# Exploring the text analytic capacities of GPT4

## Description

Tweet

**On this page:**


1. What GPT4 says it can do
2. What we can do using apps that interface with GPT4 et al
3. Evaluation of AI text analysis performance
4. How AI helped me to help AI improve its performance!


## 1. What GPT4 says it can do?

RD: What kinds of text analyses can I expect GPT4 to be able to do?
GPT4: *While I cannot provide specific details about GPT-4â??s capabilities beyond my knowledge cutoff, I can provide some insights based on GPT-3â??s functionalities. GPT-4, being an advanced version, is expected to have improved performance and more accurate results. Here are some types of text analysis that a GPT-4 model could potentially handle:*

1. *Sentiment analysis: Determine the sentiment of a given text, whether itâ??s positive, negative, or neutral, and potentially provide a sentiment score.*
2. *Text summarization: Summarize long pieces of text into shorter, more concise versions while retaining the main ideas.*
3. *Named entity recognition (NER): Identify and classify entities within the text, such as names of people, organizations, locations, dates, and other types of entities.*
4. *Keyword extraction: Extract the most relevant keywords or phrases from a given text, providing a quick overview of the main topics.*
5. *Text classification: Categorize text into different topics, genres, or other predefined classes based on its content.*
6. *Relationship extraction: Identify relationships between entities within a text, such as a personâ??s affiliation with an organization or the connection between two events.*
7. *Language translation: Translate text between different languages while maintaining the original meaning and context.*
8. *Paraphrasing: Reword a given text while preserving its original meaning, which can be useful for rephrasing sentences or generating alternative versions of content.*
9. *Grammar and style corrections: Identify and correct grammatical errors, typos, and stylistic issues in a text.*
10. *Content generation: Create new, coherent, and contextually relevant content based on user inputs, prompts, or templates.*

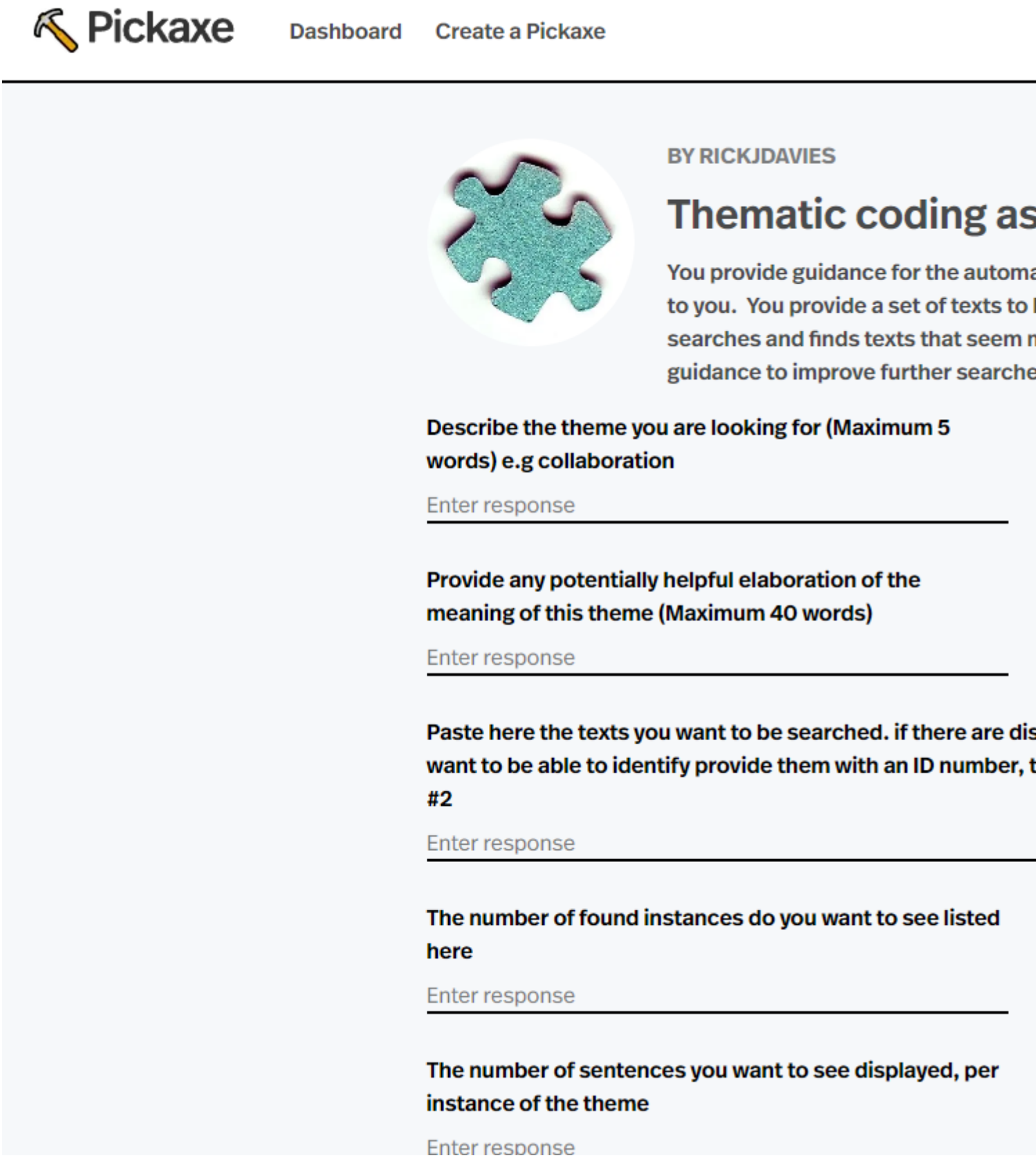
*Please note that these are examples of text analysis tasks that an advanced language model like GPT-4 might be able to handle, but specific capabilities might vary depending on the modelâ??s training and*

