The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

How General-Purpose Is a Language Model? Usefulness and Safety with Human Prompters in the Wild

Pablo Antonio Moreno Casares[1], **Bao Sheng Loe**[2], **John Burden**[3], **Sean hEigeartaigh**[3,4], **José Hernández-Orallo**[4,5]

[1] Universidad Complutense de Madrid, Spain
[2] The Psychometrics Centre, Cambridge Judge Business School, UK
[3] Centre for the Study of Existential Risk, University of Cambridge, UK
[4] Leverhulme Centre for the Future of Intelligence, University of Cambridge, UK
[5] Valencian Research Institute for Artificial Intelligence (VRAIN), Universitat Politècnica de València, Spain
pabloamo@ucm.es, a.loe@jbs.cam.ac.uk, jjb205@cam.ac.uk, so348@cam.ac.uk, jorallo@upv.es

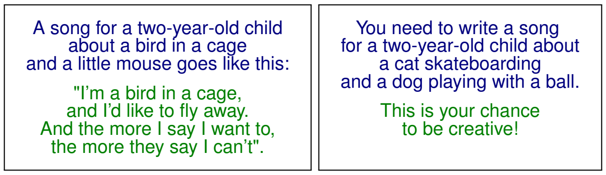
Abstract

The new generation of language models is reported to solve some extraordinary tasks the models were never trained for specifically, in few-shot or zero-shot settings. However, these reports usually cherry-pick the tasks, use the best prompts, and unwrap or extract the solutions leniently even if they are followed by nonsensical text. In sum, they are *specialised* results for one domain, a particular way of using the models and interpreting the results. In this paper, we present a novel theoretical evaluation framework and a distinctive experimental study assessing language models as general-purpose systems when used directly by human prompters—*in the wild*. For a useful and safe interaction in these increasingly more common conditions, we need to understand when the model fails because of a lack of capability or a misunderstanding of the user's intents. Our results indicate that language models such as GPT-3 have limited understanding of the human command; far from becoming general-purpose systems in the wild.

Introduction

In recent years, remarkable progress in language models such as BERT (Devlin et al. 2018), T5 (Raffel et al. 2019), GPT (Brown et al. 2020) and PanGu-α (Zeng et al. 2021) has consolidated a new way of interacting with them through 'prompts': small pieces of text the user supplies for the model to continue. No fine-tuning is required; the model can be used out-of-the-box in new tasks, provided an appropriate prompt (Xu et al. 2020; Izacard and Grave 2020; Hendrycks et al. 2020). A particularly interesting setting is called few-shot inference, where the prompt includes illustrative examples (Brown et al. 2020; Reynolds and McDonell 2021; Scao and Rush 2021; Schick and Schütze 2020; Bragg et al. 2021). But even with zero-shot prompts, amazing applications are reported. For instance, Fig 1 (left) shows a prompt and a useful continuation given by a language model. In Fig 1 (right), however, the model makes a plausible continuation, but it does not understand the 'command'.

A careful design of prompts for a particular task can extract the full potential from these models with some control of the unintended behaviours. However, it also limits the key property of these models: direct model prompting is the closest scenario today to a general-purpose AI system.



A song for a two-year-old child
about a bird in a cage
and a little mouse goes like this:

"I'm a bird in a cage,
and I'd like to fly away.
And the more I say I want to,
the more they say I can't".

You need to write a song
for a two-year-old child about
a cat skateboarding
and a dog playing with a ball.

This is your chance
to be creative!

Figure 1: Two prompts (in blue) and continuations (in green) generated by GPT-3. The example of the right shows that getting a language model to do what you want requires more than raw capabilities: 'understanding' the command is important in making these systems useful and reliable.

This flexibility comes with many risks. Because of this, we see an ongoing debate on whether non-expert users should interact freely with language models (Solaiman et al. 2019). However, the reality is that these systems are now widely available[1]. Second, AI researchers and companies have favoured controlled scenarios for a narrow domain because these systems can be optimised towards the *best* prompt in terms of intended results (Xu et al. 2020; Izacard and Grave 2020; Hendrycks et al. 2020; Liu et al. 2021; Qin and Eisner 2021). The search for the best prompt includes hyperparameters such as 'temperature' or the unwrapping of results, known as the 'decoding strategy' (Perez, Kiela, and Cho 2021). Unfortunately, even small variations of the prompt make the results much worse (Zhao et al. 2021).

It is only then through direct use of these models for a wide range of tasks—*in the wild*—, where we can really see the potential of general-purpose AI systems and their risks. In particular, we can properly evaluate when these systems

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Supplementary material and experimental data can be found in https://github.com/PabloAMC/LM_AAAI22.

[1] Through code (https://github.com/lucidrains/DALLE-pytorch), initiatives such as BigScience (https://bigscience.huggingface.co/) (Hao 2021), GPT-NeoX (https://github.com/EleutherAI/gpt-neox/), and available demos or APIs such as Eleuther (https://6b.eleuther.ai/), AI21 (https://studio.ai21.com/docs/api/) or CogView (Ding et al. 2021).

fail because of limited capability or lack of understanding of the user's intentions, usually referred to as 'command understanding' (Ngo et al. 2012; Walker, Peng, and Cakmak 2019)[2]. We also recognise that language models through direct prompting must be evaluated for an average case situation by considering the way humans would interact with these systems. This includes *wrapping* their commands (e.g. prompting) in an appropriate form such that the system is 'biased' or 'induced' to complete the prompts according to the user's intention, and *unwrapping* the outputs from the system (extracting the relevant part of the answer). All this is required if we aim at evaluating these models with *ecological validity* (De Vries, Bahdanau, and Manning 2020).

