```
import torch
import seaborn as sns
import matplotlib.pyplot as plt
from transformers import BertTokenizer, BertModel

# Load pre-trained BERT model and tokenizer
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
model = BertModel.from_pretrained("bert-base-uncased",
output_attentions=True)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/
_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(
```


```
{"model_id":"e7ff778440104c89bb338f640264482d","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"9e977434473d4786bbe9a88978857949","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"b8aad03438e447539a75d97eea7cb97e","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"e3cd1358c70045f7bd1dee1a69cf7e22","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"2f00ad4abb174a409acaeafb892d79ab","version_major":2,"vers
ion_minor":0}
```


```
# Medical Note (Example 1)
text = "Patient reports persistent cough, high fever, and difficulty
breathing for the past three days."
inputs = tokenizer(text, return_tensors="pt")

# Get model outputs with attention weights
with torch.no_grad():
    outputs = model(**inputs)
    attentions = outputs.attentions  # Extract attention scores
```

```
BertSdpaSelfAttention is used but
`torch.nn.functional.scaled_dot_product_attention` does not support
non-absolute `position_embedding_type` or `output_attentions=True` or
`head_mask`. Falling back to the manual attention implementation, but
specifying the manual implementation will be required from
```

```
Transformers version v5.0.0 onwards. This warning can be removed using
the argument `attn_implementation="eager"` when loading the model.

# Select last layer's attention
last_layer_attention = attentions[-1].squeeze().mean(dim=0).numpy()

tokens =
tokenizer.convert_ids_to_tokens(inputs['input_ids'].squeeze().tolist()
)

# Plot attention heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(last_layer_attention, xticklabels=tokens,
yticklabels=tokens, cmap="Blues", annot=False)
plt.xlabel("Tokens")
plt.ylabel("Tokens")
plt.title("Attention Heatmap for Medical Note")
plt.show()
```