


## ▼ Question Answering using NLP

```
!pip install html2text --quiet
!pip install simpletransformers --quiet
```

```
!pip install -U ipykernel
!pip install modin[dask]
```

```
import numpy as np # Math
import requests # Getting text from websites
import html2text # Converting wiki pages to plain text
from googlesearch import search # To performing Google searches
import re
from simpletransformers.question_answering import QuestionAnsweringModel
from IPython.display import display
from IPython.html import widgets # Graphical display
from bs4 import BeautifulSoup
from markdown import markdown
```

```
/usr/local/lib/python3.6/dist-packages/IPython/html.py:14: ShimWarning: The `IPython.html` package has been deprecated
  "`IPython.html.widgets` has moved to `ipywidgets`.", ShimWarning)
```



```
model = QuestionAnsweringModel('distilbert', 'distilbert-base-uncased-distilled-squad')
```

Downloading: 100%	451/451 [00:00<00:00, 13.5kB/s]
-------------------	---------------------------------

Downloading: 100%	265M/265M [00:03<00:00, 81.1MB/s]
-------------------	-----------------------------------

Downloading: 100%	232k/232k [00:00<00:00, 817kB/s]
-------------------	----------------------------------

```
# Testing for a single question
```

```
question_data = {
    'qas':
    [{ 'question': 'What color is the sky?',
        'id': 0,
        'answers': [{ 'text': ' ', 'answer_start': 0 }],
        'is_impossible': False}],
    'context': 'the sky is blue'
}
```

```
prediction = model.predict([question_data])
print(prediction)
```

```
convert squad examples to features: 100%|██████████| 1/1 [00:00<00:00, 278.10it/s]
```

```
add example index and unique id: 100%|██████████| 1/1 [00:00<00:00, 11275.01it/s]
```

```
Running Prediction: 100% 1/1 [00:00<00:00, 7.51it/s]
```

```
([{ 'id': 0, 'answer': [ 'blue', 'the sky is blue', 'sky is blue', 'is blue', 'the sky', 'the', 'sky', 'the sky is', ' ',
```

```
# Dividing the context into small context of size 512
```

```
def predict_answer(model, question, contexts, seq_len=512, debug=False):
```

```
    split_context=[]
```

```
    if not isinstance(contexts, list):
```

```
        contexts=[]
```

```
    for context in contexts:
```

```
        for i in range(0, len(context), seq_len):
```

```
            split_context.append(context[i:i+seq_len])
```

```
    split_context= contexts
```

```
    f_data=[]
```

```
    for i,c in enumerate(split_context):
```

```
        f_data.append(
```

```
            {
```

```
                'qas':
```

```
                [{ 'question': question,
```

```
                    'id': i,
```

```
                    'answers': [{ 'text': ' ', 'answer_start': 0 }]
```

```

        answers.append({'text': c, 'answer_start': 0}),
        'is_impossible': False}], # for unanswerable questions
    'context': c
    })
prediction = model.predict(f_data)
ans= prediction[0][0]['answer'][0]
prob= prediction[1][0]['probability'][0]
print("Answer: ",ans," Probability: ",prob)

```

```

predict_answer(model, 'king of mahishmathi?', ['On top of the waterfall, Shiva discovers that the mask belongs to Avanthika

```

```

convert squad examples to features: 100%|██████████| 1/1 [00:00<00:00, 193.13it/s]

```

```

add example index and unique id: 100%|██████████| 1/1 [00:00<00:00, 6462.72it/s]

```

```

Running Prediction: 100%

```

```

1/1 [00:00<00:00, 20.54it/s]

```

```

Answer: king Bhallala Deva/Pallvalathevan (Rana Daggubati) of Mahishmati. , Probability: 0.2554417723571843

```

## ▼ Sample Data for testing

### French and Indian War

The French and Indian War (1754–1763) was the North American theater of the worldwide Seven Years' War. The war was fought between the colonies of British America and New France, with both sides supported by military units from their parent countries of Great Britain and France, as well as Native American allies. At the start of the war, the French North American colonies had a population of roughly 60,000 European settlers, compared with 2 million in the British North American colonies. The outnumbered French particularly depended on the Indians. Long in conflict, the metropole nations declared war on each other in 1756, escalating the war from a regional affair into an intercontinental conflict.