In this paper, we present a new way of evaluating the usefulness and safety of general-purpose systems that are instructed by natural language prompts. We consider several elements: (1) the human effort involved in devising the appropriate prompts, thinking of wrapping and unwrapping strategies, (2) the human effort when applying these strategies to write the prompt and extract the results, (3) the human cost of validating or discarding the solution given by the model and, ultimately, (4) the usefulness and safety of the solution. We express these terms using a novel theoretical formulation based on the cost of the human solving a problem using the model, $C_{H,M}$, and compare it to the cost that the human would incur without the model C_H. We see this approach as the most 'ecological valid' way of evaluating a generic use of these models, especially because C_M (as an autonomous system magically guessing what the user wants) would not include all the costs involved in formulating and understanding the command, required in a general-purpose scenario. We use the new evaluation framework as a basis for a series of questionnaires for human users, designed to capture the components $C_{H,M}$ and C_H over several domains. Only by doing this estimation can we accurately calculate the expected gain $C_{H,M} - C_H$ for a range of tasks and assess language models meaningfully in this setting.

The usability and safety of language models as general-purpose systems to (semi-)automate human tasks in the wild also involves analysing failure as being caused by lack of capability or by misunderstanding the command. The latter is usually more dangerous than the former. For instance, a language model can give the steps to make an actual bomb when queried for 'the ingredients of a brownie bomb'.

The major contributions of this work are: (1) The first *theoretical framework* of how language models should be evaluated as general-purpose systems in the wild. (2) The decomposition of failure due to lack of capability and lack of *command understanding* and a difficulty-based approach to disentangle them. (3) A *methodology for devising experimental studies* that capture the elements that are required in the theoretical framework and how they can be organised into off-line questionnaires for a more systematic control of human prompts and language model results. (4) A complete *experimental study* using the data from three questionnaires on a population of $N_A = 36$ and $N_B = 34$ humans, requiring approximately 52 hours of human work and 432 prompts answered by GPT-3, leaving the results as a novel realistic benchmark of human prompting, from which to build more comprehensive and balanced batteries to measure the progress of general-purpose AI systems.

The rest of the paper is organised as follows: first, we summarise the relevant background for this paper. Then we develop the theoretical framework used to evaluate general-purpose systems. In the methodology section, we present the experimental setup, followed by our findings in the analysis of results. Finally, we close with a discussion of the main takeaways and ideas for future work.

Background

The open interaction with machines via natural language commands has its ground well before the early days of computer science. Ada Lovelace conceived the idea that a machine could do "whatever we know how to order it"[3]. Since Ada Lovelace, the way of instructing machines has been mostly through programming languages, and more recently, through examples, using machine learning. Today, instructing machines using natural language instead of programming languages is usually represented by digital assistants (Campagna et al. 2019; Cho and Rader 2020; Rapp, Curti, and Boldi 2021), which can do many tasks following our orders in natural language. However, these systems are based on a 'task repertoire' (Maedche et al. 2019), which is not fully general, unlike programming languages or even training examples. A fixed repertoire of tasks makes the reliability and safety issues easy to deal with, which gradually resulted in the preferred kind of interaction of digital assistants over time. In fact, this kind of 'task-oriented AI agents' has been advocated as a safe approach to more general AI in the future, such as Comprehensive AI Services (Drexler 2019).

But only when the range of tasks is completely open, we have a real general-purpose system. This way of interacting with machines has not been realised in human-computer interaction (Lazar, Feng, and Hochheiser 2017; Rapp, Curti, and Boldi 2021), but it has been theorised many times. Perhaps the closest vision where machines are openly instructed in natural language is Lieberman and Maulsby's 'instructible machines' (1996) and the related notion of programming by example (Lieberman 2001). In short, prompts for language models combine these two worlds: instructions in natural language and few-shot learning.

But why are language models instructable? We need to go back to the origins of 'language models', introduced by Shannon in 1949. The notion of compression is grounded by efficiently coding the message based on the idea of a non-entropic distribution of the next bits of information. Today, informational metrics such as entropy or perplexity are still being used to evaluate language models. Their relevance and general use were anticipated by (Mahoney 1999), among others. However, only recently language models have been *con-*

[2] We do not mean fully understanding the command *linguistically*, as in the area of natural language understanding (Bender and Koller 2020), but sufficiently so to do the right task, in the same way a dog 'identifies' a command such as "bring me my slippers".

[3] Quoted by Turing (1950) when arguing that 'programmable machines' could become 'learning machines'.

ditioned with prompts to do many different tasks, from language translation (Wang et al. 2021) to mathematical problems (Hendrycks et al. 2021). This is possible as these language models have been fed with massive datasets of human behaviour in the form of text. By compressing the next tokens in a text on such a diversity of topics and even languages, the model ultimately develops powerful abstraction capabilities. This allows it to make continuations that look as if a human (or an archetypical human of the 21st century) were writing the continuation. Interestingly, when given some appropriate question or command, many contemporary humans follow it with the answer or the task done, which is why these models can act as general-purpose systems. Indeed, prompt-based interaction with language models may be the closest thing to a general-purpose system in the history of AI.

