Amaresh Goshika

Mac Automator to do voice search in Perplexity

**Introduction:**

Mac Automator is a powerful automation tool provided by Apple that allows users to create custom workflows and automate various tasks on their Mac computers. One interesting application of Mac Automator is leveraging its capabilities to perform voice searches in Perplexity, a language model developed by OpenAI.

Perplexity is an advanced language model designed to understand and generate human-like text based on prompts or queries. By combining the natural language processing capabilities of Perplexity with the automation features of Mac Automator, users can create a seamless voice search experience on their Mac.

Using Mac Automator, you can design a workflow that captures voice input from the user, sends it to Perplexity for analysis and understanding, and then retrieves the generated text-based response. This enables you to interact with Perplexity using your voice, making the process more intuitive and efficient.

By employing this integration, you can perform a wide range of tasks such as asking questions, requesting information, or seeking creative suggestions. The power of voice input combined with the natural language processing capabilities of Perplexity opens up new possibilities for hands-free interactions and enhanced productivity.

In this tutorial, we will explore how to set up and utilize Mac Automator to create a voice search workflow with Perplexity. We will guide you through the necessary steps to capture voice input, send it to Perplexity for processing, and display or utilize the generated text output.

With Mac Automator and Perplexity, you can elevate your Mac experience by harnessing the power of voice and cutting-edge natural language processing. Let's dive in and unlock a world of possibilities with voice search in Perplexity using Mac Automator!

**Python Script:**

import speech\_recognition as sr

import webbrowser

def google\_search(query):

url = f"https://www.perplexity.ai/search?q={query}"

webbrowser.open(url)

# Function to listen for voice input

def listen():

recognizer = sr.Recognizer()

with sr.Microphone() as source:

print("Speak now...")

audio = recognizer.listen(source)

try:

print("Recognizing...")

query = recognizer.recognize\_google(audio)

print("You said:", query)

return query

except sr.UnknownValueError:

print("Sorry, I could not understand your voice.")

except sr.RequestError:

print("Sorry, my speech recognition service is currently unavailable.")

# Listen for voice input

query = listen()

# If the query is not empty, perform a Google search

if query:

google\_search(query)

**Steps to create shrortcut:**

1. Open Automator on your Mac. You can find it in the Applications folder or by using Spotlight search.
2. When Automator opens, select "New Document" or press ⌘+N to create a new workflow.
3. In the template chooser, select "Quick Action" and click "Choose."
4. At the top of the Automator window, set "Workflow receives current" to "no input" in "any application."
5. In the Library pane on the left side, search for "Run AppleScript" and drag it to the workflow area on the right.
6. In AppleScript action, set Shell to “/bin/bash”
7. In the AppleScript action, replace the default script with the following script:

#!/bin/bash

PYTHON\_INTERPRETER=”<enter\_path\_to\_python\_interpretor"

SCRIPT\_PATH="<enter\_path\_to\_python\_script"

$PYTHON\_INTERPRETER $SCRIPT\_PATH

1. Save the workflow by clicking "File" and then "Save" (or press ⌘+S). Give it a name like "Search in Google."
2. Close Automator.
3. Now, whenever you want to search something using voice input, select the text or activate the voice dictation on your Mac and speak the text.
4. After you've selected or dictated the text, right-click on it (or use the shortcut you set up for voice dictation) and choose "Quick Actions" from the context menu.
5. In the "Quick Actions" menu, you should see your "Search in Google" workflow. Click on it.
6. Safari will open with the Google search results for the text you provided.

**Reference:**

“Use a Shell Script Action in an Automator Workflow on Mac.” *Apple Support*, support.apple.com/guide/automator/use-a-shell-script-action-in-a-workflow-autbbd4cc11c/mac.