Internet-NLP: Allowing Natural Language Processing Models to Connect to the Internet

Thamognya Kodi
Ruamrudee International School
contact@thamognya.com

Abstract

In this paper, I present **Internet-NLP** a new control-flow wrapper abstraction to enable the utilization of data from the internet (or a knowledge-database when offline) for existing context-needing Natural Lnaguage Processing (NLP) models to function without any given context. Internet-NLP can be used, finetuned alongside existing NLP models via its config settings and additionally its Long Short Term Memory neural network (LSTM neural network) can also be trained. Additionally incorporations of Masked Language Models (MLM) such as BERT, or LinkBERT (**??**) can be utilized to improve search queries, and therfore retrieve more accurate and reliable data. Futhermore, **Internet-NLP** utilizes a LSTM, Reinforcement Learning and caches to allow for multi-turn NLP tasks, and improvement via Reinforcement Learning from user.

Additionally in this paper, I also present new NLP and Natural Language Inference (NLI) models to assist **Internet-NLP**:

- Open-book question and long answer (QA) via GPT-NeoX-20b (**??**)
- CrossEncoder NLI via LinkBERT (**???**)
- Answer to context NLP via T5 (**?**)

Along with these models, I also present new general purpose QA and NLI datasets:

- ALotNLI made from ANLI, MultiNLI, and SNLI (**???**)
- ALotOpenBookQA made from CoQA, Natural Questions, and SQuAD (**???**)

As a result of these models, datasets, and Internet-NLP, the accuracy and reliability of most context-needing NLP models on most NLP tasks, especially tasks that require more factual responses with no given context increased.

Internet-NLP and the new NLP and NLI models, which were trained on the general-purpose datasets (ALotNLI, and ALotOpenBookQA). Internet-NLP, by default utilizes an Text-Generative model GPT-NeoX (**??**) for long responses and LinkBERT (**?**) for short responses. For 2 choices (for ex: True and False) Bi-Encoder NLI has been used and for multiple choices CrossEncoder will be used (**?**).

Internet-NLP, in layperson terms, provides the context for context-needing NLP models to let them function. Internet-NLP can be improved via finetuning, and training of LSTM and Reinforcement Learning model (which can be trained alongside the NLP model), which enables for better search queries, and subsequently results. It obtains state-of-the-art results in QA and NLI without context.

Internet-NLP is a subset of a larger package, Internet-ML and is open-source. [1]. Old versions of Internet-NLP is also publicly available. [2].

1 Introduction

There are currently two main solutions for utilizing NLP tasks with no context provided:

1. Large Pre-Trained Text2Text-Generation Model
   - Pre-trained Text2Text-generation models, like T5 (**?**) that have open-domain question-answering closed-book (no context) language models (ODQA LM) capabilities (**?**). These closed-book QDQA LMs are comparatively state-of-the-art performance in many no-context NLP tasks, mainly question-answering. Text2Text-generation models for such no-context NLP tasks are usually large, slow, and have a low accuracy (**?**).
   - Example: T5 (**?**)
   - Illustration of how ODQA LM work: **??**
2. Large Knowledge Base with a Contex-Needing Language Model

[1] Internet-NLP, subset of Internet-ML is public, and open-source: https://github.com/thamognya/internet_ml

[2] Old Versions of Internet-NLP is public: https://pypi.org/project/internet-nlp/