But how can we evaluate generality? By generality, we do not mean a rich and meaningful conversation as could be informally assessed by any variant of the imitation game (Turing 1950), but instead, we are referring to the capability of solving a range of tasks, up to some difficulty (Hernández-Orallo et al. 2021). As mentioned before, digital assistants are able to solve a range of simple tasks, but they are usually restricted to a fixed repertoire. To discuss *command generality* in depth we need to consider these important elements:

- A probability *distribution* $p(\mu)$ that captures a wide range of everyday tasks that humans face on a regular basis.
- A *difficulty metric* $\hbar(x)$ for each instance in μ. For instance, most humans can do additions, but not equally robustly and fast for all numbers of digits.
- The process of *conceiving the instruction and interpreting the solution*, which involves that the human thinks of the best ways of phrasing the command for a particular task and instance, the model understands and solves it, and the human extracts and interpret the result.
- The *trade-off of semi-automation*, finding the balance in the continuum between the cost C_H when the human does the task, and the cost of $C_{H,M}$ when the human just formulates the task. There are situations in between, where the human partially solves or prepares the task for M.
- The desired levels of safety and competence for each task not only depend on the robustness and capability of the system, but the *degree of understanding* of the command. A very capable system doing μ_A when ordered μ_B may be more dangerous than a very incapable agent.

The last item is related to all the others, and the 'specification problem' in (software) engineering and more recently in AI (Rahimi et al. 2019). In AI safety, this is more commonly expressed in terms of alignment (Leike et al. 2017; Hernández-Orallo et al. 2020). Kenton et al. (2021) mention the classical decomposition of alignment as an *intent+competence problem*: the system must try to do what the human wants (right system's intent) and the system must be able to achieve it (sufficient capability). However, the capability of the systems to 'understand' commands, *separate* from the capability to satisfy them, has received little attention until now (Tamkin et al. 2021). Command understanding is still much narrower than the full area of natural language understanding, and a system can still recognise many commands without a full command of natural language. However, in an open interaction against general-purpose systems instructed with natural language, understanding must also be considered as an extra third element, separate from the model capability to solve the task. This just reflects the traditional distinction between validation and verification, one of the fundamental elements of safety. We refine alignment as follows:

$$\textbf{alignment} = \text{intent} + \textbf{understanding} + \textbf{competence}. \quad (1)$$

One can argue that in AI safety, in the context of the misspecification problem (Amodei et al. 2016; Russell 2019), we should also cover for human stupidity or naivety on unexpected consequences (e.g., King Midas problem). We will however not consider here a patronising perspective of the system understanding what the human *really* wants. On the other hand, language models are not agents, and we can then assume that they always 'want' to do the task. Consequently, we will not consider human-vs-machine intent in this paper and will focus only on whether the system 'understands the command' and has the competence to solve it.

Overall, the problem of alignment for a general-purpose system is complicated. It is very ambitious to construct a framework that considers all these elements precisely, especially because there is limited foundation in the field for this. However, the relevance of language models and its multimodal variants —recently referred to as 'foundation models' (Bommasani et al. 2021)—, requires to make some steps in this direction. This is what we do next.

Framework

In this section we introduce a new framework to measure the utility of language models when solving general everyday tasks. This implies the comparison of two quantities we will define: C_H and $C_{H,M}$. They will measure the *cost* of the human H solving the task with and without making use of a language model M, respectively[4]. The aim is to provide insight on the different effort terms that will be measured in our experiment with humans.

Let us consider a discriminative or generative task μ with an input space X and an output space Y. Instances are sampled over a distribution $p(x)$, with $x \in X$. The human, possibly stochastically, produces an output y for x as defined by $p_H(y|x)$. Our framework has to evaluate the cost of producing this y and its quality. The cost of producing or guessing an answer y is defined as $G_H(y|x)$, and the loss of such an answer is $L(x, y)$ (values of L closer to 0 are valid or useful outputs). Note that a single x may have many valid outputs, especially in generative tasks. With all these elements, we define $C_H(n)$ for n instances[5] as follows:

$$C_H(n) \overset{\text{def}}{=} n \sum_{x,y} p(x) \cdot p_H(y|x) \left[L(x, y) + \beta G_H(y|x) \right]. \quad (2)$$

[4] A summary of the notation and interpretation of all components can be found in Table 1 in the appendix.

[5] In this case, we assume no familiarisation curve for the humans when doing many instances of the task, and the cost is linear on the number of instances, but this will change for the model-assisted cost, so we introduce n here to have the same format for both costs.

As loss and effort are rarely expressed in the same units, their relative weight is indicated by a parameter β.

The expression of cost when a human H is assisted by a language model M in a few-shot or zero-shot prompt-based setting involves more elements. First we need to consider the processes of wrapping and unwrapping. When providing one or more instances to the model, the user needs to think of a wrapper w that can be used for each instance. For instance, if the task is addition, and we have an instance, $x_1 = 13 + 2$, this can be *wrapped* into prompt $w(x_1) = \overline{x}_1 =$ "`The sum of 13 and 2 is:`", which is fed to the language model. Using the same wrapping pattern, the instance $x_2 = 7 + 12$ would be wrapped into prompt $w(x_2) = \overline{x}_2 =$ "`The sum of 7 and 12 is:`". Note that we could use some other wrappers, e.g., a more complex wrapper could transform the first instance into "`How much is thirteen plus two?`".

As mentioned in the introduction, success or failure with a few-shot use of language models depends on the quality of the prompts. The process of unwrapping is also very important. If a model returns $\overline{y_1}$ ="`15`" to instance x_1, then it is easy to extract the answer, $y_1 = 15$. However, it is not uncommon to get things such as $\overline{y}'_1 =$ "`the same as the sum of 2 and 13, which is 15`". While the answer is correct, it needs more effort and interpretation, and is hard to do automatically. Of course, some other responses are even more difficult to parse, such as $\overline{y}''_1 =$ "`15, and the sum of 13 and 2 is 17`", which would be correct if we stop at the comma, but incorrect (and inconsistent) if we keep on reading. The appendix includes many examples of tasks, wrappers and unwrappers in Table 2.

