

Assignment 3: Question Answering via Reading Comprehension

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. It is a well-known NLP competition, and you can check it out via <https://rajpurkar.github.io/SQuAD-explorer/>. The top ranked solutions are also available here. For a quick review, you can read [this introduction](#).

In this assignment, you will have a chance to design your own question-answering model by using NLP + Deep Learning approach. Before you start, please check this video tutorial very carefully: [Applying BERT to Question Answering \(SQuAD v1.1\)](#) and the related document + code can be found at <https://colab.research.google.com/drive/1uSIWtJdZmLrI3FCNIIUHFxwAJiSu2J0->

You can download the dataset [here](#) or [here](#). Using a small subset is acceptable.

Download a copy of the dataset (distributed under the CC BY-SA 4.0 license):

Training Set v2.0 (40 MB)

Dev Set v2.0 (4 MB)

Task 1: Implement a question-answering model

1. You can find the official evaluation script [here](#). You only need to report a final F1 score of the model.
2. Optional. Follow the [tutorial](#) and submit your model and results to the official SQuAD leaderboard. (This is good for your resume.)

Task 2: Unanswerable question detection

Some questions can be unanswerable. For instance, the reference text is focusing on Massachusetts demographics information, but the question is “What’s the population of Indiana?” Then, this question can be answerable. Please read this paper for [SQUADRUN](#) dataset.

In this task, please propose a model to detect those unanswerable questions, e.g., binary classification. Note that you don’t need to implement your model. However, please give me your **hypothesis**, **motivation**, **model design**, and **key parameter settings** (if there is any). You can review related works, but I would like to read your own design.

Please provide **details** of your proposed model, an overall design figure is recommended.

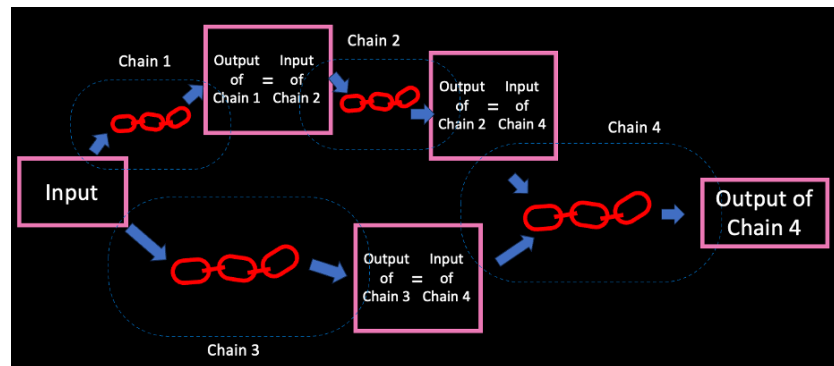
Task 3 [optional: 10 extra credit]: Implement your design from Task 2, and give me algorithm evaluation scores.

Task 4: LangChain practice.

Please review the LangChain lecture first before you start the task. You can download the slides (LangChain_with lecture voice.pptx) from Canvas. Please note that it's an **audio slideshow**, and you can click on the bottom right corner of each page to listen to the corresponding instruction. You can also find some tutorials regarding LangChain in the page 51 of the slides.

1. After you review the LangChain slide, please implement two sequential chain structures and submit the code.

- For the first sequential chain, here is the structure, related modules and sample output:



```
#For the first sequential chain
from langchain.chains import LLMChain
from langchain.chains import SequentialChain
from langchain.prompts import ChatPromptTemplate
from langchain.chat_models import ChatOpenAI
```

```
review = "Je trouve que la glace dans cette crèmerie est assez ordinaire. \
On sent un fort goût d'arômes artificiels, certaines glaces sont trop acides, \
voire pas très fraîches. Est-ce que le personnel ici triche sur la qualité et ne \
prend pas soin de faire des produits de qualité ?"
```

