from docxxx import Document

import nltk

import os

from nltk.tokenize import sent\_tokenize

# 下载 punkt 分句模型（只需一次）

nltk.download('punkt', quiet=True)

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)

def main(file1, file2):

"""主函数，接收两个文件路径参数"""

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

print("文件不存在，请检查路径是否正确。")

return

common = compare\_documents(file1, file2)

output\_report = "比较报告.docx"

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}")

# 示例调用方式

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

import sys

if len(sys.argv) < 3:

print("用法：python compare\_docx.py 文档1.docx 文档2.docx")

else:

main(sys.argv[1], sys.argv[2])