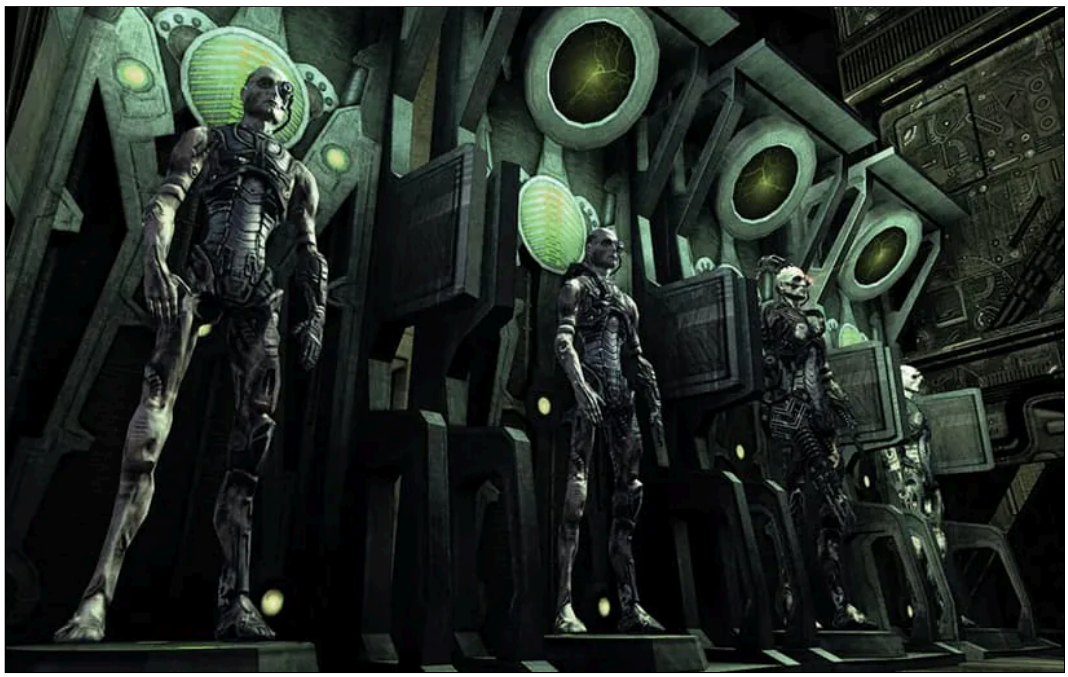
Creating Artificial Mechanical Turks With Pretrained Language Models

Originally published at https://www.unite.ai/creating-artificial-mechanical-turks-with-pretrained-language-models/



A large part of the development of machine learning systems depends on labeling of data, where hundreds, even thousands of questions (such as *Is this a picture of a cat?* and *Is this text offensive?*) must be settled in order to develop authoritative datasets on which AI systems will be trained.

Though we all contribute to this process at some point, the majority of these labeling tasks are performed for money by human workers at frameworks such as Amazon Mechanical Turk, where annotators complete minor classification tasks in a piece-work economy.

Model development would be cheaper if pretrained language models (PLMs) could in themselves undertake some of the more basic Human Intelligence Tasks (HITs) currently being crowdsourced at AMT and similar platforms.

Recent research from Germany and Huawei proposes this, in the paper *LMTurk: Few-Shot Learners as Crowdsourcing Workers*.

Language Models Performing Few-Shot Learning

The authors suggest that the simpler strata of tasks typically aimed at (human) Turk workers are analogous to few-shot learning, where an automated framework has to decide a mini-task based on a small number of examples given to it.

They therefore propose that AI systems can learn effectively from existing PLMs that were originally trained by crowdworkers – that the core knowledge imparted from people to machines has effectively been accomplished already, and that where such knowledge is relatively immutable or empirical in some way, automated language model frameworks can potentially perform these tasks in themselves.

> *'Our basic idea is that, for an NLP task T, we treat few-shot learners as non-expert workers, resembling crowdsourcing workers that annotate resources for human language technology. We are inspired by the fact that we can view a crowdsourcing worker as a type of few-shot learner.'*

The implications include the possibility that many of the ground truths that AI systems of the future depend upon will have been derived from humans quite some years earlier, thereafter treated as pre-validated and exploitable information that no longer requires human intervention.

Jobs for Mid-Range, Semi-performant Language Models

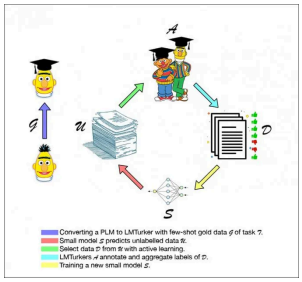
Besides the motivation to cut the cost of humans-in-the-loop, the researchers suggest that using 'mid-range' PLMs as *truly* Mechanical Turks provides useful work for these 'also ran' systems, which are increasingly being overshadowed by headline-grabbing, hyperscale and costly language models such as GPT-3, which are too expensive and over-specced for such tasks.

