```
import csv

import json

import os


from swarms.utils.pdf_to_text import pdf_to_text




def csv_to_text(file: str) -> str:

    """

    Converts a CSV file to text format.


    Args:

        file (str): The path to the CSV file.


    Returns:

        str: The text representation of the CSV file.


    Raises:

        FileNotFoundError: If the file does not exist.

        IOError: If there is an error reading the file.


    """

    with open(file) as file:

        reader = csv.reader(file)

        data = list(reader)

    return str(data)
```

```
def json_to_text(file: str) -> str:
    """
    Converts a JSON file to text format.

    Args:
        file (str): The path to the JSON file.

    Returns:
        str: The text representation of the JSON file.

    Raises:
        FileNotFoundError: If the file does not exist.
        IOError: If there is an error reading the file.

    """
    with open(file) as file:
        data = json.load(file)
    return json.dumps(data)


def txt_to_text(file: str) -> str:
    """
    Reads a text file and returns its content as a string.
    Args:
        file (str): The path to the text file.

    Returns:
        str: The content of the text file.

    Raises:
        FileNotFoundError: If the file does not exist.
        IOError: If there is an error reading the file.

    """
    with open(file) as file:
        data = file.read()
    return data


def md_to_text(file: str) -> str:
    """
    Reads a Markdown file and returns its content as a string.

    Args:
        file (str): The path to the Markdown file.

    Returns:
        str: The content of the Markdown file.
    Raises:
        FileNotFoundError: If the file does not exist.
        IOError: If there is an error reading the file.

    """
    if not os.path.exists(file):
        raise FileNotFoundError(
            f"No such file or directory: '{file}'"
        )
    with open(file) as file:
        data = file.read()
    return data


def data_to_text(file: str) -> str:
    """
    Converts the given data file to text format.

    Args:
        file (str): The path to the data file.

    Returns:
        str: The text representation of the data file.

    Raises:
        FileNotFoundError: If the file does not exist.
        IOError: If there is an error reading the file.

    Examples:
        >>> data_to_text("data.csv")
        'This is the text representation of the data file.'

    """
    if not os.path.exists(file):
        raise FileNotFoundError(f"File not found: {file}")
    try:
        _, ext = os.path.splitext(file)
        ext = (
            ext.lower()
        )  # Convert extension to lowercase for case-insensitive comparison
        if ext == ".csv":
            return csv_to_text(file)
        elif ext == ".json":
            return json_to_text(file)
        elif ext == ".txt":
            return txt_to_text(file)
        elif ext == ".pdf":
            return pdf_to_text(file)
        elif ext == ".md":
            return md_to_text(file)
        else:
            # Check if the file is a binary file (like an image)
            if ext in [".png", ".jpg", ".jpeg", ".gif", ".bmp"]:

                # Skip binary files

                return None

            else:

                with open(file) as file:

                    data = file.read()

                return data

    except Exception as e:

        raise OSError(f"Error reading file: {file}") from e
```