*implementation.*

## 2. What we can do using apps that interface with GPT4 et al

This is what I have been doing recently, partly through the use of a new app known as Pickaxe, which â??sits on top ofâ?• GPT4 and its predecessors. PS: The number of these apps is large and growing very rapidly. I will try to provide some guides to them here shortly.

Using Pickaxe you can draft a â??promptâ?•, or a whole structure of prompts, which will send a request to GPT, which then returns a finding. Nothing new here. But, in addition, you can â??publishâ?• your prompt, making it usable by others. Including the ability  to customise the prompt to their more specific requirements.




Here below is a list of the Pickaxes I have developed so farâ?¦mainly oriented around my interests relating to qualitative analysis of text data. Warningâ?¦ None of these is perfect. Inspect the results carefully and donâ??t make any major decisions on the basis of this information alone. Sometimes you may want to submit the same prompt multiple times, to look for variability in the results.

Please use the Comment facility to provide me with feedback on what is working, what is not and what else could be tried out. This is all very much a work in progress. For some background see this other recent post of mine: Using ChatGPT as a tool for the analysis of text data

**Summarisation**

Text summariser The AI will read the text and provide three types of summary descriptions for each and all of the texts provided. Users can determine the brevity of the summaries

Key word extraction. The AI will read the text and generate ranked lists of key words that best describe the contents of each and all of the texts provided.

**Comparison**

Text pile sorting The AI will sort texts in two piles representing the most significant difference between them, within constraints defined by the user

Text pair comparisons The AI will compare two descriptions of events and identify commonalities and differences between them, within constraints defined by the user

Text ranking. The AI will rank a set of texts, on one or more criteria provided by the user. An explanation will be given for the texts in the top and bottom rank positions

**Extraction**

Thematic coding assistant You provide guidance for the automated search for a theme of interest to you. You provide a set of texts to be searched for this theme. AI searches and finds texts that seem most relevant. You provide feedback to improve further searches.

