Podcast summarization tool

Project description

The New York Times has a tool that summarizes podcasts (see for instance How The New York Times uses a custom AI tool to track the "manosphere" | Nieman Journalism Lab). This project is about building a similar tool. Employees would subscribe to podcasts in the summarizer. They would receive summaries, with links to the actual content.

This tool is an AI tool because it summarizes an audio recording just as a person would, using AI to transcribe and summarize. Further AI tools could be used to write some of the code, including prototyping the user interface.

Use case

Business context:

Visibility: The tool is internally facing. However, some summaries may lead to stories that will be reported to the public.

Release and Legal Status of media content: By summarizing podcasts, the tool uses non-company IP that, while publicly available, may have some protection. It creates what may be considered a derivative product for internal use, which again could result in external-facing material. However, it replicates an existing workflow (listening and summarizing a podcast) and makes it more efficient. The Inputs sub-category below states the release and legal status of that content precisely.

The project has both a **commercial and non-commercial nature**. Given the low cost of summarizing a podcast (few cents), we envision making the tool accessible to all employees as a benefit. But the primary use case is to speed up surfacing news stories from podcasts.

The **business benefit** is clear: it allows journalists to broaden their sources, collect more information faster, and sense the pulse of a community more accurately, enabling easier and deeper reporting on any topic, from sports to arts to health to politics.

We estimate that building a POC to demonstrate

- Tracking new podcast episodes,
- Downloading the new episode,
- Summarizing the new episode,
- Presenting it in a simple UI to a user

will take less than 12 hours, including some user feedback.

Should this POC be successful, we will need about

- 20 hours to design the backend and how the tool will work (data schema, UI mockups),
- 160 hours of engineering (2 weeks, 2 engineers) to have a running product on which we could iterate depending on user feedback.

Stakeholders: The tool is primarily built for the various news departments.

The **primary users** are journalists.

Approval context:

Approval subject: We are seeking approval for:

- Doing the POC
- Building the tool
- Using AI speech to text technology
- Using LLM for summarization

AI for coding seems out of scope given that we are planning to use tools approved at the company level.

AI in Scope: the speech to text and summarization models behind those two capabilities. Their origin and training data are described under AI provenance below.

In the build phase, the output will be reviewed by some of the stakeholders to validate the format of summaries (initially, a few bullet points, 500 to 1,000 words, with exact quotations supporting the bullet points, and a link to the actual audio). After that initial period, stakeholders will be able to give feedback and possibly customize the output to some extent. If legal review is needed for these customizations, then we will need to define the proper workflow.

Inputs

- Audio or video files of podcasts available to the public on podcast platforms. From a legal perspective, we need clarification on:
    - Can the platform be accessed by automated means?
    - Can the content be downloaded from the platform (vs. streamed)?
    - Are we allowed to summarize it?
    - Are we allowed to use the summaries, which could be considered derived work, to do some reporting? But is this different from a human taking notes and using some of the material for reporting?
- **Input Types**: audio dialogue, and video where only a video file is published.
- **Release Status**: not applicable; the podcasts are third-party content already published rather than in-house material.
- **Legal Status**: copyrighted, and likeness-protected in the voices the recordings carry.
- **Legal Subject**: the podcast recording itself and the voices of the identified participants.

At this point we do not know if we will store the input (audio/video file). If we can point to the actual content at a given timestamp, our intent is to not store the input.

Output

Output type:

- The output contains:
    - Title and episode of the podcast,
    - Links to the actual podcast,
    - Summary of the podcast episode
    - Transcription of the episode with associated timestamps
    - Metadata related to the podcast (url of rss feed, last episode downloaded).

The output will be stored in databases and delivered by a UI in HTML format. Furthermore, it may be stored locally by users.

Output novelty: The output is a summary of the input. Therefore, it is derivative, yet as a summary it is creative in its choice of words and what to emphasize.

Intended output role: The output is intended to be read by employees who will use it as source of information for possible reporting to the public. It is no different from the current use of podcasts, albeit using a more efficient and automated workflow.

AI Interaction: The output is controlled by the prompt, which does not dictate in detail how the content is summarized but controls well how the summary is structured and how many words and bullets it uses. While this lack of control may result in inaccurate summaries, we anticipate that users will listen to the actual content before using the material for reporting purposes. Hence, the risk associated with inaccuracy is negligible in terms of presenting accurate information to the public. The risk lies mostly in internal users finding the tool useless if the summaries are not helpful.

In the future we may allow users some options for the length and structure of the summary.

Final use potential: The output will not be available directly to the public, but will be

- stored in our podcast summarization database for at least 1 year
- could be stored by journalists and editors as supporting material for their stories.

Talent likeness: The output will contain quotations from the podcast, which may not be attributed since we will not attempt voice identification. It is anticipated that employees will listen to the actual podcast and do the proper attribution should they quote one of the participants.

Brand perception: It will follow the current attribution practice.

Tool used

Vendor profile

We are planning to use tools for which we already have an MSA or that open source with licensing allowing for commercial applications. For tools with an MSA, we are planning to use them through API, while for those open-source ones, we are planning to run the software on our infrastructure.

AI provenance

Model provisioning

All models were trained by vendors/providers. The sources of the data used to train these models are unknown, though the data was likely taken from the internet.

Operational characteristics

AI deployment

For tools from vendors, refer to specific MSA as to where they are deployed.

For open source tools, we plan to deploy them on our laptops for prototyping and on our cloud infrastructure.

Customer data residency

The podcast audio/video file residency will depend on the specific podcast. But to start with, we should assume that podcasts would be obtained from podcast platforms. However, if these present legal challenges, then we could seek alternatives.

The backend for our application will be on our cloud infrastructure, which, to my understanding, means US based.

Data Collection: refer to specific MSA for whether customer data (prompts, input audio, generated summaries, workflow configurations) is collected, the Collection Retention Period, and the Vendor Use of Collected Data.

Metadata collection

Refer to specific MSA for what metadata is collected by vendors, how long it is retained, and what the vendor uses it for.

Model refinement

We anticipate that as models evolve and are phased out, we will need to update the models used. We may also update the prompt used to run these models.