### Oxygen element

Oxygen is a chemical element with symbol O and atomic number 8. It is a member of the chalcogen group on the periodic table and is a highly reactive nonmetal and oxidizing agent that readily forms compounds (notably oxides) with most elements. By mass, oxygen is the third-most abundant element in the universe, after hydrogen and helium. At standard temperature and pressure, two atoms of the element bind to form dioxygen, a colorless and odorless diatomic gas with the formula O<sub>2</sub>. Diatomic oxygen gas constitutes 20.8% of the Earth's atmosphere. However, monitoring of atmospheric oxygen levels show a global downward trend, because of fossil-fuel burning. Oxygen is the most abundant element by mass in the Earth's crust as part of oxide compounds such as silicon dioxide, making up almost half of the crust's mass.

## ▼ GUI based Output

```
text = widgets.Text(description='Question:', width=300)
display(text)

context_text = widgets.Text(description='Context:', width=300)
display(context_text)

button = widgets.Button(description='Get an Answer')
display(button)
context_input=[]
def on_button_click(b):
    context_input.append(context_text.value)
    print(predict_answer(model, text.value, context_input))

button.on_click(on_button_click)
```

Question: How many atoms combine to form

Context: Oxygen is a chemical element with

Get an Answer

convert squad examples to features: 100%|██████████| 1/1 [00:00<00:00, 65.87it/s]

add example index and unique id: 100%|██████████| 1/1 [00:00<00:00, 8577.31it/s]

Running Prediction: 100% 1/1 [00:00<00:00, 22.98it/s]

Answer: 8 , Probability: 0.9586549731592827

None

convert squad examples to features: 100%|██████████| 1/1 [00:00<00:00, 88.74it/s]

add example index and unique id: 100%|██████████| 1/1 [00:00<00:00, 11618.57it/s]

Running Prediction: 100% 1/1 [00:00<00:00, 25.36it/s]

Answer: 8 , Probability: 0.9304792258417508

convert squad examples to features: 100%|██████████| 2/2 [00:00<00:00, 111.95it/s]

add example index and unique id: 100%|██████████| 2/2 [00:00<00:00, 20815.40it/s]

Running Prediction: 100% 1/1 [00:00<00:00, 21.84it/s]

Answer: 8 , Probability: 0.9304792258417508

None

convert squad examples to features: 100%|██████████| 1/1 [00:00<00:00, 67.56it/s]

add example index and unique id: 100%|██████████| 1/1 [00:00<00:00, 8594.89it/s]

Running Prediction: 100% 1/1 [00:00<00:00, 19.94it/s]

Answer: oxygen , Probability: 0.4694070330191595

convert squad examples to features: 100%|██████████| 2/2 [00:00<00:00, 108.91it/s]

add example index and unique id: 100%|██████████| 2/2 [00:00<00:00, 16844.59it/s]

Running Prediction: 100% 1/1 [00:00<00:00, 21.38it/s]

Answer: oxygen , Probability: 0.4694070330191595

convert squad examples to features: 100%|██████████| 3/3 [00:00<00:00, 116.18it/s]

add example index and unique id: 100%|██████████| 3/3 [00:00<00:00, 34100.03it/s]

Running Prediction: 100% 1/1 [00:00<00:00, 18.17it/s]

Answer: oxygen , Probability: 0.4694070330191595

None

convert squad examples to features: 100%|██████████| 1/1 [00:00<00:00, 107.45it/s]

add example index and unique id: 100%|██████████| 1/1 [00:00<00:00, 5729.92it/s]

Running Prediction: 100% 1/1 [00:00<00:00, 23.61it/s]

Answer: two , Probability: 0.685902042079051

convert squad examples to features: 100%|██████████| 2/2 [00:00<00:00, 112.28it/s]

convert squad examples to features: 100%|██████████| 2/2 [00:00<00:00, 112.38it/s]

add example index and unique id: 100%|██████████| 2/2 [00:00<00:00, 15947.92it/s]

Running Prediction: 100%

1/1 [00:00<00:00, 23.18it/s]

Answer: two , Probability: 0.685902042079051

convert squad examples to features: 100%|██████████| 3/3 [00:00<00:00, 111.24it/s]

add example index and unique id: 100%|██████████| 3/3 [00:00<00:00, 28532.68it/s]

Running Prediction: 100%

1/1 [00:00<00:00, 21.59it/s]

Answer: two , Probability: 0.685902042079051

convert squad examples to features: 100%|██████████| 4/4 [00:00<00:00, 128.59it/s]

add example index and unique id: 100%|██████████| 4/4 [00:00<00:00, 24708.71it/s]

Running Prediction: 100%

1/1 [00:00<00:00, 22.81it/s]

Answer: two , Probability: 0.685902042079051

None