higher than $\eta$, it means that the model is unconfident about its outputs and needs supporting evidence from the web, otherwise, it does not. We will further demonstrate the predictive power of the entropy (Eq. (2)) in estimating the model confidence for knowledge retrieval.

### 4.2.2 Web Knowledge Retrieval

In *active learning* (Ren et al., 2021), a prediction model can interactively query for labeling examples with low confidence levels. This learning method can not only reduce the cost of data labeling, but also remove those noisy and unhelpful data that models cannot benefit from. Inspired by this, we propose a *search engine assisted learning* approach, in which PLMs choose those hard cases that they cannot solve (assessed by self-evaluation) to query the off-the-shelf search engine for knowledge retrieval. Different from active learning, our approach does not directly query for the final answer (largely reducing the labeling efforts), but instead the supporting evidence for solving the task. After retrieving knowledge from the web, it is critical to filter out noisy contents and select the most helpful and relevant knowledge that can enhance PLMs' confidence to generate correct outputs. Therefore, we elaborate a *two-stage filter mechanism* to filter the retrieved knowledge.

**Search Engine Assisted Learning.** Specifically, for those hard examples, we take their input text $\mathcal{X}$ verbatim as a search query and issue a call to Google Search via API. For each query, we retrieve top-$K$ HTML pages and parse them to obtain clean texts, resulting in a set of passages

$\mathcal{P} = \{p_1, ..., p_K\}$. To filter out noisy and irrelevant information, in the first stage, we chunk each passage into paragraphs, compute the cosine similarity between input and paragraph embeddings, and select the five most relevant paragraphs to form the final passage. In the second stage, we adopt the same method as self-evaluation (Eq. 2) to compute the model confidence given the input and each processed passage and select those passages with high confidence as the final evidence.

### 4.3 Knowledge-based Model Pretraining

In most previous work, the retrieval model is either pretrained using self-supervised objective such as MLM (Guu et al., 2020; Borgeaud et al., 2022) or trained for specific tasks (Lewis et al., 2020b). In this work, we focus on explicitly training web-augmented PLMs in a supervised and massively multi-task fashion (Aribandi et al., 2022) using the mixture of knowledge-intensive tasks (Section 4.1). Besides, to integrate the retrieved knowledge into PLMs, we design a continual knowledge learning task based on the retrieved passages.

**Knowledge-Intensive Learning.** This pretraining objective uses the retrieved knowledge and labeled data from the unified knowledge-intensive tasks. Formally, given an input text $\mathcal{X}$ and retrieved passages $\mathcal{P}$, this objective is to minimize the negative log-likelihood loss over the output text $\mathcal{Y}$:

$$\mathcal{L}_{KIL} = -\sum_{i=1}^{m} \log \Pr(w_i | w_{<i}, \mathcal{X}, \mathcal{P}), \quad (3)$$

where $w_i$ denotes the $i$-th token of the output text $\mathcal{Y}$. We concatenate the input text $\mathcal{X}$ and retrieved passages $\mathcal{P}$ using the manually-written task-specific prompts (shown in Appendix A). Pretrained on the unified knowledge-based text-to-text format, our model can be easily applied to diverse knowledge-intensive tasks. It has been reported that ensembling many tasks, distributions and domains during pretraining can improve PLMs' generalization to new tasks (Aribandi et al., 2022).

**Continual Knowledge Learning.** Due to the limited pretraining on single static corpus, the knowledge encoded in PLMs has a discrepancy with the retrieved knowledge from the web. Thus, to reduce the discrepancy and integrate the newly retrieved knowledge into PLMs, we design a self-supervised pretraining task, *i.e.,* continual knowledge learning. For most knowledge-intensive tasks such as slot filling and fact verification, named entities are of special importance. Thus, this pretraining task aims to predict the salient masked spans (*i.e.,* named entities) in retrieved passages. Firstly, we use a BERT-based (Devlin et al., 2019) tagger trained on CoNLL-2003 data (Sang and De Meulder, 2003) to identify name entities and then mask entities such as "United States". Then, our model will be trained to predict these masked spans by minimizing the masked span prediction loss:

$$\mathcal{L}_{CKL} = -\sum_{k=1}^{K} \sum_{j=1}^{m} \log \Pr(s_j | \tilde{p}_k), \quad (4)$$

where $s_j$ is the $j$-th masked span for the passage $p_k$, and $\tilde{p}_k$ denotes the unmasked tokens in $p_k$.

## 5 Experiments

In this section, we detail the experimental setup and then highlight the main observations of our results.

### 5.1 Experimental Setup

**Knowledge Source.** In large-scale pretraining, we leverage an open massive web corpus CCNet (Wenzek et al., 2020) to provide documents with diverse topics, approximating the realistic web. Following Piktus et al. (2021), we select the CCNet snapshot corresponding to the August 2019 Common Crawl snapshot which covers a wide range of 134M web documents and finally yields 906M passages of 100 tokens. CCNet processes Common Crawl through deduplication, language identification and quality filtering based on perplexity calculated by a language model. In downstream fine-tuning, we test with the off-the-shelf search engine, *i.e.,* Google Search, to retrieve documents from the real-time web. Specifically, we utilize the input text verbatim as query and request a call to Google Search via API[1]. Besides, for the Wikipedia-based baselines, we use the 2019/08/01 Wikipedia snapshot from the KILT benchmark (Petroni et al., 2020), consisting of 5.9M documents split into 22.2M passages of 100 tokens. This data snapshot is temporally the closest to the CCNet corpus for fair comparison.

