We have built the AIscreenR as a flexible software, allowing users to conduct multiple screenings simultaneously based on multiple prompts, API models, iterations of the same request, and nucleus samples (i.e., different top\_p or temperature values). The software further allows the user to send the same request (i.e., repeatedly asking the same question) multiple times to avoid random noise in individual model responses (especially relevant when using GPT-3.5 models). When this feature is used, the final GPT decision is based on the probability of inclusion across the iterated requests, and the specific inclusion threshold can then be determined by the user (see Figure 4 of this paper). This also allows the users to test model response consistency. Moreover, the software has been built to draw on multi-core processing, thereby allowing users to significantly speed up the screening. Finally, we built the package so that reviewers can work with two different types of function calls; one yielding simple/trinary (i.e., 1 = {include}, 1.1 ={uncertain}, and 0 = {exclude}) decisions and/or another yielding descriptive responses to the screening requests. We consider the former to be the main work engine, whereas the latter can be pivotal when examining discrepancies between GPT and human screener decisions.

Although we have tried to accommodate the requirements set forth by evidence organizations, we do not consider our solution to be a final one. Our aim has merely been to show one way in which GPT API models can be used for TAB screening in large-scale systematic reviews that can inspire and be transferred to future applications of TAB screening with all kinds of LLMs

A side-effect of such research would further be that the costs of using GPT models may be substantially reduced, which can be a major barrier to using GPT-4 models for TAB screening at the current point in time. These models are still rather expensive (in absolute terms, not compared to hiring a human screener). Thus, another line for future research could be to investigate the performance of cheaper GPT-4 models, such as GPT-4o and GPT-4-turbo.

*Evidence from previous research about common human performance*:

In fact, research suggests that screeners on average tend to miss between 3% and 24% of all eligible studies depending on the level of content knowledge – often with a substantial impact on the final quantitative results (Buscemi et al., 2006; Waffenschmidt et al., 2019). In medicine, the number of missed studies may be even higher, especially when relying on student screeners (Ng et al., 2014).

An additional challenge is that most automated screening tools are based on supervised and active learning methods. This means that they need to be trained on a large enough set of included and excluded references to perform adequately, which is time-consuming as well. Finally, when automation tools are used for prioritized screening, there is no clear rule for determining when it is safe to stop screening with regard to finding all or close to all eligible references. Although various stopping rules have been proposed, the adequacy of these rules is sensitive to a range of factors, such as the length of the database, the prevalence of relevant studies, and the balance between relevant and irrelevant records (Campos et al., 2023; König et al., 2023; Van De Schoot et al., 2021)

By doing so, we provide a basis for making sense of specific TAB screening performances, which is needed in order to meaningfully assess the fruitfulness of using GPT models (and other LLMs) as independent TAB screeners in systematic reviews.

This seems to create a screening paradox, which might be one of the main reasons for reviewers to mistrust the application of machine-learning tools (O’Connor et al., 2019). While trying to reduce selection biases caused by single screening, automated screening potentially introduces a novel type of publication/selection bias, referred to by König et al. (2023) as ‘artificial screening bias’.[[1]](#footnote-1)

Moving on to the *balanced accuracy* metric, we found average performance levels of 0.874, 95% *CI*[0.857, 0.890] among assistant screeners and 0.933, 95% *CI*[0.899, 0.961] among author screeners. We found the difference between the group means to be statistically significant with *F*(1, 10.1) = 18.22, *p* = .002. Finally, the overall *nMCC* values were 0.860, 95% *CI*[0.835, 0.882] and 0.925, 95% *CI*[0.880, 0.953] for the assistant and author screeners, respectively. The group averages were found to be statistically different from each other with *F*(1, 11) = 9.65, *p* = .01.

The average balanced accuracy and *nMCC* values of the NIPH researchers were 0.905, 95% *CI*[0.859, 0.943] and 0.879, 95% *CI*[0.720, 0.951], respectively.

For the *nMCC* metric, we calculated the sampling variance and confidence interval by transforming the correlations to Fisher’s z-scores, as typically done in meta-analysis (Borenstein et al., 2009).

A side-effect of conducting these experiments was further to quality assure the AIscreenR package (Vembye, 2024) for automated TAB screenings using GPT API models. We consider this test to be important in order to be able to scale up our suggested screening approach.

An impressive fact regarding these results is further that we approximately spent 5-10 minutes engineering the used prompt presented in Textbox 1. In fact, the prompt represents a first trial prompt. Unless we were extremely lucky to hit the right prompt in the first trial, this might indicate that in some applications the GPT API models are not as prompt-sensitive as would be theoretically expected. Yet, we only presume this to be the case in very specific circumstances as is the case here where the screening involves a standardized intervention with a specific name.

Concretely, we have aimed to accommodate requirement *(a)* by building our framework and codes so that they can be remodeled to work with other API models than OpenAI’s. This means that our setup aims to be agnostic to the given provider of the given LLM and will be viable as long as reviewers have public access to LLM models. We aimed to support Campbell’s requirement *(b)* by developing the new benchmark scheme and by showing that GPT API screening is appropriate in high-quality reviews, whereas the development of the AIscreenR package and the quality tests hereof were meant to accommodate Campbell’s requirement *(c)*. Moreover, to fulfill requirement *(f)*, we built the AIscreenR package as open-source software, allowing others (e.g., the Evidence Synthesis Hackathon, Campbell Collaboration, or the EPPI-Reviewer team) to contribute to the development and ongoing support of the software. Finally, we developed our workflow and guidelines to underpin requirements *(d)* and *(e**)*. Requirement *(e)* is as such not necessary in our case since we are working we *pre-*trained models. Instead, the performance of the prompt(s) used for screening needs to be *tested* and compared against human performance measures before credible TAB screening can be initiated.

