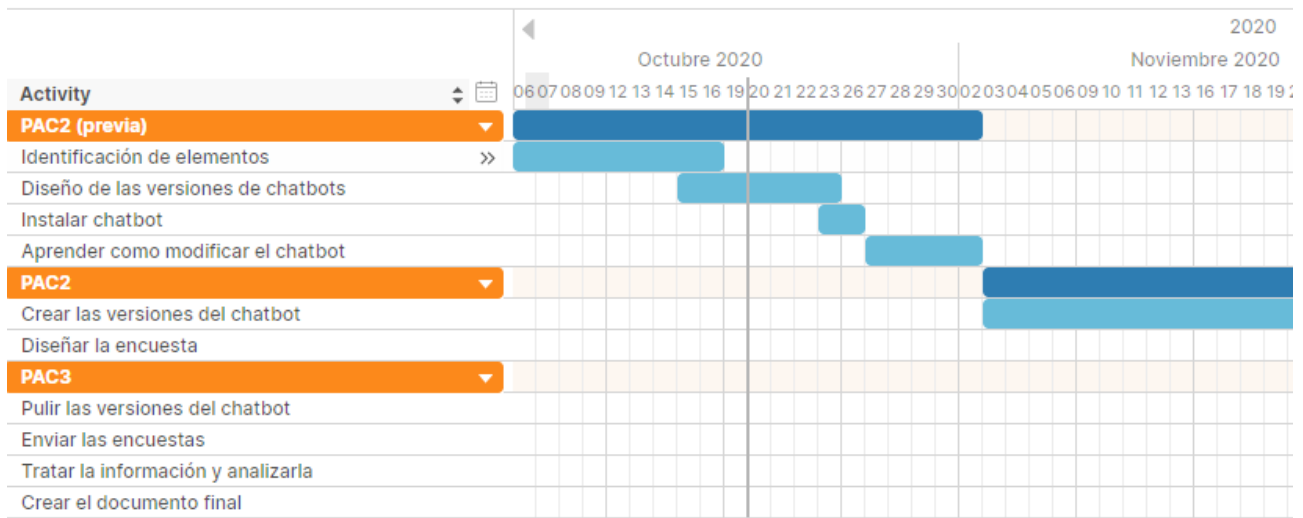


PAC2

Elementos de la interfaz que
modifican el resultado de
la teoria del Valle Inquietante
en los chatbots

Estado actual del proyecto



Según la planificación, a estas alturas debería tener los elementos a incluir en el proyecto identificados, las versiones de los chatbots que voy a usar el chatbot instalado y cierto aprendizaje adquirido para que sea más sencillo modificar el chatbot para la siguiente entrega.

Como a continuación explico, he tenido que modificar ciertos aspectos de la investigación para adaptar los problemas que van surgiendo y hacer algo más claro y conciso. En algunos puntos me he quedado atrasada respecto a lo que se tenía planificado, y en respecto a otros he avanzado más de lo esperado.

Diseño del chatbot

En cuanto al diseño, los elementos a identificar como posibles candidatos que afecten a la teoría de uncanny valley están bien definidos y con la forma clara de hacerlo.

Versiones de chatbots

Cuando tengamos que realizar las encuestas, los usuarios deberán probar una serie de versiones diferentes de chatbots, por lo que no deberían ser más de 3 versiones distintas o los encuestados se aburrirán y cansarán.

El problema aquí es decidir qué elementos poner en cada chatbot, porque si los mezclamos, ¿cómo podremos saber si lo que incomoda más es el avatar, la velocidad del texto, los sonidos, etc?

Quizás debería acotarse mucho más la investigación solo a un elemento visual, sino puede verse difuminada y no podremos sacar conclusiones claras de esta.

Una opción es que no todos los encuestados se enfrenten a los mismos chatbots, siendo que puedan haber 2 versiones diferentes de la misma encuesta y tener un total de 6 versiones de chatbot diferentes, pero habría que valorar qué es lo óptimo.

Si solo se decide un elemento, creo que el más significativo sería el del avatar.

Avatar del chatbot

La idea era crear un avatar que modificase su expresión según el ánimo con la que el chatbot contestaba. Como no tenemos una forma de identificar el estado de ánimo con la que el chatbot contesta, pero si que contesta en base al estado de ánimo con el que se envía el mensaje por parte del usuario, identificaremos el estado en el que el usuario lo envía mediante emojis/emoticonos, y cambiaremos su rostro en base a esto.

Además, se quiere comprobar si avatares más reales o más ficticios crean una incomodidad añadida en el usuario.

