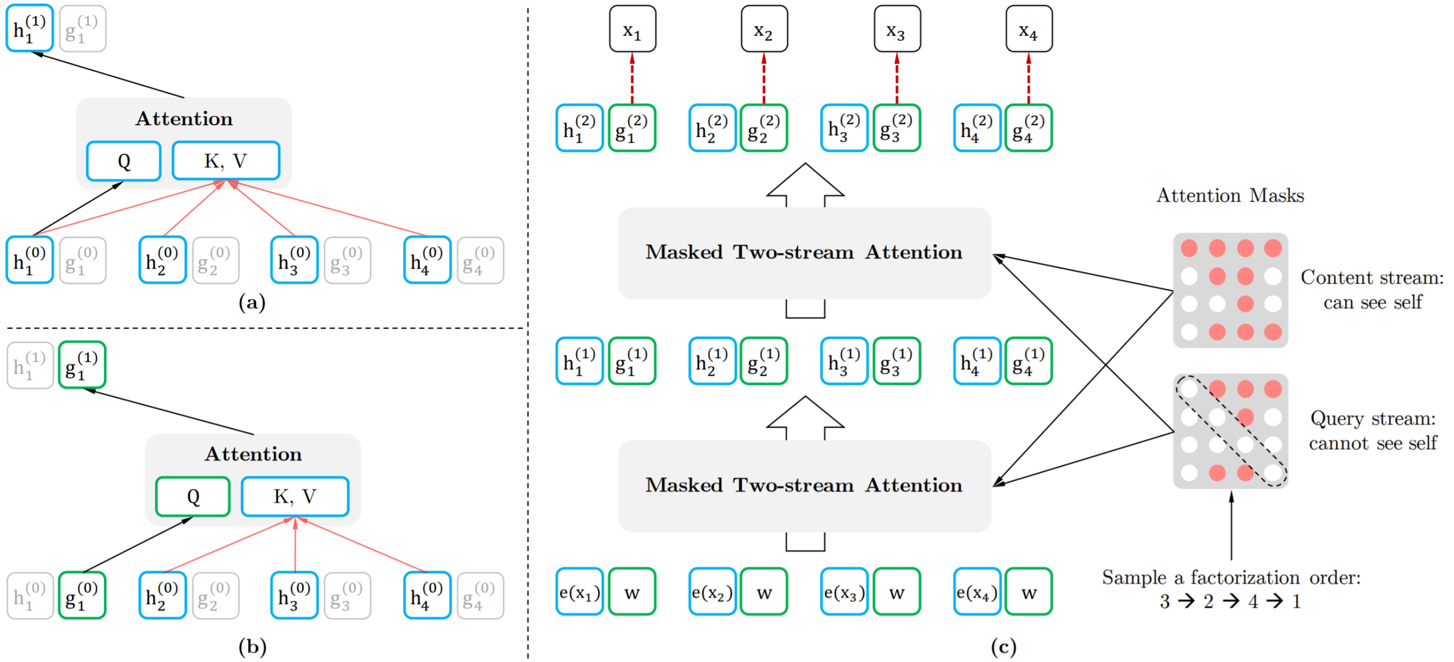
Cloud Computing Final Project Demo

Team 08

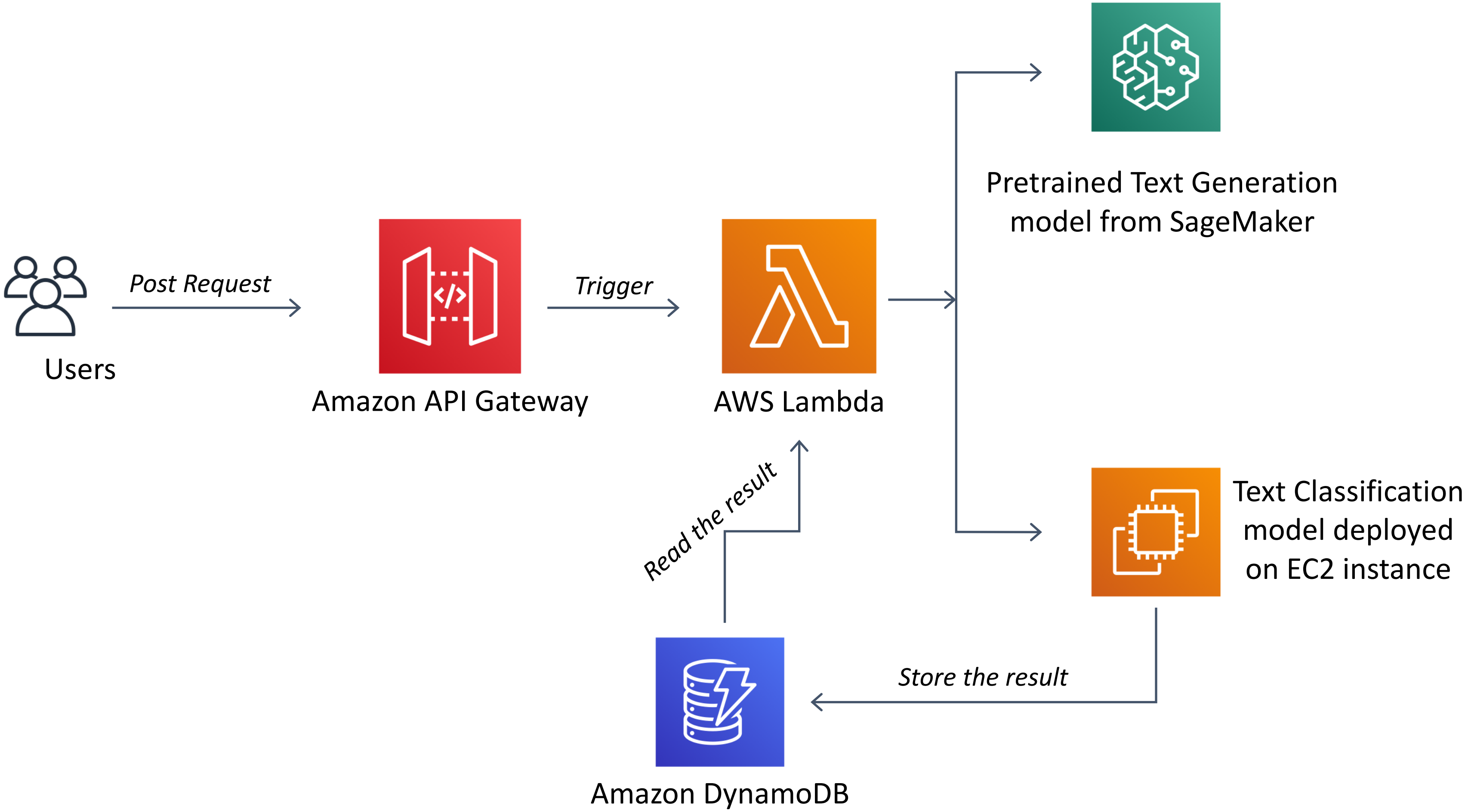
Model description (Fine tuned XLNet)