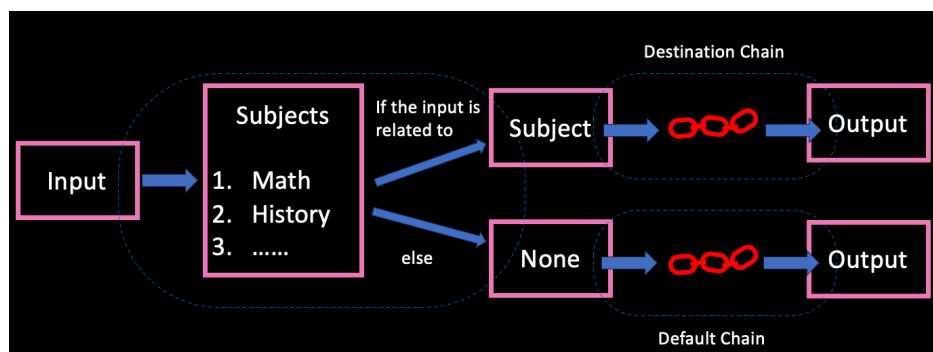
```
overall_simple_chain(review)
```

```
> Entering new SequentialChain chain...
```

```
> Finished chain.
```

```
{'review': "Je trouve que la glace dans cette crèmerie est assez ordinaire.          On sent un fort goût d'arômes a\nrtificiels, certaines glaces sont trop acides,          voire pas très fraîches. Est-ce que le personnel ici triche\nsur la qualité et ne          prend pas soin de faire des produits de qualité ?",\n  'language': 'French',\n  'English_review': 'I find that the ice cream at this creamery is quite ordinary. There is a strong taste of artifi\ncial flavors, some ice creams are too acidic, and not very fresh. Is the staff here cheating on the quality and not\ntaking care to make quality products?',\n  'summary': "This review criticizes the creamery for serving ice cream with artificial flavors, acidity, and lack o\nf freshness, questioning the staff's commitment to quality.",\n  'followup_message': "Nous vous remercions d'avoir partagé votre avis concernant notre crèmerie. Nous sommes désolé\ns d'apprendre que vous avez trouvé nos glaces avec des saveurs artificielles, de l'acidité et un manque de fraîcheu\nr. Nous prenons vos commentaires très au sérieux et nous allons prendre des mesures pour améliorer la qualité de no\ns produits. Votre satisfaction est notre priorité et nous espérons avoir l'opportunité de vous offrir une meilleure\nexpérience à l'avenir. Merci encore pour votre retour.",\n  'English_followup_message': 'Thank you for sharing your feedback about our creamery. We are sorry to hear that you\nfound our ice cream to have artificial flavors, acidity, and a lack of freshness. We take your comments very seriou\nsly and we will take steps to improve the quality of our products. Your satisfaction is our priority and we hope to\nhave the opportunity to offer you a better experience in the future. Thank you again for your feedback.'}
```

- For the second sequential chain, here is the structure, related modules and sample output:



```
#For the second sequential chain
from langchain.chains import LLMChain
from langchain.chains.router import MultiPromptChain
from langchain.chains.router.llm_router import LLMRouterChain, RouterOutputParser
from langchain.prompts import PromptTemplate
from langchain.prompts import ChatPromptTemplate
from langchain.chat_models import ChatOpenAI
```

```
res = chain.run("Who won Second World War?")
print(res)

res = chain.run("What is the sum of 27 plus 54?")
print(res)

res = chain.run("What is a black hole?")
print(res)

res = chain.run("How many stars in the sky?")
print(res)
```

```
> Entering new MultiPromptChain chain...
history: {'input': 'Who won the Second World War?'}
```

```
> Finished chain.
```

The Second World War was won by the Allied Powers, which included countries such as the United States, the Soviet Union, the United Kingdom, and other nations. The defeat of Nazi Germany and its allies, as well as the surrender of Japan in 1945, marked the end of the war and the victory of the Allies. The war was a complex and multifaceted conflict that involved numerous countries and theaters of operation, but ultimately the Allied Powers emerged victorious.

```
> Entering new MultiPromptChain chain...
math: {'input': 'What is the sum of 27 plus 54?'}
```

```
> Finished chain.
```

The sum of 27 plus 54 is 81.

```
> Entering new MultiPromptChain chain...
physics: {'input': 'What is a black hole?'}
```

```
> Finished chain.
```

A black hole is a region in space where the gravitational pull is so strong that nothing, not even light, can escape from it. This occurs when a massive star collapses in on itself, creating a singularity with infinite density and a gravitational field that distorts space and time around it. The boundary surrounding a black hole is called the event horizon, beyond which nothing can escape.

```
> Entering new MultiPromptChain chain...
None: {'input': 'How many stars in the sky?'}
```

```
> Finished chain.
```

It is estimated that there are around 100 billion stars in our Milky Way galaxy alone, and there are billions of galaxies in the universe. Therefore, the exact number of stars in the sky is difficult to determine, but it is safe to say that there are trillions upon trillions of stars in the observable universe.

2. Please design your own sequential chain structure and draw a diagram of it. Then, implement it according to your design and analyze the results.

Submission

You need to submit two separate files to Canvas: 1) your code; 2) your results doc.

1. Your code can be .py file or .ipynb file. For those who use Colab to complete this assignment, we would appreciate it if you could provide a shared link (General access: Anyone with the link can view it) to make it easy for us to review your code.
2. Your results doc can be a .pdf file or .docx file. The file needs to contain the results (e.g., figures, tables) of each task and your analysis.