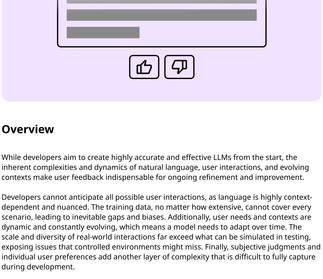
IN-CONTEXT REACTIONS

Collect quick user feedback to improve the model



Overview

While developers aim to create highly accurate and effective LLMs from the start, the inherent complexities and dynamics of natural language, user interactions, and evolving contexts make user feedback indispensable for ongoing refinement and improvement.

Developers cannot anticipate all possible user interactions, as language is highly context-dependent and nuanced. The training data, no matter how extensive, cannot cover every scenario, leading to inevitable gaps and biases. Additionally, user needs and contexts are dynamic and constantly evolving, which means a model needs to adapt over time. The scale and diversity of real-world interactions far exceed what can be simulated in testing, exposing issues that controlled environments might miss. Finally, subjective judgments and individual user preferences add another layer of complexity that is difficult to fully capture during development.

Collecting user feedback effectively involves several challenges:

- encouraging users to provide feedback can be difficult, especially if the process is not straightforward or incentivized.
- users often provide incomplete, unclear, or insufficiently detailed feedback, making it challenging to extract actionable insights.
- gathering feedback in real-time without interrupting the user experience. Intrusive requests can annoy users, while passive methods may result in lower response rates.
- frequent requests for feedback can lead to user fatigue, reducing response rates over time.

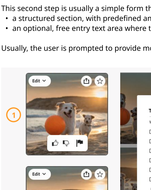
Solution

In social media apps, reaction buttons like thumbs-up or thumbs-down provide a quick and effortless way for users to express their feelings or opinions without the need to type out a response. This lowers the barrier to engagement.

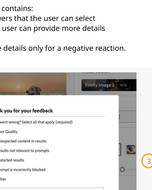
LLM powered products leverage this already popular and very recognizable interaction to gauge the usefulness or accuracy of the generated outputs.

In-context placement

Following the direct manipulation interaction style, feedback actions are best displayed close to the interface element they relate to. This makes them easy to see and interact with, increasing the user engagement.



YOU, like many other conversational AI products, displays thumbs-up and thumbs-down icon buttons at the end of each response. The actions are visible, yet subtle.



Adobe Firefly shows reaction icons on hover for each of the four images generated.

Products like UIZard, where the generated asset is a visual artefact on a canvas, display a brief feedback question in the style of a message bubble, attached to the artefact. This interaction is still optional but more noticeable and the user needs to take an action to dismiss the question.

When the algorithm generates multiple assets, like a series of screens from a user flow, a single feedback question is displayed like a sticky, floating toast message at the top-center of the screen.

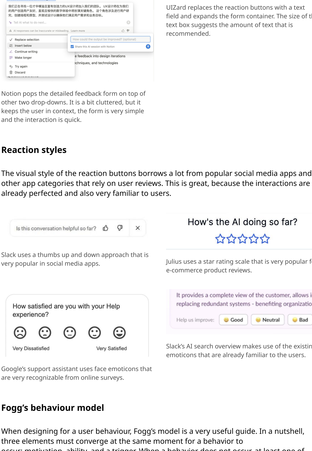
Progressive disclosure

Offering such a low effort feedback solution is great, because it increases user engagement. A like or dislike reaction doesn't provide a lot of valuable information for the data science team though. More detailed and structured feedback is needed to make it easier to incorporate it in the model and this can be solved by asking for more in a second, optional step.

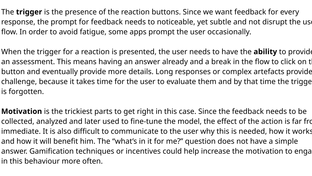
This second step is usually a simple form that contains:

- a structured section, with predefined answers that the user can select
- an optional, free entry text area where the user can provide more details

Usually, the user is prompted to provide more details only for a negative reaction.



Adobe Firefly makes a very smooth transition from the reaction to the form. After reacting negatively, the user gets a temporary feedback button to provide more details.



For such a rich answer, like the one shown by Google in the AI overview, upon a negative reaction, the components become interactive and the user is asked to select the part of the response that was unsatisfactory. Only then the form is displayed as a popup.

Bard displays an inline section that is compact, subtle and does not disrupt the user flow.

UIZard replaces the reaction buttons with a text field and expands the form container. The size of the text box suggests the amount of text that is recommended.

Notion pops the detailed feedback form on top of other two drop-downs. It is a bit cluttered, but it keeps the user in context, the form is very simple and the interaction is quick.

Reaction styles

The visual style of the reaction buttons borrows a lot from popular social media apps and other app categories that rely on user reviews. This is great, because the interactions are already perfected and also very familiar to users.

Slack uses a thumbs up and down approach that is very popular in social media apps.

Julius uses a star rating scale that is very popular for e-commerce product reviews.

Google's support assistant uses face emoticons that are very recognizable from online surveys.

Slack's AI search overview makes use of the existing emoticons that are already familiar to the users.

Fogg's behaviour model

When designing for a user behaviour, Fogg's model is a very useful guide. In a nutshell, three elements must converge at the same moment for a behavior to occur: motivation, ability, and a trigger. When a behavior does not occur, at least one of those three elements is missing.

Behaviour = Trigger x **A**bility x **M**otivation

The **trigger** is the presence of the reaction buttons. Since we want feedback for every response, the prompt for feedback needs to noticeable, yet subtle and not disrupt the user flow. In order to avoid fatigue, some apps prompt the user occasionally.

When the trigger for a reaction is presented, the user needs to have the **ability** to provide an assessment. This means having an answer already and a break in the flow to click on the buttons and eventually provide more details. Long responses or complex artefacts provide a challenge, because it takes time for the user to evaluate them and by that time the trigger is forgotten.

Motivation is the trickiest parts to get right in this case. Since the feedback needs to be collected, analyzed and later used to fine-tune the model, the effect of the action is far from immediate. It is also difficult to communicate to the user why this is needed, how it works and how it will benefit him. The "what's in it for me?" question does not have a simple answer. Gamification techniques or incentives could help increase the motivation to engage in this behaviour more often.