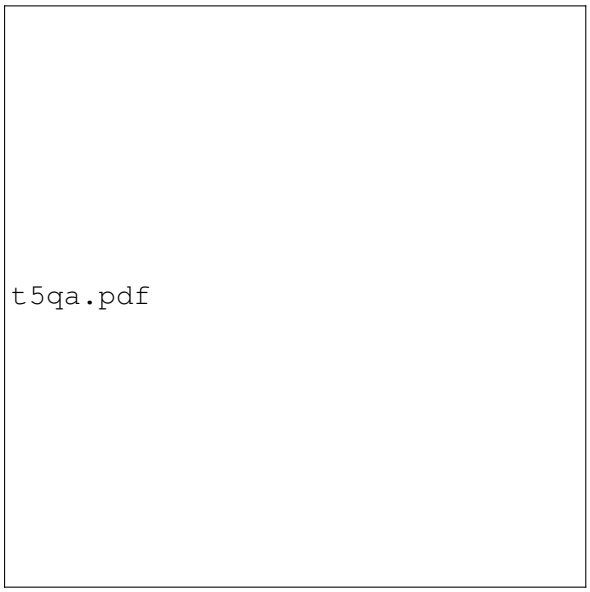

Figure 1: This is an illustration of how ODQA LMs work (**?**).

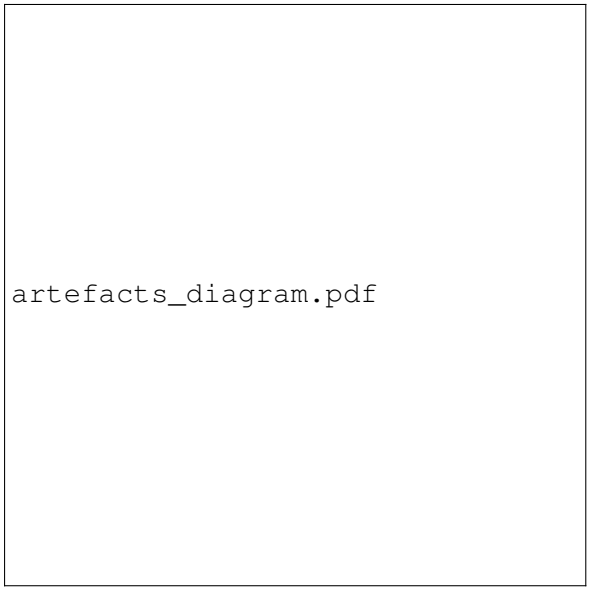

Figure 2: This is an illustration of how LMs with a knowledge base and artifact retriver work (**?**).

- Large Knowledge base with an pre-trained open-book LM and retriever, provides an comparatively higher performance, accuracy and the model itself is small. These models however require — usually — a large knowledge base which makes the overall solution large, but still fast and with more accuracy on the field the knowledge base specalizes in.
- Example: LinkBERT (**?**) with an artifact retriver (**?**) with a knowledge base such as DBpedia or WikiData (**??**)
- Illustration of how LMs with a knowledge base and artifact retriver work: **??**

Solution **??** and **??** achieve the same end goal of NLP tasks without context via two different methods; these current solutions restrict NLP tasks and accuracy without context, especially for more open-domain tasks. The major limitation in this case would be accuracy, efficency and size of models and their knowledge base which then limit the use cases of closed-book open-domain NLP tasks.

In this paper, I propose Internet-NLP, an direct improvement to solution **??** which allows NLP models to not require a large knowledge base (altough you can configure Internet-NLP to utilize a knowledge base) that incoproates the internet's vast knowledge along utilizing data in hyperlinks in webpages (**?**) to create a more resource-filled data for our existing or future context-needing pre-trained model to use for NLP tasks. Internet-NLP encompasses pre-trained NLP and NLI models, along with its web-data-scraper creates an small temporary on-basis data and a cache for NLP tasks to be performed without given context.

Utilizing the vast data on the internet, graph of documents as corpus (**?**) allows us to enable to reduce our solution size, increase efficency and increase accuracy. Additionally unlike usage of static data, Internet-NLP utilizes the dynamic, and frequent updating data of the internet which enables us to utilize any type of NLP model along with NLI models to allow us to follow a sequence of control flow to get the context for context-needing models. This approach utilizes a combination of data-collection (**?**) for NLPs with context-needing open-domain NLP to gain more accurate results in most no-context NLP tasks.

Additionally Internet-NLP's Text2Text-generation search query model: T5 (**?**) and LSTM noun remembrance using parts of speech tagging (**?**) on ALotClosedBookQA with it improving

search queries based on the difference on answer recieved and the answer from datasets, using parts of speech tagging on answers (**?**).