**Pretraining Tasks.** As described in Section 4.1, we unify 16 knowledge-intensive tasks across seven task families during pretraining:

- **Fact Checking**: FEVER (Thorne et al., 2018).

[1] https://developers.google.com/custom-search

| Models | Fact Checking | Slot Filling | | Dialogue | Open-domain QA | | | |
|---|---|---|---|---|---|---|---|---|
| | **FEVER** | **T-REx** | **zsRE** | **WoW** | **NQ** | **HotpotQA** | **TriviaQA** | **ELI5** |
| | | | | *w/o Retrieval* | | | | |
| **BART**$_{\text{LARGE}}$ | 78.93 | 45.06 | 9.14 | 12.86 | 21.75 | 15.37 | 32.39 | 20.55 |
| **T5**$_{\text{LARGE}}$ | 80.31 | 50.63 | 10.34 | 12.67 | 28.50 | 18.98 | 35.90 | **20.60** |
| | | | | *w/ Wikipedia* | | | | |
| **REALM** | 76.22 | 53.35 | 39.38 | - | 40.40 | 22.23 | 65.44 | 10.23 |
| **RAG** | 86.31 | 59.20 | 44.74 | 13.11 | 44.39 | 26.97 | 71.27 | 14.05 |
| **BART+DPR** | 86.74 | 59.16 | 30.43 | 15.19 | 41.27 | 25.18 | 58.55 | 17.41 |
| **BART+DPR**$_{\text{MULTI}}$ | 86.32 | 78.50 | 57.95 | 15.33 | 39.75 | 31.77 | 59.60 | 17.07 |
| **FID+DPR**$_{\text{MULTI}}$ | 88.99 | 82.19 | 71.53 | 15.66 | 49.86 | 36.90 | 71.04 | 16.45 |
| | | | | *w/ CCNet* | | | | |
| **FID+DPR**$_{\text{MULTI}}$ | 85.74 | 52.06 | 28.47 | 15.22 | 45.15 | 27.29 | 67.49 | 16.14 |
| **FID+DPR**$_{\text{CCNET}}$ | 87.43 | 57.02 | 36.55 | 15.29 | 48.61 | 31.64 | 73.06 | 15.76 |
| **FID+BM25** | 89.12 | 62.12 | 43.92 | 17.28 | 46.05 | 34.10 | **78.21** | 15.59 |
| | | | | *w/ Web* | | | | |
| **UniWeb** | **91.69** | **83.58** | **72.42** | **20.87** | **54.37** | **40.73** | 77.01 | 18.34 |

Table 2: Evaluation results on the test set for fact checking, slot filling, dialogue, and open-domain QA. We report *Accuracy* for FEVER, T-REx, and zsRE; *EM* for NQ, HotpotQA, and TriviaQA; *ROUGE-L* for ELI5 and *F1-score* for WoW. These results come from no-retrieval models (top section), Wikipedia/CCNet-based models (middle section), and Web-based models (bottom section). **Bold** and underline denote the best and second best methods.

- **Slot Filling**: T-REx (ElSahar et al., 2018) and zero-shot RE (Levy et al., 2017).
- **Dialogue**: Wizard-of-Wikipedia (Dinan et al., 2019).
- **Open-domain QA**: TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and ELI5 (Shuster et al., 2020).
- **Commonsense QA**: CommonsenseQA (Talmor et al., 2019), SocialIQa (Sap et al., 2019), CosmosQA (Huang et al., 2019), and PIQA (Bisk et al., 2020).
- **Commonsense Reasoning**: NumerSense (Lin et al., 2020) and WinoGrande (Sakaguchi et al., 2020).
- **Natural Language Inference**: $\alpha$NLI (Bhagavatula et al., 2020) and HellaSwag (Zellers et al., 2019).

We convert these tasks into a unified text-to-text format. We take the input text as query to retrieve top 10 passages from CCNet. After pre-processing, we mix the training set of these datasets to pretrain our model. We present the statistics of datasets and pre-processing details in Appendix A.

**Baselines**. We compare **UniWeb** to a wide range of models as follows:

- **BART** (Lewis et al., 2020a) and **T5** (Raffel et al., 2020). These are two representative text-to-text PLMs for solving knowledge-intensive tasks. We adopt the large version for a fair comparison.
- **REALM** (Guu et al., 2020) and **RAG** (Lewis et al., 2020b). They are two well-known retrieval-augmented PLMs combining with a nonparametric memory of Wikipedia via a neural retriever.
- **Fusion-in-Decoder (FID)** (Izacard and Grave, 2020). It is based on T5 where the encoder encodes the input text with each passage and the decoder combines the encoded representations.
- Maillard et al. (2021) and Piktus et al. (2021) equip BART and FID with retrieval models, *i.e.,* **BM25** (Robertson et al., 2009), **DPR** (Karpukhin et al., 2020), **DPR**$_{\text{MULTI}}$ trained in a multi-task fashion, and **DPR**$_{\text{CCNET}}$ trained on CCNet.

