/extract_sizes.py

Given a filename, opens the PDF and extracts words and metadata from each slide.

:param file: String representing file path :type: string :rtype: dict :return: dictionary

```
def extract_words_word(file: str) -> list:
    """
    Given a filename, opens the Word and extracts words and metadata from each slide
    """



    basename = os.path.basename(file)
    template=f"soffice --headless --convert-to pdf {basename}"
    os.system(template)
    outputfile= basename[:-5]+".pdf"
    #convert(inputfile,outputfile)
    return extract_words(outputfile)
```

/ui.py

Takes input file name from the user through UI and when submitted generates the corresponding .apkg file

```
from tkinter import *

# import filedialog module

from tkinter import filedialog

from user_cli import *

from PIL import Image, ImageTk

def process_(file):

    lect_name = file.split("/")[-1].split(".")[0]

    raw_data = extract_words(file)
```

```
    raw_data = text_to_groupings(raw_data)
    keyword_data = wp.extract_noun_chunks(raw_data)
    keyword_data = wp.merge_slide_with_same_headers(keyword_data)
    keyword_data = wp.duplicate_word_removal(keyword_data)
    search_query = wp.construct_search_query(keyword_data)
    with concurrent.futures.ThreadPoolExecutor(max_workers=10) as executor:
        # when testing use searchquery[:10 or less].
        # Still working on better threading to get faster results
        results = executor.map(get_people_also_ask_links, search_query[:3])
    auto_anki_model = get_model()
    deck = get_deck(deck_name=lect_name)
    for result in results:
        for qapair in result:
            question = qapair["Question"]
            answer = qapair["Answer"]
            qa = add_question(question=f'{question}',
answer=f'{answer}',curr_model=auto_anki_model)
            deck.add_note(qa)
    add_package(deck, lect_name)
```

```
# Function for opening the

# file explorer window

def browseFiles():

    # file = filedialog.askopenfilename(initialdir="/",title="Select a File",filetypes=(("Text files","*.txt*"),("all files","*.*")))

    file = filedialog.askopenfilename(parent=window, title="Choose a file", filetypes=[("Pdf file", "*.pdf")])




    # Change label contents

    text_box = Text(window, height=10, width=50, padx=15, pady=15)

    text_box.insert(1.0, file)

    text_box.tag_configure("center", justify="center")

    text_box.tag_add("center", 1.0, "end")

    text_box.grid(column=0, row=3)

    process_(file)

# Create the root window

window = Tk()

canvas = Canvas(window, width=600, height=300)

canvas.grid(columnspan=2, rowspan=4)

# Set window title

window.title('Auto-Anki')

# Set window size

window.geometry("500x500")
```

```
# Set window background color

window.config(background="white")

#set logo

logo = ImageTk.PhotoImage(file='code/Auto_Anki_Logo.jpg')

logo_label = Label(image=logo)

logo_label.image = logo

logo_label.grid(column=0, row=0)

instructions = Label(window, text="Select a PDF file on your computer", font="Raleway")

instructions.grid(column=0, row=1)

button_explore = Button(window,

                    text="Browse Files",

                    command=browseFiles)

button_exit = Button(window,

                text="Exit",

                command=exit)

button_explore.grid(column=0, row=2)

button_exit.grid(column=0, row=3)

# Let the window wait for any events

window.mainloop()
```