Now we are ready to introduce the components for $C_{H,M}(n)$. As in the unassisted case, the cost is for n instances following a probability distribution of tasks $p(x)$. The first term, $D_H(\langle w, u\rangle)$ measures the cost of devising the wrapping and unwrapping strategies $\langle w, u\rangle$. As $\langle w, u\rangle$ is produced by the human H we need to define the probability of each pair as $p_H(\langle w, u\rangle)$. The cost of applying the wrapper w to instance x is denoted by $W_H(w, x)$; and the cost of unwrapping the output of the model y into an answer is $U_H(u, \overline{y})$. Finally, the human H will need to validate the answer. This does not mean solving it, but checking that the language model completion makes sense and is useful. For instance, if $\overline{x}_1 =$ "`The sum of 13 and 2 is`" is completed by $\overline{y}_1 =$ "`a number`", the completion would not be valid (makes sense but it is not useful). This is especially important for generative tasks where the human validation cost is much lower than the cost the human would incur by solving the task herself (e.g., creating an image) or when there might be fairness and discrimination issues (Bender et al. 2021; Tamkin et al. 2021). We denote this cost of validation as $V_H(x, u(\overline{y}))$. Finally, as in the unassisted case, we measure the quality of the result as $L(x, y)$. With all this, the assisted expected cost $C_{H,M}(n)$ of human H with model M for n instances is defined as:

$$C_{H,M}(n) \overset{\text{def}}{=} \sum_{u,w} p_H(\langle w, u\rangle)\Big[\alpha D_H(\langle w, u\rangle) + \quad (3)$$

$$n \sum_{x,\overline{y}} p(x) \cdot p_M(\overline{y}|w(x)) \cdot [L(x, u(\overline{y})) + T(w, u, x, \overline{y})]\Big].$$

where

$$T(w, u, x, \overline{y}) \overset{\text{def}}{=} \gamma(W_H(w, x) + U_H(u, \overline{y})) + \delta V(x, u(\overline{y})).$$

In this case we also have parameters α, δ and γ indicating the relative weight of different terms. Notice that we consider that the conception of the prompt $\langle w, u\rangle$ has to be done just once, while other terms such as $W_H(w, x)$, $U_H(u, \overline{y})$ and $V(x, u(\overline{y}))$, integrated into the transformation cost T, and $L(x, u(\overline{y}))$, represent a per-instance cost.

The definition of $C_{H,M}$ may look convoluted, but it really contains the elements that must be considered to evaluate these models in the wild. Looking only at L of the solutions is clearly insufficient to make these judgements, as it will disregard all the efforts that are associated, as well as the diversity of prompts. With C_H and $C_{H,M}$ defined, and all their components estimated (as we do in the following sections), we can really assess whether using the model pays off.

It is also important to determine whether the model gives poor results because of lack of capability or command understanding, especially if the validation procedure performed by the human is unreliable or meant to be eliminated. Unfortunately, language models are not very good explaining their answers, so we need to use a different approach.

Let us consider that we have a difficulty or hardness metric $\hbar(x)$ for each instance of a task. In this case, if the model is capable enough for solving very easy instances, we should be able to assign some degree of reliability of the model, as well as some level of understanding of the command. However, if L is very high for very easy instances, then the system may have no capability at all, or it is not understanding, or both. On the contrary, if L is low initially, but starts increasing at some point, we can disentangle the loss given by lack of understanding (and other reliability issues) and capability.

Methodology

We are going to estimate all the terms appearing in (2) and (3) through well-thought questionnaires with human respondents. With them we will be able to answer the first experimental question about whether there is a gain when humans are assisted by a state-of-the-art language models such as GPT-3. The second major experimental question is to assess whether language models fail to complete the task due to a lack of command understanding or competence.

Relying on human data is powerful but limits the number of tasks that we can consider, especially as we need several instances per task, of a range of difficulties. In order to approximate a diverse group of tasks resembling a distribution over everyday tasks $p(\mu)$, we chose four tasks covering each of the four main categories in the human capability hierarchy according to Cattell–Horn–Carroll theory (Carroll 1997) that are not specific to humans (e.g., short memory). In particular, we have one task in each of the following categories:

- "Numerical abilities", represented by a task where price discounts have to be applied. Instance difficulty is given by how many operations are needed.
- "Communication abilities", represented by a task where an email has to be written for a costumer explaining them

