from docx import Document

import nltk

import os

# 第一次使用需要下载分句器

nltk.download('punkt')

from nltk.tokenize import sent\_tokenize

def read\_docx\_sentences(file\_path):

"""读取 Word 文件并按句子分割"""

doc = Document(file\_path)

full\_text = "\n".join([para.text for para in doc.paragraphs])

sentences = sent\_tokenize(full\_text)

return [s.strip() for s in sentences if s.strip()]

def compare\_documents(file1, file2):

"""比较两个文档中完全相同的句子"""

sentences1 = set(read\_docx\_sentences(file1))

sentences2 = set(read\_docx\_sentences(file2))

common\_sentences = sentences1.intersection(sentences2)

return list(common\_sentences)

def generate\_report(common\_sentences, output\_path):

"""将相同句子写入一个新 Word 报告"""

report = Document()

report.add\_heading("文档比较报告", level=1)

report.add\_paragraph(f"相同句子总数：{len(common\_sentences)}\n")

for i, sentence in enumerate(common\_sentences, 1):

report.add\_paragraph(f"{i}. {sentence}")

report.save(output\_path)

if \_\_name\_\_ == "\_\_main\_\_":

file1 = "文档A.docx"

file2 = "文档B.docx"

output\_report = "比较报告.docx"

if not os.path.exists(file1) or not os.path.exists(file2):

print("请确保文档A和文档B存在")

else:

common = compare\_documents(file1, file2)

generate\_report(common, output\_report)

print(f"比较完成！发现 {len(common)} 个相同句子，报告已生成：{output\_report}")