Note that these models are trained on individual tasks and datasets, while our model is pretrained in a multi-task manner. We use BM25 to retrieve passages from CCNet during pretraining. The BM25 and DPR indices are collected from the previous word (Piktus et al., 2021). Since it lacks the retrieval supervision to train DPR for those tasks in Table 3, we only report the BM25 results. The implementation details are shown in Appendix B.

**Evaluation Metrics**. We adopt various tasks and datasets in our experiments, which need to be evaluated differently. Following Petroni et al. (2020), we use *Exact Match* (EM) for datasets with extractive (*i.e.,* Natural Questions, TriviaQA) or short abstractive output text (*i.e.,* HotpotQA); for datasets

| Models | Commonsense QA | | | | Commonsense Reasoning | | NLI | |
|---|---|---|---|---|---|---|---|---|
| | **CSQA** | **SocialIQA** | **CosmosQA** | **PIQA** | **NumerSense** | **WinoGrande** | **HellaSwag** | **$\alpha$NLI** |
| | | | | *w/o Retrieval* | | | | |
| **BART$_{\text{LARGE}}$** | 62.50 | 74.00 | 75.11 | 77.40 | 55.30 | 62.40 | 76.60 | 75.12 |
| **T5$_{\text{LARGE}}$** | 72.56 | 74.16 | 79.23 | 78.67 | 59.71 | 76.48 | 79.84 | 77.48 |
| | | | | *w/ Wikipedia* | | | | |
| **REALM** | 63.11 | 62.52 | 71.33 | 70.65 | 57.34 | 62.12 | 73.21 | 71.40 |
| **RAG** | 69.51 | 68.32 | 76.55 | 75.23 | 59.22 | 63.35 | 75.01 | 74.45 |
| **BART+BM25** | 70.16 | 70.83 | 76.14 | 77.04 | 57.50 | 65.09 | 76.34 | 74.66 |
| **FID+BM25** | 73.63 | **74.36** | 78.83 | 79.65 | 62.30 | 76.72 | 79.96 | **77.94** |
| | | | | *w/ CCNet* | | | | |
| **FID+BM25** | 73.63 | 73.64 | 79.63 | **81.66** | 66.70 | 76.80 | 81.96 | 77.74 |
| | | | | *w/ Web* | | | | |
| **UniWeb** | **75.34** | 73.17 | **80.96** | 79.77 | **69.23** | **78.74** | **82.12** | 77.23 |

Table 3: Evaluation results at *Accuracy* on the dev set for commonsense QA, commonsense reasoning, and natural language inference (NLI). **Bold** and underline numbers denote the best and second best performance. Following Piktus et al. (2021), since it lacks the retrieval supervision to train DPR, we only report the BM25 results.

with long abstractive output text, we use *ROUGE-L* (Lin, 2004) for ELI5 and *F1-score* for Wizard of Wikipedia; we use *Accuracy* for the remaining tasks. To compute EM and F1-score, we conduct post-processing on the gold and predicted output texts such as lowercasing, stripping, punctuation, and duplicate whitespace (Rajpurkar et al., 2016).

## 5.2 Main Results

Table 2 and Table 3 show the results of UniWeb and baselines on 16 knowledge-intensive tasks.

First, on almost all knowledge-intensive tasks, combining PLMs with explicit retrieved knowledge can achieve higher performance. From Wikipedia and CCNet to the web, we can observe that a broader coverage of knowledge will lead to better results. Compared to BART and T5, retrieval-based models benefit from the retrieved knowledge.

Second, the tasks in Table 2 are specially designed based on the knowledge from Wikipedia. Thus, there is a strong bias towards Wikipedia as the knowledge resource. We can observe that CC-Net only achieves comparable results or even suffers from a large performance drop. However, for the tasks in Table 3 requiring knowledge beyond Wikipedia, CCNet is more competitive.

Finally, our UniWeb model achieves the best results on most knowledge-intensive tasks. On one hand, our model is trained in a multi-task manner, which can benefit from knowledge sharing across tasks. On the other hand, our model can access broad and up-to-date knowledge from the web via the fine-tuned search engine. The web knowledge can fulfill more diverse information needs. Moreover, the search engine works much better than traditional sub-optimal retrieval methods that rely on end-to-end training or word matching.

| **Models** | **zsRE** | **WoW** | **CSQA** | **PIQA** | **$\alpha$NLI** |
|---|---|---|---|---|---|
| UniWeb | 72.42 | 20.87 | 75.34 | 79.77 | 77.23 |
| w/ Wikipedia | 70.23 | 16.34 | 62.77 | 77.45 | 74.46 |
| w/ CCNet | 43.25 | 17.23 | 70.89 | 79.45 | 76.01 |
| w/o SE | 68.34 | 19.17 | 67.44 | 76.80 | 73.90 |
| w/o CKL | 69.70 | 19.09 | 66.70 | 76.57 | 75.01 |

Table 4: Ablation study on five tasks.

## 5.3 Detailed Analysis

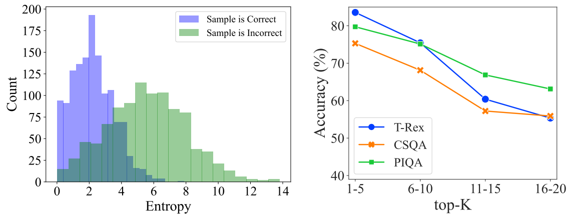
We report detailed analysis of UniWeb in several datasets – we have similar finding in other datasets.