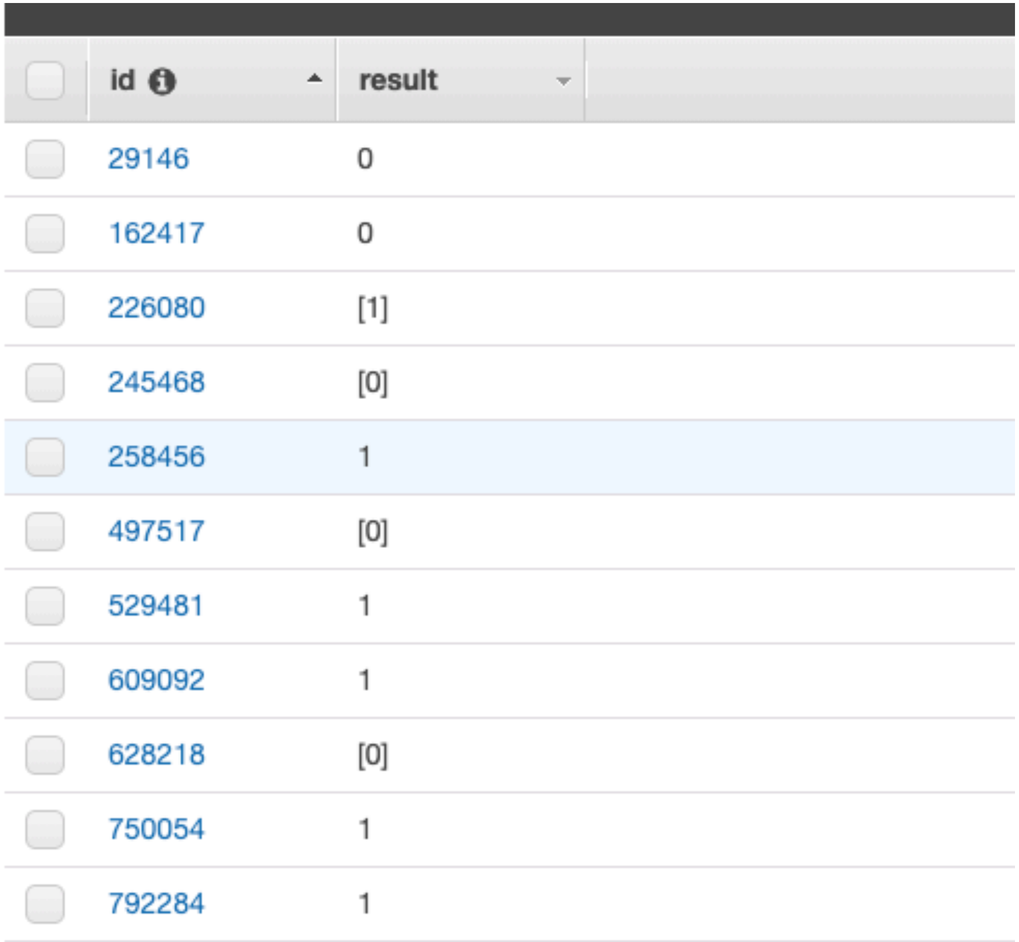


$h_1^{(1)}$ $g_1^{(1)}$
Attention
Q
K, V
$h_1^{(0)}$ $g_1^{(0)}$ $h_2^{(0)}$ $g_2^{(0)}$ $h_3^{(0)}$ $g_3^{(0)}$ $h_4^{(0)}$ $g_4^{(0)}$
(a)
$h_1^{(1)}$ $g_1^{(1)}$
Attention
Q
K, V
$h_1^{(0)}$ $g_1^{(0)}$ $h_2^{(0)}$ $g_2^{(0)}$ $h_3^{(0)}$ $g_3^{(0)}$ $h_4^{(0)}$ $g_4^{(0)}$
(b)
x_1 x_2 x_3 x_4
$h_1^{(2)}$ $g_1^{(2)}$ $h_2^{(2)}$ $g_2^{(2)}$ $h_3^{(2)}$ $g_3^{(2)}$ $h_4^{(2)}$ $g_4^{(2)}$
Masked Two-stream Attention
$h_1^{(1)}$ $g_1^{(1)}$ $h_2^{(1)}$ $g_2^{(1)}$ $h_3^{(1)}$ $g_3^{(1)}$ $h_4^{(1)}$ $g_4^{(1)}$