> *'Our goal in this paper is to devise methods that make more effective use of current few-shot learners. This is crucial because an increasing number of gigantic few-shot learners are trained; how to use them effectively is thus an important question. In particular, we want an alternative to hard-to-deploy huge models.*
>
> *'At the same time, we want to take full advantage of the PLMs' strengths: Their versatility ensures wide applicability across tasks; their vast store of knowledge about language and the world (learned in pretraining) manifests in the data efficiency of few-shot learners, reducing labor and time consumption in data annotation.'*

To date, the authors argue, few-shot learners in NLP have been treated as disposable interstitial stages on the road to high-level natural language systems that are far more resource intensive, and that such work has been undertaken abstractly and without consideration for the possible utility of these systems.

Method

The authors' offer *LMTurk* (Language Model as mechanical Turk), in a workflow where input from this automated HIT provides labels for a mid-level NLP model.

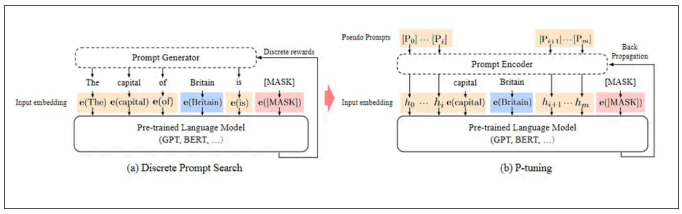


A basic concept model for LMTurk. Source: https://arxiv.org/pdf/2112.07522.pdf

This first iteration relies on few-shot human-labeled 'gold' data, where meatware Turks have annotated labels for a limited number of tasks, and the labels have been scored well, either via direct human oversight or through consensus voting. The implication for this schema is that forks or developments from this human-grounded starting point might not need additional human input down the road.

Though the authors suggest further experiments with later hybrid models (where human input would be present, but greatly reduced), they did not, for the purposes of their research, pit LMTurk models against equivalent results from human-generated HIT workers, considering that the gold-labeled data is itself 'human input'.

The PLM designed to perform Turk operations was adapted for the task by P-Tuning, a method published by researchers from China in 2021, which proposed trainable continuous prompt embeddings to improve the performance of GPT-3-style models on Natural Language Understanding (NLU) tasks.



P-Tuning attempts to deepen a GPT-style model's predictive power, and its appearance of conceptual understanding of language, by incorporating embedded pseudo-prompts. In this case, the start query is 'The capital of Britain is a [x]'. Source: https://arxiv.org/pdf/2103.10385.pdf

Data and Architecture

LMTurk was evaluated on five datasets: two from the Stanford Sentiment Treebank; AG's News Corpus; Recognizing Textual Entailment (RTE); and Corpus of Linguistic Acceptability (CoLA).

For its larger model, LMTurk uses the publicly available PLMs ALBERT-XXLarge-v2 (AXLV2) as the source model for conversion into an automated Turk. The model features 223 million parameters (as opposed to the 175 billion parameters in GPT-3). AXLV2, the authors observe, has proven itself capable of outperforming higher scale models such as 334M BERT-Large.

For a more agile, lightweight and edge-deployable model, the project uses TinyBERT-General-4L-312D (TBG), which features 14.5 million parameters with performance comparable to BERT-base (which has 110 million parameters).

Prompt-enabled training took place on PyTorch and HuggingFace for AXLV2 over 100 batch steps at a batch size of 13, on a learning rate of 5e-4, using linear decay. Each experiment was originated with three different random seeds.

Results

The LMTurk project runs diverse models against so many specific sub-sectors of NLP that the complex results of the researchers' experiments are not easy to reduce down to empirical evidence that LMTurk offers in itself a viable approach to re-use of historical, human-originated HIT-style few shot learning scenarios.

However, for evaluation purposes, the authors compare their method to two prior works: *Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference* by German researchers Timo Schick and Hinrich Schutze; and results from *Prompt-Based Auto*, featured in *Making Pre-trained Language Models Better Few-shot Learners* by Gao, Chen and Fisch (respectively from Princeton and MIT).

	Schick and Schütze (2021a,b)	Gao et al. (2021)	Ours
SST2	n/a	93.0±0.6	93.08±0.62
SST5	n/a	49.5±1.7	46.70±0.93
RTE	69.8	71.1±5.3	70.88±1.70
AGN.	86.3±0.0	n/a	87.71±0.07
CoLA	n/a	21.8±15.9	19.71±1.89

Results from the LMTurk experiments, with the researchers reporting 'comparable' performance.

In short, LMTurk offers a relatively promising line-of-inquiry for researchers seeking to embed and enshrine gold-labeled human-originated data into evolving, mid-complexity language models where automated systems stand in for human input.

As with the relatively small amount of prior work in this field, the central concept relies on the immutability of the original human data, and the presumption that temporal factors – which can represent significant roadblocks to NLP development – will not require further human intervention as the machine-only lineage evolves.

Originally published 30th December 2022

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More