> PS: This Pickaxe needs testing against data generated by manual searches of the same set of text for the same themes. If you have any already coded text that could be used for such a test please let me know: rick.davies@gmail.com For more on how to do such a test see section 3 below.

Actor & relationship extraction AI will identify names of actors mentioned in texts, and kinds of relationships between them. The output will be in the form of two text lists and two matrices (affiliation and adjacency), in csv format.

Adjective Analysis Extraction The AI will identify ranked lists of adjectives that are found in one or more texts, within constraints identified by the user.

Adverb extraction
The AI will identify a ranked list of adverbs that are found within a text, within constraints identified by the user.

**Others of possible interest**

Find a relevant journalâ?¦that covers the subject that you are interested in. Then have those journals ranked on widely recognised quality criteria. And presented in a table format

## 3. Evaluation of AI text analysis performance

It is worth thinking how we could usefully compare the performance of GPT4 to that of humans on text analysis tasks. This would be easiest with responses that generate multiple items, such as lists and rankings, which lend themselves to judgements about degrees of similarity/difference â?? the use of which is made clearer below.
There are three possibilities of interest:

1. A human and the AI might both agree that a text, or instance in a text, meets the search criteria built into a prompt. For example, it is an instance of the theme â??conflictâ?•.
2. A human might agree that a text, or instance in a text, meets the search criteria built into a prompt. But the AI may not. This will evident if this instance has not been included in its list. But will be on a list developed by the human.
3. The AI might agree that a text, or instance in a text, meets the search criteria built into a prompt. But the human may not. This will evident if this instance has been included in its list. But will not be on a list developed by the human.

Matrix

| | | Human says text does describe instance of a theme | |
|---|---|---|---|
| | | Yes | No |
| AI says text does describe instance of a theme | Yes | 1 True Positives | 3 False Positives |
| | No | 2 False Negatives | True Negatives |

Ideally both human and AI would agree in their judgements on which texts were relevant instances. In which case all the found instances by both parties would be in the True Positive cell, and all the rest of the texts were in effect in the True Negative box. (TP+TN)/(TP+FP+FN+TN) is a formula for measuring this form of performance, known as Classification Accuracy. This example would have 100% classification accuracy. But such findings are uncommon.

How would you identify the actual numbers in each of cells above? This would have to be done by comparing the results returned by an AI to those already identified by the human. Some instances would be agreed upon as the same as those already identified â?? which we can treat as TPs. Others might strike them as new and relevant and had not previously been identified (FN)s. The humanâ??s coding would then be updated so that such instances were now deemed TPs. Others would be seen as inappropriate and non-relevant instances (FPs).
If there were some FPs what could be done. There are two possibilities:

1. The human could ask themselves how can they can edit the AI prompt to improve its identification of these kinds of instances. In doing so it would be learning how to work better with the AI. This seems likely to be a common response, judging from a sample of the

rapidly growing prompt literature that I have scanned so far.

2. The text of one or more identified FP instances could be inserted into body of the prompt, as a source of additional guidance. Then the use of that prompt could be reiterated. In doing so the AI would be adapting its response in the light of human feedback. It would be doing the learning. This is a different kind of approach, which is happening already within GPT4, but probably much less often in the prompts designed by non-specialist human users.

After the second iteration of the prompt the incidence of FPs could be reviewed again. A third iteration could be prepared, including an updated feedback example generated by the AIâ??s second iteration. The process could be continued. Ideally the classification accuracy of the AIs work would improvised with each iteration. In practice progress may not may not be so smooth.

**A wider perspective**

What I have described is an evolutionary search strategy, involving variation, selection and reproduction.:

1. Variation: A population of possibly relevant solutions is identified by the first iteration of the prompt. That is, a list of identified instances is generated.
2. Selection: The poorest fitting instance is selected as an example of what is not relevant, and inserted into the original prompt text with that label.
3. Reproduction: The revised prompt is reiterated, to generate a new and improved set of variant instances

There is a similar process built into the design of Stable Diffusion, which is designed to generate images from text prompts. An initial text prompt generates four images, which are variants of the prompt idea. The user selects one and can then reiterate the use of the prompt, either in its original form or with some further edits. Again four new variants are generated.