Masked Two-stream Attention
$e(x_1)$ w $e(x_2)$ w $e(x_3)$ w $e(x_4)$ w
Attention Masks
Content stream: can see self
Query stream: cannot see self
Sample a factorization order: 3 → 2 → 4 → 1
(c)

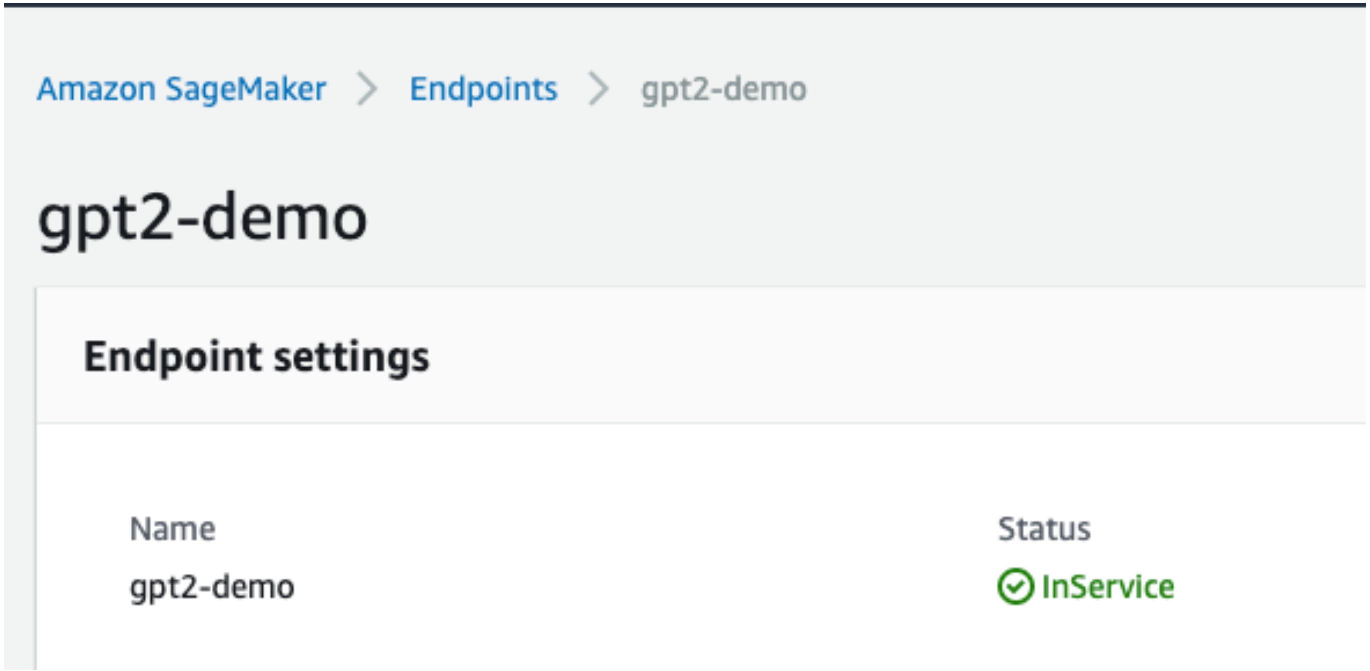
Figure 2: (a): Content stream attention, which is the same as the standard self-attention. (b): Query stream attention, which does not have access information about the content x_{z_t}. (c): Overview of the permutation language modeling training with two-stream attention.