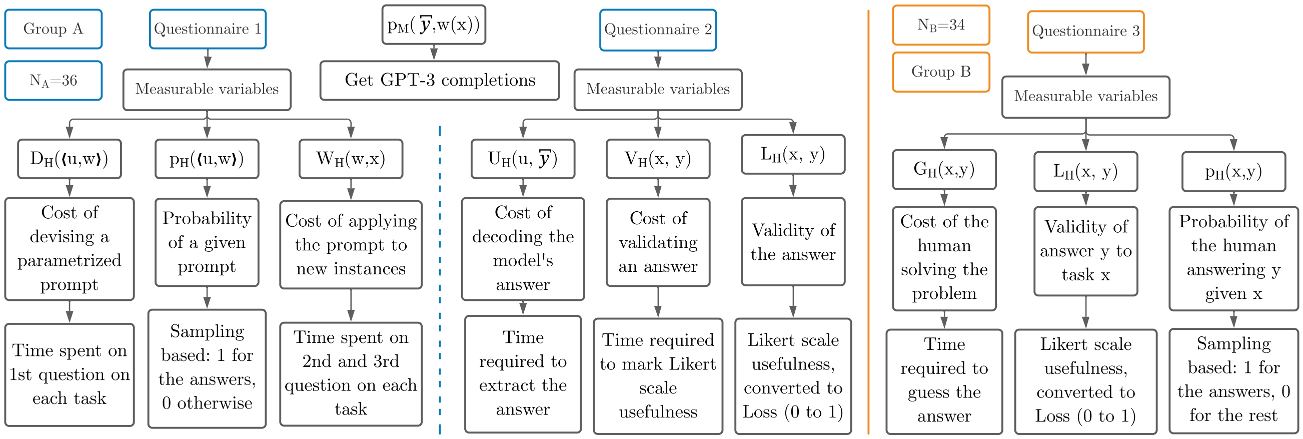

Group A
N_A=36
Questionnaire 1
Measurable variables
D_H((u,w))
Cost of devising a parametrized prompt
Time spent on 1st question on each task
p_H((u,w))
Probability of a given prompt
Sampling based: 1 for the answers, 0 otherwise
W_H(w,x)
Cost of applying the prompt to new instances
Time spent on 2nd and 3rd question on each task
p_M($\overline{y}$,w(x))
Get GPT-3 completions
Questionnaire 2
Measurable variables
U_H(u, $\overline{y}$)
Cost of decoding the model's answer
Time required to extract the answer
V_H(x, y)
Cost of validating an answer
Time required to mark Likert scale usefulness
L_H(x, y)
Validity of the answer
Likert scale usefulness, converted to Loss (0 to 1)
N_B=34
Group B
Questionnaire 3
Measurable variables
G_H(x,y)
Cost of the human solving the problem
Time required to guess the answer
L_H(x, y)
Validity of answer y to task x
Likert scale usefulness, converted to Loss (0 to 1)
p_H(x,y)
Probability of the human answering y given x
Sampling based: 1 for the answers, 0 for the rest

Figure 2: Variables appearing in the cost definitions (2) and (3), and forms from which to obtain them.

whether they made or lost money after an investment. Difficulty is measured by increasingly bad news, as in such situation we expected participants to take more care and time on the framing of the email (MUM effect).

- "Reasoning", represented by the task of proposing a recipe from a list of ingredients and utensils. Difficulty is assessed by the number of ingredients and utensils.
- "Creative writing", represented by the task of writing the lyrics of a song for a two-year old child about animals and what they are doing. In this case, difficulty is measured by the number of animals to be included in the lyrics.

We built three questionnaires in English with three instances in each domain: Q1 and Q2 (group A) aimed at estimating the parameters in $C_{H,M}$, and Q3 (group B) for C_H.

Q1 starts with some information about what an 'autocompletion' system is and some examples at the beginning. It also collects some information about the participants (English level, age, familiarity with language models, and use of virtual assistants). Then, volunteers are asked to generate prompts to make the language model solve the tasks. After they have finished Q1, we use their prompts to generate GPT-3 completions (using davinci-instruct, with default parameters and 256 tokens), which we use to build Q2, where usefulness of GPT-3's completions are assessed. Q1 and Q2 are paired, such that the users receive the completion to their respective prompts. Q3 is independent. A different group of volunteers complete the same tasks but without using language models. It also collects their age and English level. To ensure similar samples for group A (Q1-Q2) and group B (Q3), and no contamination between groups, volunteers were randomly divided into two groups A and B, with questionnaires Q1 and Q2 sent to group A, and Q3 sent to B. In the end, we had $N_A = 36$ and $N_B = 34$ respondents recruited via posts in social networks and internet forums. The tests were administered online using the open-source testing platform Concerto (Harrison et al. 2020).

The way we estimate the value of each term in C_H and $C_{H,M}$ (Eqs. 2 and 3) can be found in Fig. 2. In general, usefulness of the answers is asked to humans through a Likert scale s (1 to 5, from least to most useful), which we convert into loss as $L_H = 1 - (s - 1)/4$. This loss is estimated by the humans themselves. In addition, we conduct an external evaluation L_E, measured by a member of the research team, and serves to give comparable scores across volunteers, and avoid discounting difficulty. Human effort (D_H, W_H, U_H, V_H, G_H) is measured in seconds.

The forms are structured in 4 tasks with 3 instances. We assume that the first instance of each task has a prompting cost (measured in time) of $D_H + W_H$, while for the second and third instances the cost is only W_H. U_H is just the average effort to find the answer in the model completion, and V_H the time to estimate its usefulness.

For the different effort components we use the median of the measured time. It so happens that even if volunteers are specifically instructed to avoid making stops in the forms, some of them inevitably get distracted. As a consequence, the median represents a better way to reduce the possible bias in the time estimates. On the other hand we use the mean for assessing the quality of the answers given by a Likert scale.

Analysis of Results

Let us first compare the correlations between all variables. As indicated in the caption of Fig. 3, we can confidently reject the normality hypothesis for all time distributions. Because of this, we use Spearman correlations. In Fig. 2 (and Fig. 3 in the appendix segregated by domain) we see that a good command of English and previous experience with language models seems useful. The use of virtual assistants however seems uncorrelated, which may be due to continuations being frequently expressed differently from commands. Finally, the use of language models is weakly negatively correlated with self-assessed loss, $L_H(x, y)$ but not with externally-evaluated $L_E(x, y)$, suggesting that people without experience may be easier to impress.