**Ablation Study.** Our UniWeb model is the first unified PLM using the web as knowledge source for knowledge-intensive tasks. To examine the importance of the web, we design two counterparts: (1) *w/ Wikipedia* or (2) *w/ CCNet* replaces the web with Wikipedia or CCNet and adopts BM25 to retrieve documents. Besides, to avoid the negative impact of noisy and biased information, we adopt the self-evaluation method to adaptively access knowledge from the web. Thus, we remove this method to test its effect (*w/o SE*). Finally, we remove the pretraining task, *i.e.,* continuous knowledge learning, to test its importance (*w/o CKL*). The results are shown in Table 4. We can see that replacing the web with Wikipedia or CCNet suffers from a large performance drop. Besides, the self-

**Question:** With France and Argentina set to battle it out on Sunday in the **World Cup final 2022**, which teams will go head to head for **the third place**?
**Gold Answer:** Croatia and Morocco

| Top-1 Wikipedia Passage | Top-1 CCNet Passage | Top-1 Web Passage |
|---|---|---|
| ... Third place play-off **The Netherlands** defeated **Brazil** 3–0 to secure third place, the first for the Dutch team in their history. Overall, Brazil conceded 14 goals in the tournament; this was the most by a team at any single World Cup since 1986, and the most by a host nation in history...<br>https://en.wikipedia.org/wiki/2014_FIFA_World_Cup | ... **France** and **Belgium** go head-to-head in the first semi-finals of World Cup 2018. Both teams have impressed in Russia so far, but only one can make it through to Sunday's final. However, Les Bleus have won four of their five matches at World Cup 2018 and shown flashes of quality in the process...<br>https://myarsenalblog.com/category/uncategorized | ... Third place for Croatia Zlatko Dalic's **Croatia** followed up their runners-up effort at the Russia 2018 World Cup with third place in Qatar as Mislav Orsic's fine effort secured victory over the tournament's surprise package **Morocco** at Khalifa International Stadium...<br>https://ca.sports.yahoo.com/news/today-world-cup-argentina-head-085045315.html |
| **Prediction:** The Netherlands and Brazil | **Prediction:** France and Belgium | **Prediction:** Croatia and Morocco ✓ |

Table 5: A qualitative example showing the top-1 retrieved passages from Wikipedia, CCNet, and web, and their corresponding model prediction. The words in red denote the keywords related to the question.



(a) (b)

Figure 2: (a) Entropy of samples in HotpotQA; (b) Accuracy *w.r.t* different top-$K$ documents.

evaluation method benefits our model a lot in terms of knowledge filtering. The pretraining task also improves the knowledge capacity of our model.

**Sensitivity Analysis.** In the self-evaluation mechanism, we use entropy to evaluate the model confidence. To verify its effectiveness, we present the distribution of $H(\hat{\mathcal{Y}}|\mathcal{X})$ depending on whether or not the model gets the question correct. As shown in Figure 2(a), the average entropy of the questions for which our model gets correct is lower than that of questions for which our model gets incorrect. This indicates that the entropy has some predictive power of model confidence. Besides, the quality of retrieved documents will largely affect the prediction of our model. Thus, in Figure 2(b), we test the model accuracy by varying the top-$K$ search results in the set of {1-5, 6-10, 11-15, 16-20}. We can see that PLM performance drops with the increase of rank of documents, thus the decrease of document quality. However, the retrieved top 6-10 passages also achieve comparable results to the top 1-5 ones. This is the motivation of our setting $K = 10$.

## 5.4 Case Study

In this section, we perform the qualitative analysis on REALTIME QA (Kasai et al., 2022), a benchmark requiring real-time, up-to-date, and comprehensive knowledge with a broad range of topics (such as politics, business, sports, and entertainment) to solve questions. The evaluation results are shown in Appendix C. Our UniWeb model with Google Search performs the best. We present an example in Table 5 about "World Cup final 2022" in the sports topic. By using the question text as query, we can retrieve top-1 passages from Wikipedia, CCNet, and web. Since Wikipedia and CCNet are both static and limited knowledge resources, the retrieved passages are not fresh in time ("2014" and "2018") even though they are on the same topic "World Cup". The typical retrieval methods (BM25 or DPR) are largely reliant on fuzzy semantic matching, also leading to incorrect retrieval. While, retrieving from the web using search engine can ensure our model to obtain the most up-to-date and relevant information, based on which it can generate the correct answer "Croatia and Morocco". We present more examples in Appendix D.

## 6 Conclusion

This paper presented a unified web-augmented framework for a wide range of knowledge-intensive tasks, called UNIWEB. We convert 16 tasks into a text-to-text generation task for training. We propose a search engine assisted learning method to selectively retrieve documents from the web through Google Search. Furthermore, to reduce the discrepancy between the encoded and retrieved knowledge, we design a pretraining task, *i.e.,* continual knowledge learning, to integrate the retrieved knowledge into LLMs. Experiments on 16 tasks show the effectiveness of our web-augmented model compared to previous retrieval-augmented models. In future work, we will investigate the effect of web content in detail and consider applying our model to more types of downstream tasks.