Among other things, this allows reviewers to draw on features such as function calling (i.e., making prompts without the need to explicitly specify how the model shall respond to the screening request) as well as multi-core processing, something that speeds up the screening significantly. After having used the AIscreenR software to conduct our classifier experiments, we feel confident to conclude that the software works as expected. Hence, we believe that reviewers can confidently use this software in their high-quality systematic reviews as well.

A key part of setting up a reliable GPT API screening is to thoroughly validate the performances of one’s screening prompt(s) before making any full-scale screening. For such assessments, we developed a new, empirically informed benchmark scheme for interpreting acceptable and unacceptable screening performance in high-quality reviews. Based on typical human screening performances in 22 high-standard systematic reviews, we suggest that if an automated screening can yield a recall value above .75, it should be acknowledged as being on par with typical human performance and can be confidently used as an independent second screener. In addition, we suggest that a specificity value equal to or above .8 should be accepted in high-standard reviews as long as the recall is equal to above .75 as well since a low specificity does not threaten the interval validity of reviews.

Since our screening guideline allows the GPT API models to err to a degree similar to human screeners, GPT API screening most be combined with other traditional screening techniques, such as forward and backward citation tracking, to ensure that potentially missed studies re-enter the review. Moreover, we never think a GPT API model should be used as a stand-alone screener. There must always be a human in the loop, meaning that humans must always take the role of the first screener of titles and abstracts in high-quality systematic reviews.

FIGURE 4: Decision sensibility of the gpt-3.5-0613 model across inclusion probability thresholds

|  |
| --- |
| A |
| B |

*Note*: The inclusion probability threshold on the x-axis indicates the required number of times the given title and abstract record needed to be included over the 10 repeated requests in order to be coded as relevant. For example an inclusion probability threshold of 0.1 means that the record was coded as relevant if the GPT model included it in 1 or more out of 10 requests.

In the initial phase of our prompt engineering, we assumed that the more detailed background information we could add to a single prompt the better the GPT API model would perform. This approach was based on the conception that the model needed to be “trained” with the correct wording. Yet, from our experience, this was a misperception of how this type of model works. As indicated in the GPT acronym, these models are *pre-trained*, meaning that they do not need to be further trained in terms of wording. What they need to work properly are concise prompts and thus, test performances increased dramatically when given more precise prompts with fewer inclusion/exclusion criteria at a time. As such, for Experiment 3, we developed and evaluated the concept of multi-prompt screening where each inclusion criteria were prompted individually relative to adding all in- and exclusion criteria to the same prompts.

If conducting a complex review including many inclusion/exclusion criteria, we suggest conducting what we have coined multi-prompt screening. That is screening with multiple prompts, where each inclusion criteria is prompted individually Alternatively, one could conduct what we define as *hierarchical screening* where a study record is considered irrelevant if it is excluded at any step in the multi-prompt screening. This procedure is depicted in Figure 5.[[2]](#footnote-2)

FIGURE 5: *Hierarchical screening*

Prompt 1:   
With inclusion/exclusion criertion 1

*If included, the record moves to next prompt, otherwise the record is excluded*

Prompt 2:   
With inclusion/exclusion criertion 2

*If included, the record moves to next prompt, otherwise the record is excluded*

Prompt 3:   
With inclusion/exclusion criertion 3

*Continue as long as necessary*

Prompt x:   
With inclusion/exclusion criertion x

If using hierarchical screening, we suggest ordering the prompts so that the prompts excluding the largest body of references appear first and prompts with more specific inclusion criteria follow thereafter (as suggested by Brunton et al., 2017). This approach will be more efficient both in terms of money and time. A further side-effect of this approach is that all title and abstract records will be mapped on what exact inclusion criteria they were excluded upon. However, a shortage of this screening approach is that it is strongly dependent on the quality of the used prompts. Although more costly, we therefore recommend using multi-prompt screening where all title and abstract records are screened with all prompts, since this approach potentially guards against insufficient prompting. Assume, for example, that one made six prompts, one for each of the inclusion criteria, but one of the prompts wrongly excluded a large share of relevant records at an early stage of the screening when scaled up from the test setting. Then those studies would be lost in the hierarchical screening in Figure 5. If instead all records had been screened with all prompts, then one could overcome the above problem by coding records as relevant if they were included in 5 out of 6 prompts, as we did in Experiment 3 above.

**Add to conclusion:**

Although we cannot reject that single-prompt screening might be viable in complex review settings with many inclusion criteria, and multi-prompt screening is more costly, we think multi-prompt screening is most appropriate to use in complex review settings. This is so because multi-prompt screening makes the screening more flexible allowing the model to cast a larger degree of uncertainty. Moreover, when using multi-prompt screening all titles and abstracts will be mapped on the exact reasons for exclusion, increasing the transparency of the review and making it much easier to decode what factors made the GPT model work well.

1. To alleviate this issue, a new tentative guideline termed SAFE has been developed in which it is suggested to use multiple machine learning algoritmes in order to detect all relevant references in the bulk of records (Boetje & van de Schoot, 2024). However, this framework has not yet been thouroughly enough tested to know if the SAFE procedure allows reviewers to detect all relevant studies using machine learning algoritms included in screening softwares, such as ASReview (Van De Schoot et al., 2021). [↑](#footnote-ref-1)
2. To support this type of screening, we show how this can be practically executed in one of the accompanied vignettes to the AIscreenR (Vembye, 2024). [↑](#footnote-ref-2)