Resume Builder

Introduction

This is a resume builder that can help user facilitate their recruiting process. The builder takes user information as input and turns it into a formatted resume in PDF as output. The information will go through a Python script that converts the information into LaTex format via templates. Then the LaTex engine will be called to turn it into a PDF and return the file. We may provide a more user-friendly GUI version, but for simplicity, we may also do an easy-to-use command-line version that asks for information as CSV files and then outputs the PDF.

The project also includes an AI resume improver, which elaborates the wording and removes grammatical errors, and an AI resume rater, which rates the resume (currently only texts without format) and provides feedback on how to improve it. Both use GPT-3.5 as the model.

Getting Started

The Release Version

The Release version for Windows is now available at https://github.com/MikeChen012345/resume_builder/rele
You can download the latest version there without needing to go through some of the process below.

Due to the limitation of the system, it is not possible to use LaTeX without full installation. If you have not, get LaTeX at https://miktex.org/download. Make sure that you get the version that corresponds to your system and follow the installation instructions.

First, make sure that you have put your information correctly in all the CSV files in the `\data` folder. We have provided some example in the CSV files, so you can remove the example and fill in your own information in the same way.

Then, simply run the `.exe` file in the folder. You can find various buttons on the main window. Please go to the Using the GUI section for more information.

The Code Version

To start using the resume builder, simply clone the repository to your local repository by using the following command in your terminal:

```
git clone git@github.com:MikeChen012345/resume_builder.git
```

or

```
git clone https://github.com/MikeChen012345/resume_builder.git
```

Then, make sure that you have Python installed. If not, go to https://www.python.org/downloads/ to get the interpreter and follow the installation instructions. After this, go to your local repository and run the following command in your terminal to install all the required packages:

```
pip install -r requirements.txt
```

Due to the limitation of the system, it is not possible to use LaTeX without full installation. If you have not, get LaTeX at https://miktex.org/download. Make sure that you get the version that corresponds to your system and follow the installation instructions.

Now we can move on to build the resume!

Using the GUI

First, make sure that you have put your information correctly in all the CSV files in the `\data` folder. We have provided some example in the CSV files, so you can remove the example and fill in your own information in the same way.

Then, run the following command in your terminal to open the GUI:

```
python gui.py
```

The main window

You can find various buttons on the main window. This includes:

- Generate resume: generate the resume from the information that you have filled in the CSV files. You need to first enter a desired output filename in the “Output Filename” textbox. The generated resume will be saved in the `\output` folder. The current progress of the resume builder will also be saved in the `\saved` folder. There is also an option “Preserver LaTeX File” to preserve the transitional LaTeX file after the generation of the PDF. This is useful if you want to manually compile the PDF.
- Resume improver: improve the resume from the information that you have filled in the CSV files. The improved resume will be saved in the `\output` folder. The name will be “improved_resume_<time generated>”.
- Resume rater: rate the resume from the information that you have filled in the CSV files. You need to first enter the theme of your resume (i.e., what position the resume is for) in the “Input Text” textbox. The rating and the feedback will be shown in the “Feedback” textbox.

Note: during the generation, the software may be stuck due to waiting for responses from the url. Please wait patiently.

The side bar

You can find various buttons on the side bar. This includes:

- Fill .env file: fill in the .env file for the resume improver and rater. This includes the url (in case that you choose to use a 3rd-party AI model) and the API-key. You can get the API key at https://beta.openai.com/signup/, or you can get a free openai API at https://github.com/chatanywhere/GPT_API_free.
- Save resume builder: save the current progress of the resume builder. This includes the information that you have filled in the CSV files. You need to first enter a desired output filename in the "Output Filename" textbox. The saved file will be saved in the `\saved` folder and in the .pkl format.
- Load resume builder: load the saved progress of a chosen resume builder. This includes the information that you have filled in the CSV files. The saved file is in the .pkl format.
- Clean output directory: clean the output directory. This includes the PDF files that have been generated.
- Clean saved directory: clean the saved directory. This includes the saved progress of the resume builder (.pkl).
- Compile resume PDF: compile the resume PDF from a chosen .tex file. This is useful if you want to compile the PDF manually. You need to first enter a desired output filename in the "Output Filename" textbox and then choose the .tex file that you want to compile.
- Generate LaTeX file: generate the LaTeX file from the information that you have filled in the CSV files. You need to first enter a desired output filename in the "Output Filename" textbox. Then you are prompted to choose a LaTeX template file (.tex) that you want to use (The template can be filled with example information). The generated LaTeX file and the compiled PDF will be saved in the `\output` folder.