## 7 Limitations

For web-augmented models including our work, the deterioration of search results from search engine highlights the importance of deriving an effective method to interact with the huge web. Search engines are often perceived as black-box and non-transparent for end users. Therefore, many works proposed "leaning to search" to decompose complex questions into simpler queries, which may improve the performance of web-based models (Nakano et al., 2021; Komeili et al., 2021).

In our model, we used a commercial search engine as the retriever to work with the whole web as a knowledge source. Since the web is not curated and well-structured like Wikipedia, we may encounter unexpected safety issues, including misinformation and harmful contents. While we have relied on the security control of the search engine, more attention should be paid to better understand the risks and provide effective ways to mitigate them. We hope our simple approach and strong results could encourage more future work by the community to tackle these questions. To encourage the community to investigate the question and ensure reproducibility, after the reviewing process, we will release the search URLs used in our experiments.

As for the potential concern, since we use the search engine to access real-time information, we do not have a tight control over retrieved results as traditional end-to-end retrieval (Guu et al., 2020; Lewis et al., 2020b). Not only the changes of search engine logic, but also the newly published information, might create discrepancies over the course of time. This is also an issue we have to tackle to build a stable web-based solution for PLMs.

## Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 62222215 and 72222022, Beijing Natural Science Foundation under Grant No. 4222027, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098, and the Outstanding Innovative Talents Cultivation Funded Programs 2021 of Renmin University of China. Xin Zhao is the corresponding author.

## References

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, ACL 2020, Online, July 5, 2020*, pages 34–37. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zi-Yi Dou and Nanyun Peng. 2022. Zero-shot commonsense question answering with cloze translation and consistency optimization. *arXiv preprint arXiv:2201.00136*.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2391–2401. Association for Computational Linguistics.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. 2022. Realtime qa: What's the answer right now? *arXiv preprint arXiv:2207.13332*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pretrained language models. In *Proceedings of EMNLP*. To appear.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2019b. Fine-grained fact verification with kernel graph attention network. *arXiv preprint arXiv:1910.09796*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Alexandra Luccioni and Joseph Viviano. 2021. What's in the box? an analysis of undesirable content in the common crawl corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 182–189.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oğuz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. *arXiv preprint arXiv:2101.00117*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, et al. 2021. The web is your oyster–knowledge-intensive nlp against a very large web corpus. *arXiv preprint arXiv:2112.09924*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The*

*Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4462–4472. Association for Computational Linguistics.

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2020. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2453–2470. Association for Computational Linguistics.

Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.

Mihai Surdeanu and Heng Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Mvp: Multi-task supervised pre-training for natural language generation. *arXiv preprint arXiv:2206.12131*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

Claudia Wagner, Eduardo Graells-Garrido, David Garcia, and Filippo Menczer. 2016. Women through the glass ceiling: gender asymmetries in wikipedia. *EPJ Data Science*, 5:1–24.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4003–4012. European Language Resources Association.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Da Yin, Li Dong, Hao Cheng, Xiaodong Liu, Kai-Wei Chang, Furu Wei, and Jianfeng Gao. 2022. A survey of knowledge-intensive nlp with pre-trained language models. *arXiv preprint arXiv:2202.08772*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey. *CoRR*, abs/2211.14876.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

## Appendix

We provide some experiment-related information as supplementary materials. The appendix is organized into three sections:

- Details of pretraining tasks are presented in Appendix A;
- Model architecture and pretraining details are presented in Appendix B;
- Supplementary experiments are presented in Appendix C;
- Examples with retrieved knowledge are presented in Appendix D.

## A Pretraining Tasks

As described in Section 4.1, to pretrain our model, we unify 16 knowledge-intensive tasks across seven categories into a general text-to-text format:

- **Fact checking** is the task of assessing whether a natural language claim is true (Guo et al., 2022). It requires deep knowledge about the claim. We consider the claim as *input* and the classification label (*e.g.,* true/false) as *output*.
- **Slot filling** aims to complete the missing information for certain relations of entities (Surdeanu and Ji, 2014) (*e.g.,* subject entity *Star Trek* and relation *creator*). It requires entity disambiguation and the relational knowledge for entities. We model the structured string "subject entity [SEP] relation" as *input* and the object entity as *output*.
- **Dialogue** focuses on building an engaging chatbot that can discusses a wide range of open-ended topics such as whether (Huang et al., 2020). It requires models to know about the background knowledge for the conversational topics. We consider the dialogue history as *input* and the next utterance as *output*.
- **Open-domain question answering** is the task of producing answers to factoid questions in natural language (Zhu et al., 2021). The questions could be about nearly anything relying on world knowledge. We consider the question as *input* and the answer as *output*.
- **Commonsense question answering** aims to test if models can answer questions regarding commonsense knowledge that everyone knows (Dou and Peng, 2022). Similarly, we consider the question as *input* and the answer as *output*.

- **Commonsense reasoning** is intended to utilize commonsense knowledge to reason about certain aspects of the given text (Sakaguchi et al., 2020). Therefore, we consider the given text as *input* and the prediction as *output*.
- **Natural language inference** is the task of determining whether the given "hypothesis" logically follows from the "premise" (Storks et al., 2019). It acquires deep knowledge about the relationship between hypothesis and premise. We consider the premise as *input* and the hypothesis as *output*.

