Technical Interview

You are required to complete a coding challenge as part of your interview process for the PhD student role. This has been designed to showcase the following skills:

- Ability to extract entities from transcripts of conversations.
- Ability to use those entities to ascribe utterances to named speakers.
- Ability to identify limitations in generalising your approach to unseen transcripts.

The Context

The coding task asks you to extract names from a conversation and use those extracted names to match speakers to their utterances. We have given you a transcript of a brief conversation between four attendees at a meeting. The conversation consists of each attendee introducing themselves in different ways, while also referring to other people. Three of the attendees speak more than once.

Your task is to extract the individual names from each utterance in the transcript. You will then determine which name identifies each person and use that determination to replace **all** *"SpeakerX"* labels in the transcript.

To illustrate, consider the following example:

> *Speaker1: Hi, my name is John Smith.*
>
> *Speaker2: Hi, my name is Jane Brown.*
>
> *Speaker1: Nice to meet you, Jane.*

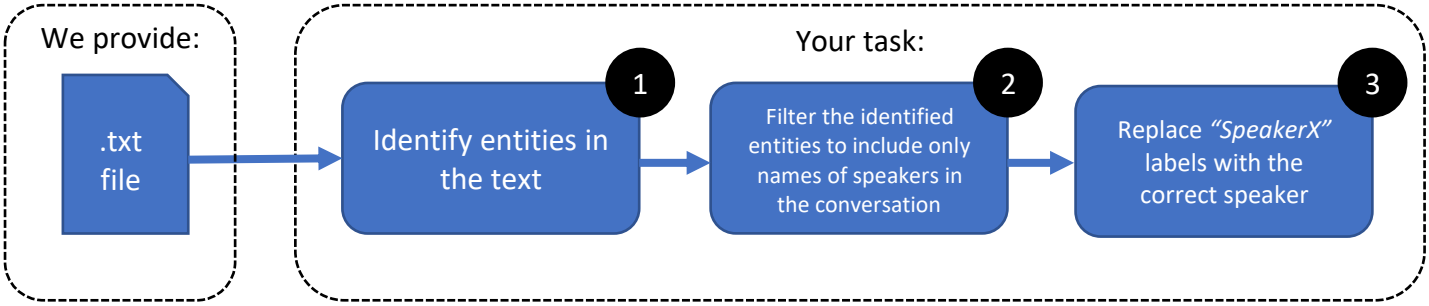
should become:

> *John Smith: Hi, my name is John Smith.*
>
> *Jane Brown: Hi, my name is Jane Brown.*
>
> *John Smith: Nice to meet you, Jane.*

This example has been simplified such that when introducing themselves each speaker says only their own name. In the transcript provided for this task, you will face the additional challenge of some speakers mentioning other names.

The Coding Task

You are required to develop code to complete the workflow illustrated below to create a transcript where speakers are labelled with the names determined from the text.



The following tasks are <u>essential</u> (~3 hours):

1. **Entity extraction from text**
    a. Create a function to extract text from the provided .txt file.
    b. Create a function to parse text and extract named entities. You may use libraries such as spaCy or equivalent if you feel it is appropriate.
2. **Speaker ascription**
    a. Create a function to filter the extracted named entities to include only the speakers in the conversation.
    b. Create a function to replace all *"SpeakerX"* labels with the correct speaker.
    c. Write the updated transcript to a .txt file.
3. **Identification of limitations**
    a. Your solution is required to work on the provided transcript but could also, in principle, work on any transcript. Write a README file (either as plain text or in Markdown) that identifies possible reasons why your approach might not work on a different, unseen transcript. This may include features of your approach, and/or potential characteristics of unseen transcripts.

Disclaimer

The use of suggested libraries is **not** a requirement. They have been suggested to give an idea of required functionality and the assumption the task is completed in Python. Please do use whatever coding language and libraries you feel comfortable with.

How to submit
Please submit your solution to the coding task by sending the link to a public repository (e.g. GitHub) to **m.snaith@rgu.ac.uk** by **31st January 2025, 11am GMT**.

If you would like any clarifications or have questions related to this task, please get in touch with Mark Snaith (m.snaith@rgu.ac.uk).