There is however one significate difference, at least as I currently understand the Stable Diffusion process. The selection choice focuses on the most desirable variant and its reproduction into new variant forms, not the least desirable variant and its elimination from the population of reproduced variants. This embodies a â??survival of the fittestâ?• approach, not the â??non-survival of the least-fittingâ?•. In this respect it appears to be a more convergent process â?? though there do see to be other user adjustable parameters which can affect how varied new variants are â?? see the â??Guidance scaleâ?•

**Implications for the use of GPT4 et al**

In GPT4 you can â??regenerateâ?• new responses to a given prompt i.e create set of variants. But these are presented as a sequence of opportunities, not as a set that is available all once, which can be viewed in comparison to each other. One possible way around this this is to build into a prompt a request for N number of versions of the response â?? which would be all visible at the same time. The best of these could then be investigated, and with any further prompts also making a request for N variants of a response. The main challenge with this approach seems likely to be space constraints, the size limit on the word length of the complete response. But this seems likely to increase over time, if computing capacity continues to increase.

## 4. How AI helped me to help AI improve its performance!

Okay, I know this is a very long post, but do continue reading.  After the long ramble of ideas just above, I decided to ask GPT4 about the differences and similarities between genetic algorithm type searches and reinforcement learning approaches. I will copy and past what happened next, below:

**Me: Please describe the similaritiesâ?? and differences between reinforcement learning and genetic algorithms**
GPT4: Reinforcement Learning (RL) and Genetic Algorithms (GA) are both optimization techniques that can be used to solve various problems, particularly in artificial intelligence and machine learning. While they share some similarities, they are fundamentally different in their approaches and underlying mechanisms.

Similarities:

1. Optimization: Both RL and GA are optimization methods aiming to find the best solution for a given problem. They iteratively adjust their solutions based on feedback or evaluation.
2. Exploration vs. Exploitation: Both techniques must balance exploration (searching new areas of the solution space) and exploitation (refining existing solutions) to achieve optimal results.
3. Stochastic Processes: Both methods involve randomness in their search procedures. In RL, agents often take random actions to explore the environment, while GA introduces randomness through mutation and crossover.
4. Population-based: Although not always the case in RL, some algorithms, like Evolutionary Strategies or Genetic Reinforcement Learning, employ a population-based approach similar to GA. These methods maintain multiple candidate solutions and update them over time based on their performance.

Differences:

1. Representation: RL represents solutions as policies or value functions that map states (or state-action pairs) to actions (or values). GA represents solutions as individuals in a population, typically encoded as strings or data structures, which directly or indirectly represent a solution to the problem.
2. Learning Mechanism: RL learns through interaction with an environment, where an agent takes actions, observes the resulting states and receives feedback in the form of rewards or penalties. GA is based on the principles of natural evolution, where the fittest individuals are selected to reproduce and generate offspring that inherit their traits.
3. Evaluation: In RL, the agentâ??s performance is assessed through the cumulative rewards obtained in an episode or over multiple episodes. In GA, a fitness function evaluates each individual in the population based on how well it solves the problem.
4. Update Mechanism: RL algorithms update their policies or value functions using methods like Temporal Difference (TD) learning, Q-learning, or Policy Gradient. GA updates its population through genetic operators like selection, crossover, and mutation.
5. Applicability: RL is particularly suited for sequential decision-making problems in which an agent interacts with an environment. GA is more versatile and can be applied to a wide range of optimization problems, including those without sequential decision-making or clear environment

interactions.

In summary, reinforcement learning and genetic algorithms share some similarities in their optimization goals, exploration-exploitation balance, and stochastic processes. However, they differ in terms of representation, learning mechanisms, evaluation, update mechanisms, and applicability to various problem domains.