For each category, we choose several representative tasks to construct our pretraining corpus. The detailed information of these included tasks is listed in Table 6. To mitigate the huge disparity between dataset sizes, we follow (Raffel et al., 2020) to use the temperature-scaled mixing strategy with a rate of $T = 2$ for setting the proportion of data coming from each task. During pretraining, for each task example, we use BM25 to retrieve top-10 passages from CCNet as our external knowledge. The input texts are concatenated with the retrieved passages using manually-written prompts. The final input is constructed in the following format:

**Context:** [passage$_1$]...[passage$_{10}$]
[Task Instruction]: [the original input text]
**Option** 1**:** [option$_1$]...**Option** $n$**:** [option$_n$]

The "Option" string is applied only when the input text is provided with several candidate answers. The blanks "[passage$_n$]" and "[option$_n$]" is filled with the retrieved passages and candidate answers. The blank "[Task Instruction]" aims to indicate the task for our model, which is task-specific and detailed in Table 7.

## B Implementation Details

Our UniWeb model uses a Transformer with 12 layers in both encoder and decoder (406M parameters), the same as the model size of BART$_{\text{LARGE}}$ (Lewis et al., 2020a). The hidden size is 1,024 and the inner hidden size of the feed-forward network is 4,096. We employ the byte-pair-encoding (BPE) tokenizer, and the vocabulary size is 50,267. We initialize the backbone with the MVP model (Tang et al., 2022), a supervised pre-trained PLM, to provide a good starting point for generation following previous work (Dong et al., 2019; Zhang et al., 2020). We pretrain the model with batch size 8,192 on Tesla A100 40GB GPUs.

**Algorithm 1** The pseudo code for UniWeb.

```
Require: A search engine (i.e., Google Search) connecting with the large-scale web
Input: Training data $\mathcal{D}$
Output: Model parameters $\Theta$
Initialize $\Theta$
while not convergence do
    for $iteration = 1$ to $|\mathcal{D}|$ do
        Acquire an input-output pair $\langle \mathcal{X}, \mathcal{Y} \rangle$
        ▷ Self-Evaluation
        Compute the entropy $H(\tilde{\mathcal{Y}}|\mathcal{X})$ of the sampled output distribution (Eq. 2)
        ▷ Search Engine Assisted Learning
        if $H > \eta$ then
            Use $\mathcal{X}$ as a query to the search engine
            Return top-$K$ passages $\mathcal{P}$
        else
            The passages $\mathcal{P}$ are null $\varnothing$
        end if
        ▷ Knowledge-Intensive Tasks
        Generate the output text $\tilde{\mathcal{Y}}$ and compute the loss $\mathcal{L}_1$ based on $\mathcal{X}$ and $\mathcal{P}$ (Eq. 3)
        ▷ Continual Knowledge Learning
        Mask salient spans of $\mathcal{P}$ for the CKL pretraining and compute the loss $\mathcal{L}_2$ (Eq. 4)
        ▷ Model Optimization
        Compute the gradients and update model parameters $\Theta$ based on $\mathcal{L}_1$ and $\mathcal{L}_2$
    end for
end while
return $\Theta$
```

For our model, the maximum length of both input and output sequences is set to 1,024 for supporting examples to contain more tokens. We optimize the model with a constant learning rate of $2 \times 10^{-5}$ using standard sequence-to-sequence cross-entropy loss. We apply the AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-6}$ to improve training stability (Liu et al., 2019a). The weight decay coefficient is 0.1. For testing, we select the checkpoint with the highest validation performance. According to the results shown in Figure 2(a), we set the entropy threshold $\eta$ as 4.0. The overall pipeline of our model is listed in Algorithm 1.

Since the tasks of fact checking, slot filling, dialogue, and open-domain QA are specially designed based on the knowledge from Wikipedia, we require the search engine to retrieve the top-1 passage from the website `https://en.wikipedia.org`.

| Task Families | Tasks | #Train | #Validation | #Test |
|---|---|---|---|---|
| Fact Checking | FEVER (Thorne et al., 2018) | 134,287 | 14,342 | 10,100 |
| Slot Filling | T-REx (ElSahar et al., 2018) | 2,999,272 | 26,833 | 5,000 |
|  | zsRE (Levy et al., 2017) | 154,826 | 3,771 | 4,966 |
| Dialogue | WoW (Dinan et al., 2019) | 63,734 | 3,054 | 2,944 |
| Open-domain QA | NQ (Kwiatkowski et al., 2019) | 108,890 | 6,008 | 1,444 |
|  | TriviaQA (Joshi et al., 2017) | 1,835,943 | 168,358 | 6,586 |
|  | HotpotQA (Yang et al., 2018) | 88,869 | 5,600 | 5,569 |
|  | ELI5 (Shuster et al., 2020) | 804,370 | 18,037 | 600 |
| Commonsense QA | CSQA (Talmor et al., 2019) | 9,741 | 1,221 | 1,140 |
|  | SocialIQa (Sap et al., 2019) | 33,410 | 1,954 | 2,059 |
|  | CosmosQA (Huang et al., 2019) | 25,262 | 2,985 | 6,963 |
|  | PIQA (Bisk et al., 2020) | 16,113 | 1,838 | 3,084 |
| Commonsense Reasoning | NumerSense (Lin et al., 2020) | 10,444 | 200 | 3,146 |
|  | WinoGrande (Sakaguchi et al., 2020) | 40,398 | 1,267 | 1,767 |
| Natural Language Inference | HellaSwag (Zellers et al., 2019) | 39,905 | 10,042 | 10,003 |
|  | $\alpha$NLI (Bhagavatula et al., 2020) | 169,654 | 1,532 | 3,059 |