Note: the "Generate LaTeX file" feature is very unstable and may not work properly. Please use it with caution. You can retry or manually fix the syntax error in the generated .tex file and recompile using the "Compile resume PDF" feature.

Note: during the generation, the software may be stuck due to waiting for responses from the url. Please wait patiently.

The ReadMe window

You can find the ReadMe window at the right of the main window. This includes the ReadMe of the project. You can find the ReadMe of the project in the `\ReadMe.md` file.

Using the Commandline Interface

Note: We recommend you to use the GUI if you can as the commandline can be troublesome if you don't have enough experience. Also, several features are only available via the GUI.

First, make sure that you have put your information correctly in all the CSV files in the `\data` folder.

Then, run the following command in your terminal to generate the resume:

```
python builder.py (<output_filename>) (<preserve LaTeX file=False>)
```

The first parameter is the filename of the output PDF. The second parameter is whether to preserve the transitional LaTeX file. Note that if only one parameter exists, if the parameter is "True" or "False", then the parameter will be considered as the preserve_LaTeX_file flag. Otherwise, it will be considered as the output filename. If not specified, the default output filename will be "new_resume.pdf", and the default transitional LaTeX file will not be preserved after the generation of the PDF.

You might be asked to install some LaTeX dependencies for formatting. Just follow the instructions to install them all.

Note: the commandline versions of resume improver and resume rater only provide basic support. To fully utilize them, please use the GUI version.

To use the `resume_improver.py` and `resume_rater.py`, you first need to create a file named `.env` under the local repository. Then, add the following lines inside:

```
url = "<your openai API url>"

auth_key = <your openai API key>
```

The url parameter is designed as such to allow 3rd-party openai API. For example, you can get a free openai API at https://github.com/chatanywhere/GPT_API_free.

Then, simply run the two programs using `python resume_improver.py` and `python resume_rater.py`. You will be asked to provide your resume and a job description of the job that you are applying for. You can also fill in the your resume in the variable `content` and your job description in the variable `theme` (only for `resume_rater.py`).

The resume latex generator is also available via the commandline. After changing the parameters in `resume_latex_generator.py`, you can run the following command to generate the LaTeX file:

```
python latex_generator.py
```

Note: the "Generate LaTeX file" feature is very unstable and may not work properly. Please use it with caution. You can retry or manually fix the syntax *error in the generated .tex file and recompile using the "Compile resume PDF" feature.*

Note: during the generation, the software may be stuck due to waiting for responses from the url. Please wait patiently.

Trouble Shooting

I am using MacOS and the `python`/`pip` commands do not work for me

For MacOS, try changing `python` into `python3`, `pip` into `pip3` and leaving the other parts the same.

The resume builder fails to generate new resume. What should I do?

First, make sure that you have filled in the CSV files correctly. If you have, then the problem may be due to the LaTeX engine. Make sure that you have installed LaTeX and all the required packages. If you have, then the problem may be due to the syntax error in the LaTeX file. You can try to manually fix the syntax error in the generated .tex file and recompile using the "Compile resume PDF" feature.

I don't know how to install LaTeX on MacOS. What should I do?

Please follow the tutorial at https://miktex.org/howto/install-miktex-mac to install LaTeX on MacOS.

The resume improver fails to create new resume. What should I do?

The current resume improver can be unstable because sometimes GPT changes the format, causing the resume improver to fail to recognize the response. You can retry and it should be working within a few tries.

The application is stuck during the generation of the resume. What should I do?

The application may be stuck due to waiting for responses from the url. Please wait patiently.

The application does not work properly. What should I do?

Due to the usage of large language models, the response from the models can be unstable. Please retry the application and it should work within a few tries. For the “generate from template” or `resume_latex_generator.py` feature, you may also want to manually fix the syntax error in the generated .tex file and recompile using the “Compile resume PDF” feature.

Other issues/bugs

Please open an issue :) I will get back ASAP