

# WIOMSA AI Workshop: Prompts

Smit, A. J.

University of the Western Cape

## Table of contents

1	1.3 Orientation and Demonstration .....	1
1.1	Voice to text prompt (lectures) .....	1
1.2	Use AI to generate the prompt .....	2
2	1.4 Prompt “Engineering” .....	3
2.1	ChatGPT (or other LLM) personalisation .....	3
3	1.5 Structured Extraction .....	4
3.1	Prompt “engineering” .....	4
4	1.6 Validate Species Names Against an Authoritative Source .....	5
5	1.7 Access GBIF and IUCN Red List Distribution and IUCN Status Records .....	6
6	1.9 Parse Survey Responses .....	6
	Bibliography .....	7

## 1 1.3 Orientation and Demonstration

### 1.1 Voice to text prompt (lectures)

The prompt here was used to take my spoken words and translate it to the paragraphs about voice to text that you see immediately above.

#### GENERAL:

- Use British English consistently and religiously.
- Please transcribe the my voice, keeping more or less my mode and style of speaking intact.
- The intention is to maintain a style of writing that closely mirrors my natural way of speaking.
- Apply corrections to ensure my grammar and language are clear and correct after translation to text.
- Use proper paragraphs, and apply punctuation liberally.
- Apply strict fact-checking. Indicate, where necessary, where the factual material that I talk about is clearly incorrect. Insert a pointer such as 'attention' in square brackets next to the statement that has some doubt associated with it.
- The audience is a scientifically-trained group of people (staff and students at my university).
- The intended use of the material will be to serve as a faithful reproduction of my lecture content (typically the content of my narrative, but also sometimes to write emails and so on).

- Translate any numbers with units or math to LaTeX math and wrap the command in `\$ ... \$` for use in Quarto. E.g.,  $2,500\text{--}3,000 \mu\text{mol m}^{-2} \text{s}^{-1}$  becomes `$(2{,}500\text{text\{--\}}3{,}000\mu\mathit{mol},\mathit{m}^{-2},\mathit{s}^{-1})$`.

#### NOTES ON FORMATTING:

- If headings are required, please start with the highest-level heading (#) that has the name of the transcribed file, such as "# Lecture Transcript: Plant Stresses", omitting any reference to the model name or lecture number.
- Insert deeper level headings (## and ###) as necessary to add some structure to the textual content.

#### IMPORTANT:

- Don't add any embellishments, such as acknowledging my request or conclusion statement. Simply return the transcribed text.

## 1.2 Use AI to generate the prompt

One can use AI to help generate prompts. Here is an example of a prompt that should recreate something similar to what we have seen above:

Please create a prompt that I may give to an AI tool capable of voice to text conversion that would take a voice memo of a recorded lecture and convert it to the meeting's minutes.

#### INSTRUCTIONS:

1. Provide a professional, comprehensive yet concise (not verbose) reporting of the minutes
2. Use British English consistently and religiously.
3. Please transcribe the voice, keeping more or less the mode and style of speaking
4. Identify any dubious factual information.
5. Anything else you think is important to include in the prompt.

This generates:

#### Prompt for AI Transcription Tool

Transcribe the following recorded lecture into a written transcript that:

1. Accuracy and Style
  - Transcribes the spoken words faithfully, preserving the speaker's academic tone and style without paraphrasing.
  - Use British English consistently for spelling, punctuation, and grammar.
  - Retain key pauses, emphases, and rhetorical flourishes where they

matter for meaning, but smooth over fillers (e.g. “um,” “you know,” “like”) unless they are integral to the speaker’s argument or style.

2. Formatting and Structure

- Present the transcript in clear, readable prose.
- Break the text into paragraphs that reflect shifts in topic, argument, or emphasis.
- Insert time stamps at reasonable intervals (e.g. every 5 minutes) and at points where topics change significantly.
- Indicate slide references or visual cues (e.g. [Slide 7 shown]) if audible.

3. Factual Integrity

- Where the speaker provides information that seems dubious, uncertain, or factually questionable, flag it clearly in square brackets (e.g. [Possible error: date inconsistent with established record]).