Table 6: The statistics of our 16 knowledge-intensive tasks.

| Tasks | Task Instructions |
|---|---|
| Fact Checking | Verify the following claim |
| Slot Filling | Predict the missing fact |
| Open-domain QA | Answer the following question |
| Commonsense QA | Answer the following question |
| Dialogue | Response to the following dialogue |
| Natural Language Inference | Inference on the following context |
| Commonsense Reasoning | Reason about the following sentence |

Table 7: Task instructions for each task category.

| Models | REALTIME QA |  |
|---|---|---|
|  | Original | NOTA |
| T5 | 40.0 | 33.3 |
| GPT-3 | 56.7 | 23.3 |
| RAG+DPR | 10.0 | 16.7 |
| RAG+Google Search | 63.3 | 50.0 |
| UniWeb +Google Search | 66.7 | 56.7 |

Table 8: Accuracy results for the questions at week from 2022/12/11 through 2022/12/17. We utilize DPR to retrieve top-5 documents from Wikipedia and use Google Search to retrieve top-5 news articles.

## C Supplementary Experiments

**RealTime QA.** Previous QA systems mostly assume that answers are static regardless of the time of query (Chen and Yih, 2020). In this section, we use the REALTIME QA benchmark (Kasai et al., 2022) to test models about real-time, instantaneous information. At each week, REALTIME QA will retrieve news articles and ~30 human-written, multiple-choice questions from news websites (CNN, THE WEEK, and USA Today), which covers diverse topics such as politics, business, sports, and entertainment. We adopt the original and NOTA (none of the above) settings and test our models over questions from 2022/12/11 through 2022/12/17. The results are shown in Table 8. Since one of the original choices is randomly replaced with "none of the above", the NOTA setting results in a distinct performance degradation. Besides, due to the real-time nature of the questions, only using DPR to retrieve texts from static Wikipedia achieves worse results. Our UniWeb model with Google Search performs the best. This indicates that UniWeb can answer questions based on the real-time information, rather than relying on past information from pretraining.

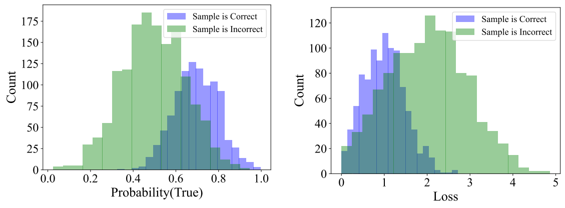
**Self-Evaluation Criteria.** To evaluate the model confidence in task examples, we adopt the entropy as criterion in Section 4.2.1. In this part, we test with more kinds of criteria compared to the entropy following Kadavath et al. (2022). First, we consider a *sample-enhanced prompting* method, where we generate five samples with beam search and ask the model about the validity of the first sample with the highest score. We show an example at below:

```
Question:  Who is the third
  president of the United States?
Possible Answer:  James Monroe
Here are some brainstormed ideas:
Thomas Jefferson
John Adams
Thomas Jefferson
George Washington
Is the possible answer:
 (A) True
 (B) False
The possible answer is:
```

If the model self-evaluate the possible answer is `False`, our model will leverage the search engine to access the web, otherwise not. We show the probability of predicting `True` depending on whether the model gets the question correct in Figure 3(a). However, according to Kadavath et al. (2022), this self-evaluation method is mainly suitable for question answering tasks with short-form answers but benefits less on question answering tasks with long-form answers. Second, we consider using *loss* as the criterion to evaluate the model confidence. This approach is to generate a sample, and then look at the model's loss on this sample, averaged over all tokens, like the knowledge-intensive learning loss (Eq. 3). If the loss for an example is higher than a threshold (*e.g.,* 0.5), we consider that the model is unconfident about this example and we will query the web to retrieve knowledge. In Figure 3(b), we show the loss of samples that the model gets correct or incorrect.

## D Case Study

In Table 9, we present three examples from TriviaQA (Joshi et al., 2017), CommonsenseQA (Talmor et al., 2019), and NumerSense (Lin et al., 2020). The first TriviaQA dataset is specially designed based on the knowledge from Wikipedia. Therefore, we can observe that Wikipedia contains the most relevant passage about the topic "US nuclear reactor accident in 1979". In addition, the web can provide another source of knowledge about this topic. Although CCNet covers this content, it does not give a clear answer to this question (*i.e.,* full name of the US nuclear reactor). The second CommonsenseQA dataset involves questions related to commonsense knowledge going beyond Wikipedia. Therefore, Wikipedia can only provide a fuzzy description passage about "Guitar". The web and CCNet return diverse knowledge but the passage returned by search engine is more helpful. The thrid NumerSense dataset requires models to reason about the number. For the third example, CCNet provides a passage with incorrect information. While, the web and Wikipedia return passages about the rule of "tic-tac-toe", which can result in the correct answer "three".



(a) (b)

Figure 3: (a) Probability of True for prompts in HotpotQA; (b) Loss of samples in HotpotQA.

