We are building a clean dataset of transition examples from regional French news articles to support AI training and linguistic analysis. The goal is to extract short transitions from long narrative paragraphs using .docx source files, and structure the output into JSON, JSONL, and TXT formats.

This project involves writing a Streamlit app that parses the document structure, detects transitions, and generates structured outputs based on selected options.

You will create a working Streamlit app and Python scripts that:

- Accept .docx uploads
- Identify transitions embedded in long paragraphs using a list provided below each article
- Extract structured data around each transition (before/after context)
- Allow the operator to choose which output files to generate
- Save the selected outputs locally on the operator's machine

Your script must generate the following output files:

1. fewshot_examples.json: Extracted triplets with paragraph_a, transition, and paragraph_b (cap each transition at 3 uses)
2. fewshots_rejected.txt: List of transitions used more than 3 times with actual count
3. transitions_only.txt: Unique transitions, one per line, taken from the listed transitions under each article
4. transitions_only_rejected.txt: Transitions used more than once, with actual count
5. fewshot_examples.jsonl: JSONL format for fine-tuning, using structured messages with role:system, role:user, and role:assistant
6. fewshots-fineTuning_rejected.txt: List of transitions used more than 3 times in the fine-tuning format

The app must:

- Allow upload of .docx files
- Let the operator select which outputs to generate
- Display the number of extracted few-shot examples
- Save the selected files locally on the operator's computer

Each article in the Word file follows a fixed format:

1. Header with number and date
2. Title and blurb
3. Marker line: À savoir également dans votre département
4. One long paragraph containing embedded transitions
5. A final line listing 2–3 transitions actually used

The header, title, blurb, and marker line must be ignored. Focus only on the long paragraph after the marker. In the provided .docx sample, colors are used to orient the reader (green = fixed structure, yellow = transitions). These are not for programmatic use.

Your script must:

- Extract the long paragraph after the marker
- Use the transitions listed at the end of the article to detect and split that paragraph
- Produce structured examples: paragraph_a → transition → paragraph_b
- Cap each transition at 3 uses
- Optionally, you can implement broader detection using a master list of transitions (provided upon request)

You are also provided with a structured .txt file named ccm_raw_paragraphs_dump.txt to help you understand and test your logic. This file follows a consistent pattern that will remain the same across the project.

In this file:

- Paragraph 6 is the full narrative
- Paragraphs 7 to 10 list the transitions used in paragraph 6
- Paragraph 16 is the next narrative
- Paragraphs 17 to 20 list the transitions for paragraph 16
- And so on

Your script should work only on these narrative paragraphs (6, 16, 26, etc.), and for each, create 2 or 3 few-shot examples using the transitions listed in the next few paragraphs (7–10, 17–20, etc.). Once the examples for one paragraph are complete, the script moves on to the next narrative block.

Concrete output for paragraph 6:

{
"paragraph_a": "Ce mardi 6 mai 2025, à Nœux-les-Mines, une femme de 22 ans s'est immolée par le feu sur le parking du centre d’animation Loisinord...",
"transition": "Dans un tout autre registre,",
"paragraph_b": "le palais des congrès du Touquet-Paris-Plage accueillera une vente aux enchères de deux collections de mode..."
},
{
"paragraph_a": "Une vente d'accessoires, de foulards et de vêtements de grandes marques à des prix raisonnables sera aussi organisée.",
"transition": "Dans l'actualité sportive, sachez que",
"paragraph_b": "un nouveau complexe sportif portant le nom du président d'honneur du RC Lens, Gervais Martel, est en cours de construction à Houdain..."
},
{
"paragraph_a": "Les travaux ont débuté en présence de Gervais Martel en personne...",
"transition": "Enfin, nous apprenons que",
"paragraph_b": "un violent incendie s’est déclaré mercredi 7 mai dans l’entreprise Artois Métaux à Saint-Laurent-Blangy..."
}

Concrete output for paragraph 16:

{
"paragraph_a": "Chaque lundi, les élèves de l'école Olhette à Urrugne profitent de classes en plein air...",
"transition": "Dans un autre registre, on apprend que",
"paragraph_b": "le circuit Bellevue de Villefranque accueillera le classique de printemps du championnat Corac..."
},
{
"paragraph_a": "Entrée libre pour les spectateurs.",
"transition": "Si vous préférez des activités plus calmes, sachez que",
"paragraph_b": "le Groupe philatélique béarnais (GPB) présente une exposition de timbres à la villa Violette..."
},
{
"paragraph_a": "Le moment clé sera la création d'un timbre national avec La Poste...",
"transition": "Enfin, signalons que",
"paragraph_b": "l'édition 2025 de la grande fête des ikastolas, Herri Urrats, se déroulera ce dimanche 11 mai..."
}

You will be evaluated on two files:

- word10.docx: about 10 articles, should yield approximately 24 transitions
- word374.docx: about 374 articles, should yield around 900 transitions

Your implementation will be assessed based on:

- Accurate paragraph parsing
- Format compliance
- Transition frequency control
- App stability and output consistency

To apply, submit:

- A working Streamlit demo
- At least 20 extracted few-shot examples from word10.docx
- The ability to select which outputs to generate
- Local saving of all generated files

The file ccm_raw_paragraphs_dump.txt is provided to you as reference to help understand the input structure. This structure will remain consistent across the project.