System Structure (Ideal)





□	id	result
□	29146	0
□	162417	0
□	226080	[1]
□	245468	[0]
□	258456	1
□	497517	[0]
□	529481	1
□	609092	1
□	628218	[0]
□	750054	1
□	792284	1



Amazon SageMaker > Endpoints > gpt2-demo

gpt2-demo

Endpoint settings

Name
gpt2-demo

Status
✓ InService

Q1: What’s the purpose of the database?

- EC2 instance is not always available.
- The Lambda would first need to **turn on** the instance when requests arrive, and then sends a shell script to the instance (through SSM client).
- The script would
  - Activate the proper environment
  - Run the classification script
- The script will store the result in DynamoDB with a key from the Lambda.

Q2: Why not perform classification inside the Lambda

- XLNet is a HUGE model, which cannot easily fit into Lambda’s limited space.

- Deploying `PyTorch`, `Transformers` on Lambda has some tricky dependency problems.

- Lambda does not have GPU.

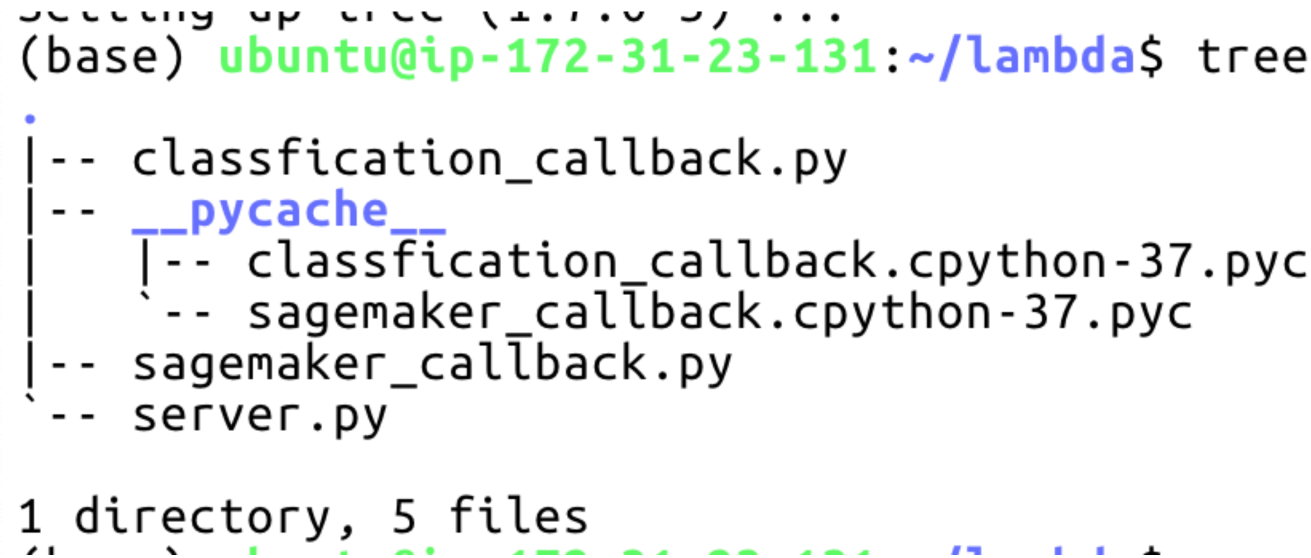
Practical Issue

- For some reasons, we cannot create IAM role to make components interact with each other.

- So we replace the Lambda and the API Gateway with another EC2 instance, which runs a Flask server.

- Meanwhile, SSM agent is replaced by another Flask server hosted on inference server.

Fake Lambda =_=



```
Setting up tree (1.7.0-5) ...
(base) ubuntu@ip-172-31-23-131:~/lambda$ tree
.
|-- classfication_callback.py
|-- __pycache__
|   |-- classfication_callback.cpython-37.pyc
|   `-- sagemaker_callback.cpython-37.pyc
|-- sagemaker_callback.py
`-- server.py

1 directory, 5 files
(base) ubuntu@ip-172-31-23-131:~/lambda$
```

Future work

- Deploy custom classification task on SageMaker
- Accelerate/Scale up Giant Language Model inference.
- Taking real-world issues into account, e.g. high concurrency.