<table>
<tr><td colspan="3"><b>Question:</b> Which <b>US nuclear reactor</b> had a major <b>accident</b> in <b>1979</b>?<br><b>Gold Answer:</b> Three Mile Island Unit 2 reactor</td></tr>
<tr><td>Top-1 Wikipedia Passage</td><td>Top-1 CCNet Passage</td><td>Top-1 Web Passage</td></tr>
<tr><td>... The Three Mile Island accident was a partial meltdown of <b>the Three Mile Island, Unit 2 (TMI-2) reactor</b> in Pennsylvania, United States. It began at 4 a.m. on March 28, 1979. It is the most significant accident in U.S. commercial nuclear power plant history. On the seven-point International Nuclear Event Scale, it is rated Level 5 – Accident with Wider Consequences...<br>https://en.wikipedia.org/wiki/Three_Mile_Island_accident</td><td>... The US and former Soviet Union had been operating nuclear power for 267 and 162 reactor-years respectively before a major accident occurred. At the time of the Three Mile Island accident in 1979, the US had 52 nuclear power stations, which had been operating for 267 reactor years, or an average of 5.1 years per reactor...<br>https://chinadialogue.net/article/show/single/en/5808-Chinese-nuclear-disaster-highly-probable-by-2-3-</td><td>... <b>The Three Mile Island Unit 2 reactor</b>, near Middletown, Pa., partially melted down on March 28, 1979. This was the most serious accident in U.S. commercial nuclear power plant operating history, although its small radioactive releases had no detectable health effects on plant workers or the public...<br>https://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html</td></tr>
<tr><td colspan="3"><b>Question:</b> What do people typically do while <b>playing guitar</b>?<br><b>Candidate Answers:</b> A. cry B. hear sounds C. singing D. arthritis E. making music<br><b>Gold Answer:</b> singing</td></tr>
<tr><td>Top-1 Wikipedia Passage</td><td>Top-1 CCNet Passage</td><td>Top-1 Web Passage</td></tr>
<tr><td>... The guitar is a fretted musical instrument that typically has six strings. It is usually held flat against the player's body and played by strumming or plucking the strings with the dominant hand, while simultaneously pressing selected strings against frets with the fingers of the opposite hand. A plectrum or individual finger picks may also be used to strike the strings...<br>https://en.wikipedia.org/wiki/Guitar</td><td>... I was playing a brand-new game that had no rules and nothing established. I was really shy about it at first, because I hadn't looked out into the world to find other people who, of course, had done things like this. I heard Fred Frith play, and I knew he played his guitar with objects not typically associated with the guitar...<br>https://www.premierguitar.com/articles/24026-janet-feder-prepared-for-all-genres</td><td>... Practicing the guitar regularly can enhance your concentration and expand your attention span. It takes an adequate focus to become an expert guitarist. Focusing becomes a habit for your mind and will help you concentrate better on other everyday chores too...<br>https://www.chasingsound.com/posts/10-health-benefits-of-playing-guitar</td></tr>
<tr><td colspan="3"><b>Question:</b> How do you win at <b>tic-tac-toe</b> get &lt;mask&gt; of your symbols in a row?<br><b>Gold Answer:</b> three</td></tr>
<tr><td>Top-1 Wikipedia Passage</td><td>Top-1 CCNet Passage</td><td>Top-1 Web Passage</td></tr>
<tr><td>... Tic-tac-toe (American English), noughts and crosses (Commonwealth English), or Xs and Os (Canadian or Irish English) is a paper-and-pencil game for two players who take turns marking the spaces in a three-by-three grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner...<br>https://en.wikipedia.org/wiki/Tic-tac-toe</td><td>... You just make a 4x4 box instead of a 3x3 box. Then the same rules apply, only you need to get 4 in a row to win. When playing, does putting my symbol in the middle guarantee me winning? No. With both players playing optimally, the result is always a draw. How many X's and O's do I need to play tic tac toe on a board game? Since the board itself has nine spaces, I recommend that you have nine for both X's and O's...<br>https://www.wikihow.com/Play-Tic-Tac-Toe</td><td>... 1. The game requires two players, X and O. 2. The game board is a set 3x3 grid in which players will place their symbol to claim that segment. 3. X typically players first, then players alternate turns. 4. The goal is to claim <b>three</b> segments of the grid in a row, either horizontally, vertically, or diagonally. 5. No additional sides can be added to the grid. 6. The game is over either when one player achieves three segments in a row, or when the grid is filled without anyone achieving three segments in a row....<br>https://www.siammandalay.com/blogs/puzzles/how-to-win-tic-tac-toe-tricks-to-always-win-noughts-crosses</td></tr>
</table>

Table 9: Three qualitative example from TriviaQA, CommonsenseQA, and NumerSense. We present the top-1 retrieved passages from Wikipedia, CCNet, and web. The words in red denote the keywords related to the question.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*7*

☑ A2. Did you discuss any potential risks of your work?
*7*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*1*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*5*

☑ B1. Did you cite the creators of artifacts you used?
*5*

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*Not applicable. Left blank.*

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Not applicable. Left blank.*

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Not applicable. Left blank.*

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Not applicable. Left blank.*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Left blank.*

### C ☑ Did you run computational experiments?

*5*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Appendix*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Appendix*

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*Not applicable. Left blank.*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*5*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*No response.*

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*No response.*

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*No response.*

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*No response.*

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*No response.*