**Me: Given your explanation just provided I will now give you an example of a package of prompt commands that has been used with GHP4 on another occasion. Please read it and tell me which of the two methods (reinforcement learning or GA) it is most similar to, and why. Or, if it is not similar to either please tell me and explain why it is not. The text of the prompt package is as follows:**

> **1. The user want you to find sections of text, which will be called â??instancesâ??, that contain this theme: [user enters data here]**
> **2. Here is some extra clarification on the meaning of this theme: [user enters data here]**
> **3. Here is the body of text to be searched: [user enters data here] If there are separate sections of text each of these will have an ID number preceded by the # symbol**
> **4. Please list up to [user enters data here] instances that contain this theme, ranked in order of how well they match the theme. If only a shorter number can be found, display that number of instances only. If more than that number can be found report the number of extra instances that were found, at the end of the requested list of instances.**
> **5. With each instance, display a maximum of [user enters data here] sentences of the relevant text.**
> **Include the hashtag number of the source text, at the end of this text, if it has been provided by the user. Follow the text of each sentence with a one sentence explanation for its inclusion.**
> **6. Do not include [user enters data here]**
> **7. After you have provided your first response, the user will provide you with feedback on where you made any errors. Specifically, this will be in the form of the instances which have been listed by you as instances of the theme, which the user thinks are not correct i.e. False Positives (FPs). If no feedback has yet been provided you will see this message â??No feedback yetâ??. Please note that this does not mean all your listed instances are already correct.**
> **8. When the user clicks â??Submitâ?• please redo your analysis using this feedback to improve your performance. By this we mean a reduction in the number of listed instances which are likely to be classed as False Positives**
> **9. But do not use this feedback if the body of text to be searched, given at point 3 above, has been changed. Treat that as a new exercise.**

GPT4: The given prompt package does not directly resemble either Reinforcement Learning (RL) or Genetic Algorithms (GA) in their traditional forms. However, it is somewhat more related to a Reinforcement Learning setting, given the presence of feedback and iterative improvement. Let me explain why:

The problem itself is an instance of text analysis and information extraction, which does not inherently require optimization techniques like RL or GA. Instead, it could be solved using natural language

processing techniques, such as named entity recognition, topic modeling, or even machine learning classifiers.

The feedback mechanism mentioned in points 7 and 8 resembles a reinforcement learning setting, where an agent (in this case, the AI model) receives feedback from the user regarding its performance (False Positives in this case). The agent is expected to improve its performance using this feedback, similar to the way RL agents learn from rewards or penalties.

The iterative improvement process, mentioned in points 7, 8, and 9, is also characteristic of a reinforcement learning setup. An RL agent learns from its environment by interacting with it, receiving feedback, and updating its policy or value function accordingly.

However, it is essential to note that the prompt package does not fit the traditional reinforcement learning paradigm perfectly. The problem does not involve sequential decision-making or interaction with an environment where an agent takes actions and transitions to different states.

In conclusion, while the prompt package does not directly correspond to either Reinforcement Learning or Genetic Algorithms, it shares some similarities with Reinforcement Learning due to the presence of feedback and iterative improvement. A customized approach involving natural language processing and some form of feedback-based learning (possibly inspired by RL) would likely be more suitable for solving this problem.

**Me: How would you change your analysis, if at all, if I told you that this prompt package is being used for identification of themes in a text, which inherently involves fuzzier category boundaries than is the case with named entity recognition. It could be argued that there is an optimisation process involved here, where the AI is seeking to find the most accurate model of that theme i.e one with least false positives and least false negatives, as identified when that modelâ??s results are compared to the users own categorizations (which is implicitly happening when they provide feedback on the false positive instances)**

AI: Given this new information, it is clearer that the prompt package involves an optimization process aimed at minimizing false positives and false negatives in theme identification. While the problem still does not fit perfectly into traditional Reinforcement Learning (RL) or Genetic Algorithm (GA) frameworks, it may be possible to adapt these methods to address the problem, especially when considering the feedback mechanism and iterative improvement process.