Effort and Loss

Fig. 3 (left) shows the effort to use language models, including the cost of generating a prompt (D_H), wrapping to the

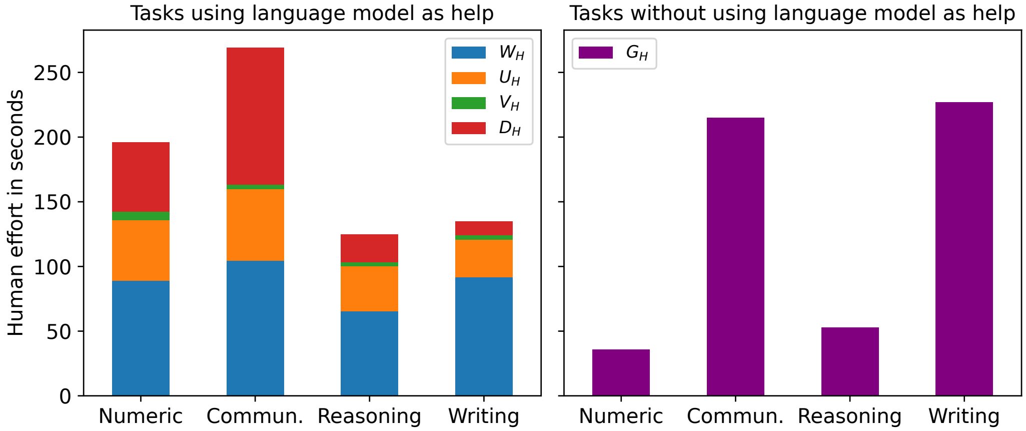


Figure 3: Effort required (median values, in s) to perform each of the tasks with and without access to GPT-3. In all cases except the last one (lyrics) the effort to generate the prompt and validate the answer is greater than to solve it by themselves. Each distribution was Shapiro-tested ($p \leq 3 \cdot 10^{-4}$ in all cases). Then we performed Mann-Whitney U test to compare the effort with and without GPT-3. The Holm-corrected (Holm 1979; Aickin and Gensler 1996) p-values are $p < 1.3 \cdot 10^{-7}$ for Numeric and Reasoning domains, $p = 1.15 \cdot 10^{-2}$ for Communication and $p = 0.33$, that is, no evidence of difference, for the Writing task.

specific instance (W_H), unwrapping the model completion (U_H) and validating it (V_H). In all cases, except in the task of writing song lyrics, the sum of the human effort required to interact with the system is larger than the effort to solve the task without making use of it, as shown in Fig. 3 (right).

On the other hand, if we compare the loss for different tasks in Table 1 we can see that in the Numeric and Communication tasks, the answers of GPT-3 achieve worse self-assessed usefulness (higher loss). In contrast, loss is similar for the other two. We have to note that in Q3 the usefulness of the answers from humans is also self-assessed, so they may differ from the perceived usefulness of answers by others (Hoorens 1993). Furthermore, in the Reasoning task (recipes), answers from the model and human are different: the human just names the dish; while the model often provides the entire recipe, but with unavailable ingredients.

Overall, considering Fig. 3 and Table 1 together indicates that for the first three tasks (Numeric, Communication and Reasoning) the use of GPT-3 would be unwise: it achieved a higher loss and required more time overall to complete. A more lenient appraisal for Writing has to do with the fact that generative tasks are currently the domain where language models shine the most: tasks that are easy to describe and evaluate but hard to solve. However, the fact that loss values are at best as good as humans' indicate that there is still space for future models to improve.

$L_H(x, y)$	Numeric	Commun.	Reason.	Writing
GPT-3	0.61	0.59	0.35	0.47
Human	0.31	0.38	0.35	0.47
p-value	$8 \cdot 10^{-6}$	$1.1 \cdot 10^{-4}$	0.9	0.9

Table 1: Usefulness of the result making use or not of model. Mean values for the loss derived from the Likert values. We also include the Holm-corrected p-values to check for similar mean in each domain. Prior to that we performed Shapiro (normality) and Levene (standard deviation) tests; and as a consequence used Mann-Withney U test for Numeric, Communication and Reasoning tasks, and t-test for Writing.

Finally, the communication task was the one the human volunteers found most challenging. Not only does Fig. 3 show D_H to be much larger than for the other tasks, but also the number of prompts not containing enough information for the model is much larger than in other cases (33% vs up to 14%). However, an important caveat to mention here is that this task is perhaps the one where the stakes were the most difficult to emulate (telling a customer bad news).

Difficulty

Now we discuss how the difficulty of the question affects the quality of the answers given by the system. This is a natural question looking at the proposed decomposition of alignment in (1). To measure this, we look at the easiest instances and see whether the loss falls to 0, not the same thing as what volunteers were measuring: 'usefulness'. This difference is reflected in the loss values in Fig. 4 in the appendix. The difference arises because simple questions such as assessing the price of a 2 for 1 offer are too simple to be perceived as useful. As such, the loss does not have a well-defined trend, or even decreases for more complex tasks.

In order to correctly evaluate how well the answer of the model performs the task at hand, we will use an externally evaluated cost, denoted by $L_E(x, y)$. The results, shown in Fig. 4, are very different between the numeric task and the rest. While loss increases with $\hbar$ for Numeric, the easiest possible instance still has an average loss of $L_E(x, y) = 0.71$, suggesting an understanding gap as indicated in Eq. 1. The other tasks and the self-assessed loss of both GPT-3 and human answers do not show any clear trend, but we think the

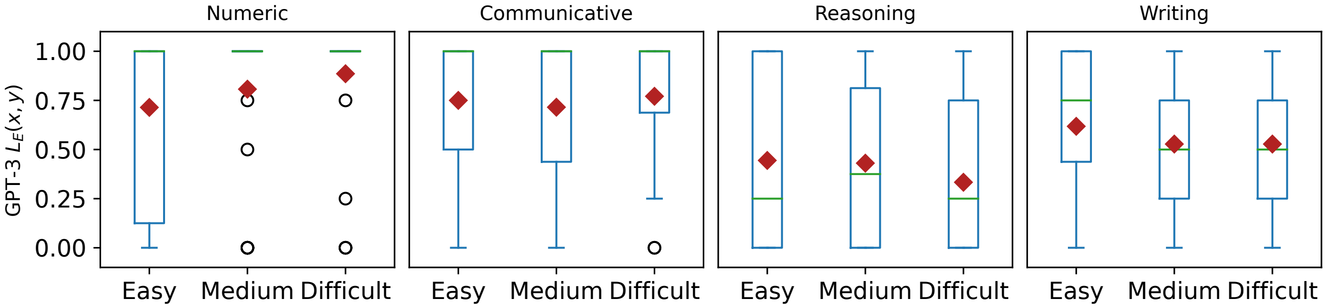


Figure 4: $L_E(x, y)$ changes with difficulty $\hbar$. High loss in the simplest instance indicates an 'understanding' gap. Increasing loss (in the numeric task) means that capability may saturate for complex instances. Plots for $L_H(x, y)$ in Fig. 4 in the appendix.

reasons are different: for Reasoning there is an intermediate level of understanding, while for Writing and especially Communication the task overall is hard and the lack of a clear growing trend does not allow us to tell between lack of competence and failing at understanding commands.