4. Additional Annotations

- Note instances where the speaker references literature, quotations, or data that may require citation.
- Capture audience questions distinctly from the lecturer’s voice, marking them as “Audience:” and “Lecturer:” in dialogue form.
- If parts of the recording are inaudible or ambiguous, mark them as [inaudible] or [unclear], not guessed.

5. Output Requirements

- Deliver the transcript in a professional academic style suitable for archival use, research, or teaching reference.
- Do not summarise; provide the lecture in full, while maintaining readability.

This may be “compressed” for ease of use – note that the structure seen in the above prompt aids the human, not the AI. The compressed version looks like this:

Transcribe this lecture accurately in \*\*British English\*\*, preserving the speaker’s academic tone and style without paraphrasing. Keep the transcript professional, clear, and readable: structure it into paragraphs reflecting topic shifts, add time stamps at sensible intervals, mark slide references if mentioned, and distinguish audience questions from the lecturer’s voice. Retain rhetorical emphasis but omit filler words unless essential. Flag any dubious or factually questionable information in square brackets, note references to literature or data for possible citation, and mark inaudible or unclear sections as [inaudible] or [unclear] rather than guessing. The output should be a full, faithful transcript suitable for academic use, not a summary.

## 2.1.4 Prompt “Engineering”

### 2.1 ChatGPT (or other LLM) personalisation

Use the following prompt to personalise the AI to your own style of writing:

Aim to be scholarly, confident, and analytical, appealing to readers accustomed to advanced academic dialogue. Maintain a poised authority, weaving historical insight/philosophical depth without slipping into empty verbosity. While the vocabulary reflects complexity—e.g. “epistemic,” “conceptual ordering,” “structured inference,” and “rigorous standard”—do not use jargon for its own sake. Emphasise clarity that respects the reader’s intelligence and allows concepts to resonate without condescension.

Let the sentence structure shift between long, layered forms that contextualise, define, and critically engage, and shorter, sharper sentences that reinforce key arguments and points. Use a variety of clauses, parenthetical asides, and em-dashes to give the writing a dynamic flow. This rhythmic variation and precise diction shape a voice that rewards a close, attentive reading.

Avoid these words: particularly, crucial(ly), essential, holistic, especially, challenge(s), sophisticated, ensuring/ensure, profound, remarkable, nuanced, emerge(s), questioning, nudge(s), robust, “stand out,” “by acknowledging,” “It’s a reminder,” “In summary.”

Aim for a Gunning-Fog index above 23.

Avoid words that flatten complexity or imply hollow emphasis. Rely on carefully chosen terms that reflect a style suited to readers ready to engage with advanced scholarly thought.

### 3 1.5 Structured Extraction

#### 3.1 Prompt “engineering”

Load the PDF of the “*UNEP Regional State of the Coast Report (WIO)*” into i) an LLM of your choice, and ii) into NotebookLM. Then, apply the following prompt.

Please provide me with a list of fish species mentioned in the report I uploaded. Combine the data from the general report with that in Tables 10.A2 and 10.A3 into the CSV file. In the following columns, add the common names, Family, and distribution identified in the report. Make available as a CSV file for download (sandboxed, if necessary).

Add the IUCN status for each species, using the following prompt:

also include the IUCN Red List status (e.g. Critically Endangered, Endangered, Vulnerable) as an additional column in the CSV file. If a species is not listed, indicate this with “Not Evaluated” in the IUCN status column.

## 4 1.6 Validate Species Names Against an Authoritative Source

Select the list of species names generated in the previous step, and use the following prompt to validate them against FishBase:

Please assess the fish species (pasted below) relative to FishBase and identify the application of the correct taxonomic names to the binomial names. Please also include some interpretive remarks that matter for dataset hygiene and integrity. Provide the output as a CSV file for download (sandboxed, if necessary).

Or...

Please produce a clean CSV that (i) standardises names to FishBase usage; (ii) adds a "WoRMS accepted" column for the handful of discordant cases (Stegostoma, the squid); and (iii) carries a provenance field indicating the authority used for each row. Please make available as a sandboxed CSV that I may download.