One approach could be to combine Reinforcement Learning with supervised learning techniques for natural language processing. In this case, the AI agent could learn a policy for selecting instances of the theme using features derived from the text. The agentâ??s performance would be evaluated based on feedback from the user (i.e., false positives). The agent would then iteratively update its policy to optimize its performance in identifying themes and minimizing both false positives and false negatives. RD: This is a feature currently built into the prompt]

To further improve the model, active learning techniques could be employed, where the AI agent would actively request feedback from the user on instances it is uncertain about. This would enable the agent to learn more effectively from a smaller amount of feedback, refining its understanding of the theme and its fuzzy boundaries. RD: This feature is not in the prompt but sounds very useful, if not more useful

Alternatively, Genetic Algorithms could be used to optimize the feature extraction process or the parameters of a machine learning classifier designed to identify themes in the text. The population in the GA could represent different feature extraction methods or classifier parameters, and the fitness function would be based on the modelâ??s ability to identify themes with minimal false positives and false negatives. The GA would then evolve the population through selection, crossover, and mutation to find the optimal solution.RD: I am not sure how one would set this up, and whether it would be worth the effort

In summary, while the problem of theme identification with fuzzy category boundaries still does not fit perfectly into traditional Reinforcement Learning or Genetic Algorithm frameworks, it is possible to adapt these methods to address the problem by leveraging the feedback mechanism and the iterative improvement process. Combining Reinforcement Learning with supervised learning techniques, using active learning, or employing Genetic Algorithms to optimize feature extraction or classifier parameters could potentially lead to an effective solution for this problem.

**Me: I especially like your second and third paragraph. Regarding this idea â??To further improve the model, active learning techniques could be employed, where the AI agent would actively request feedback from the user on instances it is uncertain about. This would enable the agent to learn more effectively from a smaller amount of feedback, refining its understanding of the theme and its fuzzy boundaries.â?• ,could you please suggest some improvements to the wording of the prompt, which i will present again here:** [same as already shown above]

AI: To incorporate active learning into the prompt package, you can add a step that allows the AI agent to request feedback from the user on uncertain instances. Here is the revised prompt package with this modification:

1. The user wants you to find sections of text, which will be called â??instancesâ??, that contain this theme: [user enters data here]
2. Here is some extra clarification on the meaning of this theme: [user enters data here]
3. Here is the body of text to be searched: [user enters data here] If there are separate sections of text, each of these will have an ID number preceded by the # symbol.
4. Please list up to [user enters data here] instances that contain this theme, ranked in order of how well they match the theme. If only a shorter number can be found, display that number of instances only. If more than that number can be found, report the number of extra instances that were found at the end of the requested list of instances.
5. With each instance, display a maximum of [user enters data here] sentences of the relevant text. Include the hashtag number of the source text at the end of this text if it has been provided by the user. Follow the text of each sentence with a one-sentence explanation for its inclusion.
6. Do not include [user enters data here]
7. In addition to the instances listed above, the AI agent may present up to [user enters data here] instances it is uncertain about. The user is requested to provide feedback on whether these instances are relevant to the theme or not.
6. After you have provided your first response, the user will provide you with feedback on where you made any errors. Specifically, this will be in the form of the instances which have been listed by you as instances of the theme, which the user thinks are not correct, i.e., False Positives (FPs). If no feedback has yet been provided, you will see this message â??No feedback yetâ??. Please note that this does

not mean all your listed instances are already correct.
When the user clicks â??Submit,â?• please redo your analysis using this feedback to improve your performance. By this, we mean a reduction in the number of listed instances which are likely to be classed as False Positives.
But do not use this feedback if the body of text to be searched, given at point 3 above, has been changed. Treat that as a new exercise.

By adding point 7, the AI agent is given the opportunity to actively request feedback from the user on instances it is uncertain about, facilitating a more effective learning process with a smaller amount of feedback.

**Me: well done, thank you!**

**Category**

1. Software

**Date**
09/11/2026
**Date Created**
20/03/2023
**Author**
admin