**DocxFormatterApp — User Guide**

This console app converts a raw Word document containing Reddit discussion excerpts into a clean, TTS-friendly transcript grouped by topic. It:

* Detects **topic titles** (and their URLs) and builds **entries** (everything from one title to the next).
* Distinguishes **“Someone wrote”** (quoted question/excerpt) from **“Krodha replied”** (your answer) using adaptive font-size and style cues.
* Optionally **reverses the entry order** for better listening flow.
* Exports a new .docx with active, clickable **hyperlinks** in the titles.

**1) Prerequisites**

* **.NET SDK** (6.0 or later recommended)  
  Check: dotnet --version
* **Open XML SDK** NuGet package: DocumentFormat.OpenXml
* A source **.docx** prepared from redditcommentsearch (see §3).

**Install & Build (CLI)**

dotnet new console -n DocxFormatterApp

cd DocxFormatterApp

dotnet add package DocumentFormat.OpenXml

# Replace Program.cs contents with your code

dotnet build

dotnet run

(Or use Visual Studio: create a Console App, add the NuGet package, paste the code, press F5.)

**2) What the app expects in your source .docx**

To correctly separate quoted text (asker) from your reply:

* **Paste with “Keep Source Formatting”** from redditcommentsearch <https://redditcommentsearch.com/>.  
  This preserves the differing **font sizes** and sometimes Word’s **Quote** styles.
* Each **topic title** should either:
  + Have **green** coloring (any run in the title line), *or*
  + Contain a **hyperlink** (modern Word hyperlink or a legacy field code like HYPERLINK "http://..." \hTitle).
* Optional: Insert a line with **Done!** wherever you want to force-end an entry (acts as a hard separator).

Tip: If your quotes are visually styled as **Quote**/**Intense Quote** in Word, the app detects that style too.

**3) How the detection works (in plain English)**

**3.1 Titles**

A paragraph is treated as a **Topic Title** if:

* It looks **green**, or
* It contains a **hyperlink** (either modern <w:hyperlink …> or legacy field-code).  
  If the field code has \hSome Link Text, that becomes the visible title.

The app **starts a new entry** at each title and closes the previous one.

**3.2 “Someone wrote” vs “Krodha replied”**

* **Two-pass approach**:
  1. The app scans all non-title paragraphs and builds a histogram of font sizes.
  2. It picks the **two most common sizes** and treats the **larger one** as the **“quote / someone wrote”** size (dynamic per document).
* For each paragraph inside an entry:
  1. If its **dominant run size** ≥ the detected **quote size**, or its style name contains **“quote”**, it’s a **Quote** (first paragraph in the entry becomes **“Someone wrote”**, subsequent large ones become **“Quote”**).
  2. Otherwise, it’s **“Krodha replied”**.

“Dominant size” = the **largest** font size among runs in that paragraph (with fallbacks to style size, then 12pt default).

**4) Running the app (what you’ll see)**

When you run dotnet run, you’ll be prompted:

1. **Path to the original DOCX**  
   e.g., C:\Users\you\Desktop\source.docx
2. **Path to the output DOCX**  
   e.g., C:\Users\you\Desktop\formatted.docx
3. **Username for replies** (default **Krodha**)  
   This label is used for reply sections: “Krodha replied:”
4. **Order selection**
   * **A = Ascending**: Oldest entry first → newest last.  
     Good for TTS, so you hear the original question before later follow-ups.
   * **D = Descending**: Newest entry first → oldest last (the way you usually collect them).

You’ll also see **DEBUG** lines that show what the app is detecting (paragraph size, whether it’s a title, etc.).

**5) Output format (what the new DOCX looks like)**

For each entry:

Topic Title: <clickable title text>

Someone wrote:

<first large paragraph (the quoted question/excerpt)>

Krodha replied:

<your reply paragraphs>

Quote:

<any later large quoted bits, if present>

----------------------------------------

* The **title text is a hyperlink** if a URL was found in the original title line.
* Sections are **merged** when multiple consecutive paragraphs share the same role (reply or quote).

**6) Troubleshooting**

**Blank output or very few entries**

* Titles were not recognized. Make sure title lines are **green** or **contain a hyperlink**.  
  If the site changed formatting, you can manually make the title line green or add a hyperlink.

**All text labeled “Krodha replied”**

* The quoted text may not be larger than your reply in that document. Ensure “Keep Source Formatting” preserves sizes, or apply Word’s **Quote**/**Intense Quote** style to quoted paragraphs.

**Wrong sections after pasting**

* Confirm you used **Keep Source Formatting**.
* Avoid pasting that flattens font sizes (e.g., “Merge Formatting” or plain text).

**Titles include raw HYPERLINK text**

* That was a problem in earlier versions; this code parses both modern and legacy field-code hyperlinks and should output a clean, clickable title. If you still see raw HYPERLINK, the title line likely wasn’t parsed as a hyperlink—try re-pasting the link in Word (Ctrl+K) or ensure it’s green so it’s still treated as a title.

**“WINWORD is running” warning**

* Close Word before running, or choose **Y** to force-close. Open Word again only after the app finishes writing the output file.

**7) Customizing (optional)**

* **Quote detection fallback**: If the app can’t calculate sizes, it falls back to **≥ 26 half-points** (13pt). You can adjust the fallback in DetectQuoteFontSize.
* **Green title colors**: Add hex codes to GreenHexCodes if your theme uses different greens.
* **Style-based quote detection**: The app treats any paragraph whose style name contains **“quote”** as a quote—even if sizes are close. You can extend IsQuoteStyled to include other style names you use.

**8) Known limitations**

* The app relies on **visual cues** (font size/style) to separate quotes vs replies. If your source has identical formatting for both, classification isn’t possible.
* Titles must be **discoverable** (green or hyperlink) to start entries. If your source has plain black titles with no links, add a link or make them green before running.

**9) Quick checklist for best results**

* ✅ Paste from redditcommentsearch with **Keep Source Formatting**.
* ✅ Ensure **each topic title** has a **link** or is **green**.
* ✅ Quoted text is **visibly larger** or styled as **Quote/Intense Quote**.
* ✅ Add **Done!** lines if you need extra separators.
* ✅ Close **Word** before running the app.

That’s it! If you hit a sticky example, share a few DEBUG: lines for those paragraphs and I can suggest tiny tweaks (e.g., adjusting the fallback size or adding a style flag).