We can also join files on a common field, such as species name. Use the following prompt:

Please join these two CSV files on "Scientific Name" (in NotebookLM\_fish\_v2.csv) and "FishBase accepted" or "WoRMS accepted" (in Fish\_standardised\_FB\_WoRMS.csv). Make available as a sandboxed CSV file for easy download. Ensure that all data

Next we want a JSON schema that we can use in future data extractions from different sources to ensure that we get the same fields. Use the following prompt:

Please generate a JSON schema that describes the structure and content of the attached file. The JSON schema will be used in future searches to generate a compatible data table from different data sources, with the intent to eventually bind the tables by row.

Adding more data... this time IUCN Red List data from FishBase:

Now, add a second column for IUCN Red List status but, for each species, populate it with the status gleaned from FishBase, together with the date assessed in the next column.

And further:

Okay, use this JSON (keep existing data fields/descriptors) and expand it to include the IUCN data harvested from any PDFs read in and searched through, as well as the FishBase IUCN status and assessment date for each species, as explained above. The intention is that the JSON schema should be "self-

describing" such that new data sources can be seamlessly extracted in a consistent format and CSV structure when provided with a new PDF with fish names and associated data (and automatically query FishBase for the IUCN status data).

## 5 1.7 Access GBIF and IUCN Red List Distribution and IUCN Status Records

Example prompt to try and access GBIF:

Please access GBIF and produce a long CSV file with species names (*Pristis pristis*, *Rhincodon typus*, *Cheilinus undulatus*) and occurrences (by country), and present it as a sandboxed long CSV file for download.

## 6 1.9 Parse Survey Responses

Produce a "rough" initial prompt to refine:

Please see the following draft prompt and refine it for use in an AI to create the code to execute:

Take the input file @AI/workshop\_materials/

Cleaned\_Survey\_Data\_Translated\_and\_Processed\_Fixed.xlsx and do the following:

- \* take each line from Lines 1 to the end of the file and parse the data (per line) into separate TXT files
- \* use the column titles in row 1 for level 1 headings (e.g., "# Timestamp") in the CSV file, underneath which the cell's content is provided
- \* do this for each column, from A to BY
- \* three of the columns contain personal information; please remove these from the parsed final TXT files to ensure anonymity of respondents and other actors
- \* save each TXT file with the names `Gansbaai\_survey\_<row\_number>.TXT` in the directory @AI/workshop\_materials/parsed\_data/

The output will look similar to this:

Take the Excel file located at @AI/workshop\_materials/

Cleaned\_Survey\_Data\_Translated\_and\_Processed\_Fixed.xlsx and perform the following steps:

1. Read the file and iterate over each row, starting from row 2 (i.e., data rows beneath the header row) to the final row.
2. For every row, create a separate TXT file.
3. Each TXT file should:
  - Use the column titles from row 1 as level-1 style headings. For example:

```
# Timestamp  
2023-06-04 11:52:00  
  
• Present the data in two lines per column: the heading (prefixed with  
# ) followed by the corresponding cell value.  
• Include all columns from A to BY, unless removed under step 4.
```

4. Three columns contain personally identifiable information (PII).  
Exclude these three columns entirely from the output to preserve anonymity.
5. Save each per-row TXT with the filename pattern:  
`Gansbaai\_survey\_<row\_number>.txt`

Example: Gansbaai\_survey\_2.txt for the second row in the Excel file.

6. Save all generated TXT files in the directory: `@AI/workshop\_materials/parsed\_data/`

Requirements for the generated code:

- Handle UTF-8 encoding correctly.
- Ensure no loss of data types (dates remain dates, numbers remain numbers).
- Preserve ordering of the columns as they appear in the source Excel file.
- Make the list of excluded PII columns configurable at the top of the script (so the user can easily update it later).
- Use portable filesystem operations so the script runs on Windows, macOS, or Linux.
- Print progress messages while processing (e.g., “Saved row 25 to Gansbaai\_survey\_25.txt”).

We can do further analyses, *e.g.*, a sentiment analysis:

Take the newly generated TXT files and, for each, do an overall sentiment analysis. Is respondent positive, neutral, or negative in their outlook? Summarise the results in a new CSV file with the respondent number in column 1 and the sentiment in column 2. Place the new file in `@AI/workshop\_materials/parsed\_data/` .

## Bibliography