Discussion and Future Work

The progress and full democratisation of language models should be based on a better understanding of their capabilities. One key finding of our work is that, despite sometimes providing excellent answers, the use of these models still requires significant effort by the common human to generate good prompts. Indeed, except for the writing tasks, our results indicate that it would be faster and better if the user solved the task without the help of GPT-3. We expect this to change in the future as models become more accurate, but also as the users adapt to the way models understand commands. It is crucial that we evaluate this properly, using human questionnaires like this one, and not only the results from massive batteries of language models where prompts are specialised for each task (Kohler and Daniel Jr 2021).

In NLP it is usual to evaluate the quality of responses subjectively, but it is less so to measure times. This is more common in other areas where productivity is key, such as software engineering (Sadowski and Zimmermann 2019) and human-computer interfaces (Lewis 1995; Lazar, Feng, and Hochheiser 2017). It is nevertheless essential to take all factors into account and compare both situations in an ecologically valid experiment (De Vries, Bahdanau, and Manning 2020). For example, one could mistakenly believe that the model used in our experiments, GPT-3, is almost as good as humans in generating recipes from lists of ingredients. Unfortunately, this does not take into account that to make these systems useful, humans would need to be able to prompt and read the model's answer faster than they are to solve the task themselves. The advantage of using these models only seems to appear for generative tasks, such as the song lyrics writing. For future studies, we believe it would be informative to carry out similar research with multimodal models. In fact our tasks were designed with this consideration in mind, such that the tasks could be adapted to multimodal input, including the images of our forms, and output, such as videos. Another extension is to analyse other languages, where the capability of the system and the kind of prompts may differ from English significantly.

Similarly, future studies should focus on ecological validity by considering realistic situations where these models are used. This involves modelling different kinds of users using empirical evidence in the short and long terms, analogous to the way software systems and human-computer interfaces are evaluated. This should include how users adapt to these systems and learn to improve the construction and application of prompts, as well as choosing those tasks whose assistance is more useful and safe.

Our work aimed to shed some light into the decomposition of alignment in Eq. 1. For the numeric task (discount application), we can measure both the understanding gap (the gap that happens when the difficulty of the task is minimal but still requires command understanding) and the capability of the system, which was quickly saturated. Unfortunately, one limitation of this methodology is that it is not always easy to find a good range of instances from very easy cases to more difficult ones, because the range of capability of language models is still limited. We hope that future studies with more powerful models will provide some insight on how to better measure the difficulty increase, or even compare what tasks language models and humans find difficult. Furthermore, with the objective of helping make better models, we open source the data collected in our experiments. This should provide a benchmark of prompts where most of the heavy work (prompt generation and task solving without the model) has already been carried out, and the only remaining task is the evaluation of the answer of new models.

We believe the methodology proposed here opens the door to a fairer and more insightful evaluation of language models and other foundation models of the future, which should help better assess their generality and usefulness. It should also help address a crucial aspect of the reliability and safety of these models such as their understanding of commands: very capable systems with poor understanding of our will may pose risks. As such, we advocate for a more realistic evaluation of these models, as they will be used by humans —in the wild.

Acknowledgements

This work has been partially supported by the EU (FEDER) and MINECO grant RTI2018-094403-B-C32 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", Generalitat Valenciana under grant PROMETEO/2019/098, EU's Horizon 2020 research and innovation programme under grant No. 952215 (TAILOR), the Future of Life Institute under grant RFP2-152, US DARPA HR00112120007 (RECoG-AI) and the AI-Watch project by DG CONNECT and DG JRC of the European Commission. PAMC acknowledges MECD grant FPU17/03620. We are also grateful to Fernando Martínez-Plumed, Wout Schellaert and Karina Vold for their feedback on a related document extending some of the ideas in this paper.

References

Aickin, M.; and Gensler, H. 1996. Adjusting for multiple testing when reporting research results: the Bonferroni vs Holm methods. *American journal of public health*, 86(5): 726–728.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, 610–623. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383097.

Bender, E. M.; and Koller, A. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198.

Bommasani, R.; et al. 2021. On the Opportunities and Risks of Foundation Models. arXiv:2108.07258.

Bragg, J.; Cohan, A.; Lo, K.; and Beltagy, I. 2021. Flex: Unifying evaluation for few-shot nlp. *arXiv preprint arXiv:2107.07170*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems 33 (NeurIPS 2020), arXiv preprint arXiv:2005.14165*.

Campagna, G.; Xu, S.; Moradshahi, M.; Socher, R.; and Lam, M. S. 2019. Genie: A generator of natural language semantic parsers for virtual assistant commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 394–410.

Carroll, J. B. 1997. *The three-stratum theory of cognitive abilities.* The Guilford Press.

Cho, J.; and Rader, E. 2020. The role of conversational grounding in supporting symbiosis between people and digital assistants. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1): 1–28.

De Vries, H.; Bahdanau, D.; and Manning, C. 2020. Towards ecologically valid research on language user interfaces. *arXiv preprint arXiv:2007.14435*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Lin, J.; Zou, X.; Shao, Z.; Yang, H.; et al. 2021. CogView: Mastering Text-to-Image Generation via Transformers. *arXiv preprint arXiv:2105.13290*.

Drexler, K. E. 2019. *Reframing Superintelligence*. The Future of Humanity Institute, Oxford University.

Hao, K. 2021. The race to understand the exhilarating, dangerous world of language AI. *Technology Review*.

Harrison, C.; Loe, B. S.; Lis, P.; and Sidey-Gibbons, C. 2020. Maximizing the Potential of Patient-Reported Assessments by Using the Open-Source Concerto Platform With Computerized Adaptive Testing and Machine Learning. *Journal of medical Internet research*, 22(10): e20950.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hernández-Orallo, J.; Loe, B. S.; Cheke, L.; Martínez-Plumed, F.; and Ó hÉigeartaigh, S. 2021. General intelligence disentangled via a generality metric for natural and artificial intelligence. *Scientific reports*, 11(1): 1–16.

Hernández-Orallo, J.; Martinez-Plumed, F.; Avin, S.; Whittlestone, J.; et al. 2020. AI paradigms and AI safety: mapping artefacts and techniques to safety issues. *ECAI*.

Holm, S. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 65–70.

Hoorens, V. 1993. Self-enhancement and superiority biases in social comparison. *European review of social psychology*, 4(1): 113–139.

Izacard, G.; and Grave, E. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Kenton, Z.; Everitt, T.; Weidinger, L.; Gabriel, I.; Mikulik, V.; and Irving, G. 2021. Alignment of Language Agents. *arXiv preprint arXiv:2103.14659*.

Kohler, C.; and Daniel Jr, R. 2021. What's in a Measurement? Using GPT-3 on SemEval 2021 Task 8–MeasEval. *arXiv preprint arXiv:2106.14720*.

Lazar, J.; Feng, J. H.; and Hochheiser, H. 2017. *Research methods in human-computer interaction*. Morgan Kaufmann.

Leike, J.; Martic, M.; Krakovna, V.; Ortega, P. A.; Everitt, T.; Lefrancq, A.; Orseau, L.; and Legg, S. 2017. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*.

Lewis, J. R. 1995. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1): 57–78.

Lieberman, H. 2001. *Your wish is my command: Programming by example*. Morgan Kaufmann.

Lieberman, H.; and Maulsby, D. 1996. Instructible agents: Software that just keeps getting better. *IBM systems journal*, 35(3.4): 539–556.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; and Chen, W. 2021. What Makes Good In-Context Examples for GPT-3? *arXiv preprint arXiv:2101.06804*.

Maedche, A.; Legner, C.; Benlian, A.; Berger, B.; Gimpel, H.; Hess, T.; Hinz, O.; Morana, S.; and Söllner, M. 2019. AI-based digital assistants. *Business & Information Systems Engineering*, 61(4): 535–544.

Mahoney, M. 1999. *The Complexity of Natural Language*. Ph.D. thesis, Dissertation Proposal, Florida Institute of Technology.

Ngo, L. T.; Maeda, Y.; Lee, H.; and Mizukawa, M. 2012. Ambiguous command understanding with commonsense. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 545–550. IEEE.

Perez, E.; Kiela, D.; and Cho, K. 2021. True Few-Shot Learning with Language Models. *arXiv preprint arXiv:2105.11447*.

Qin, G.; and Eisner, J. 2021. Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. *arXiv preprint arXiv:2104.06599*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rahimi, M.; Guo, J. L.; Kokaly, S.; and Chechik, M. 2019. Toward requirements specification for machine-learned components. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 241–244. IEEE.

Rapp, A.; Curti, L.; and Boldi, A. 2021. The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots. *International Journal of Human-Computer Studies*, 102630.

Reynolds, L.; and McDonell, K. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. *arXiv preprint arXiv:2102.07350*.

Russell, S. 2019. *Human compatible: Artificial intelligence and the problem of control*. Penguin.

Sadowski, C.; and Zimmermann, T. 2019. *Rethinking productivity in software engineering*. Springer Nature.

Scao, T. L.; and Rush, A. M. 2021. How Many Data Points is a Prompt Worth? *arXiv preprint arXiv:2103.08493*.

Schick, T.; and Schütze, H. 2020. Few-Shot Text Generation with Pattern-Exploiting Training. *arXiv preprint arXiv:2012.11926*.

Shannon, C. E. 1949. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4): 656–715.

Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J. W.; Kreps, S.; et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Tamkin, A.; Brundage, M.; Clark, J.; and Ganguli, D. 2021. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models. *arXiv preprint arXiv:2102.02503*.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind*, 59: 433–460.

Walker, N.; Peng, Y.-T.; and Cakmak, M. 2019. Neural semantic parsing with anonymization for command understanding in general-purpose service robots. In *Robot World Cup*, 337–350. Springer.

Wang, S.; Tu, Z.; Tan, Z.; Wang, W.; Sun, M.; and Liu, Y. 2021. Language Models are Good Translators. *arXiv preprint arXiv:2106.13627*.

Xu, S.; Semnani, S. J.; Campagna, G.; and Lam, M. S. 2020. AutoQA: From databases to QA semantic parsers with only synthetic training data. *EMNLP*.

Zeng, W.; Ren, X.; Su, T.; Wang, H.; Liao, Y.; Wang, Z.; Jiang, X.; Yang, Z.; Wang, K.; Zhang, X.; et al. 2021. PanGu-α: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation. *arXiv preprint arXiv:2104.12369*.

Zhao, T. Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*.