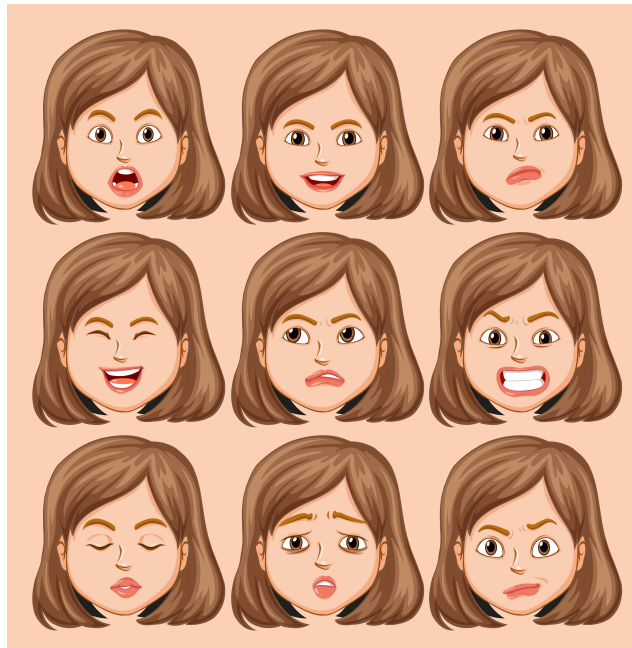
De esta forma tenemos los siguientes avatares posibles (de forma provisional), todos ellos con licencia gratis de uso:



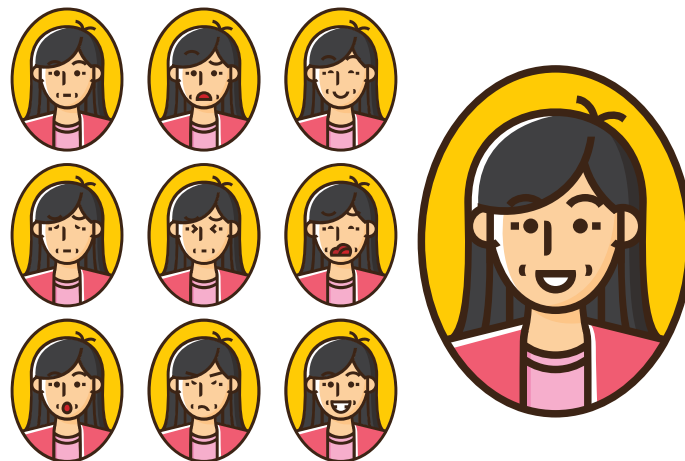
https://commons.wikimedia.org/wiki/File:Universal_emotions7.JPG



<https://www.vecteezy.com/vector-art/418094-man-with-different-facial-expressions>



<https://www.vecteezy.com/vector-art/359624-set-of-girl-head-with-different-facial-expression>



<https://www.vecteezy.com/vector-art/126996-mommy-emotions-expression-vector>



<https://www.vecteezy.com/vector-art/137674-free-border-collie-vector>

Se espera que cuanto menos real es el avatar, más se mitigará el efecto de uncanny valley, siendo el avatar del perro el que menos incomodidad cause.

Otro elemento interesante a investigar, es si un avatar femenino o masculino causa más incomodidad que el otro, ya que en uno de los papers comentados en el estado del arte indicaba que los usuarios se sentían más cómodos con avatares femeninos. Por ahora no ha sido posible encontrar un set de emociones masculinas gratis, por lo que no se si este elemento llegará a ser aplicado en la investigación.

Otros posibles elementos a incluir

Hay otras opciones que podemos incluir, pero como aún no se sabe si serán incluidos no vamos a profundizar mucho por ahora:

- Diferentes velocidades de la respuesta
- Sonidos satisfactorios al interactuar
- Animaciones de chat
- Colores de la interfaz

Diseño de la encuesta

Debido a que no tenemos aún los diseños claros de chatbots no podemos aún crear la encuesta provisional (además de ser algo planificado para la siguiente entrega), pero si que se puede empezar a definir qué queremos extraer y que cosas tener en cuenta.

Pediremos que cada usuario interactue durante 3 minutos aproximadamente con cada versión del chatbot, de esta forma si son 3 versiones de chatbot podemos vender la encuesta como que se tarda 10 minutos en hacerse, además de ser suficientes para crearse una opinión.

Lo primero sería preguntar información personal de los usuarios, para así poder identificar si elementos personales afectan sobre la incomodidad del chatbot.

Cosas a preguntar:

- Edad
- Género
- Condición mental (Neurodivergente o Neurotípico)

Otro factor interesante es conocer si el usuario ya ha interactuado previamente con otros chatbots y en qué cantidad, para de esta forma ver si esto crea una normalización o unas expectativas que para gente que no haya interactuado no influye.

Una escala del 1 al 10 sobre el nivel de incomodidad que ha sufrido con cada versión de chatbot.

Apartado técnico

Este es quizás uno de los apartados más importantes dentro del proyecto, ya que es la raíz para poder realizar el resto. Es uno de los puntos que más tiempo de proyecto ha absorbido y el que está en un estado más preocupante.

Se ha escogido cakechat como el backend del chatbot, ya que da respuestas correctas y profundas de una forma muy sencilla. Además, solo hace falta enviar los últimos tres mensajes y la emoción con la que se envía.

Para detectar la emoción del usuario al enviar el mensaje se ha descartado el modulo de python [Deepmoji](#) por su gran complejidad de implementación.

En su lugar, cada vez que se escriba un mensaje, vamos a analizar el texto en busca de un emoji o emoticono, que estarán clasificados en 5 apartados ("neutral", "joy", "anger", "sadness", "fear").

Para identificarlos vamos a usar una expresión regular:

<https://www.regextester.com/106421>

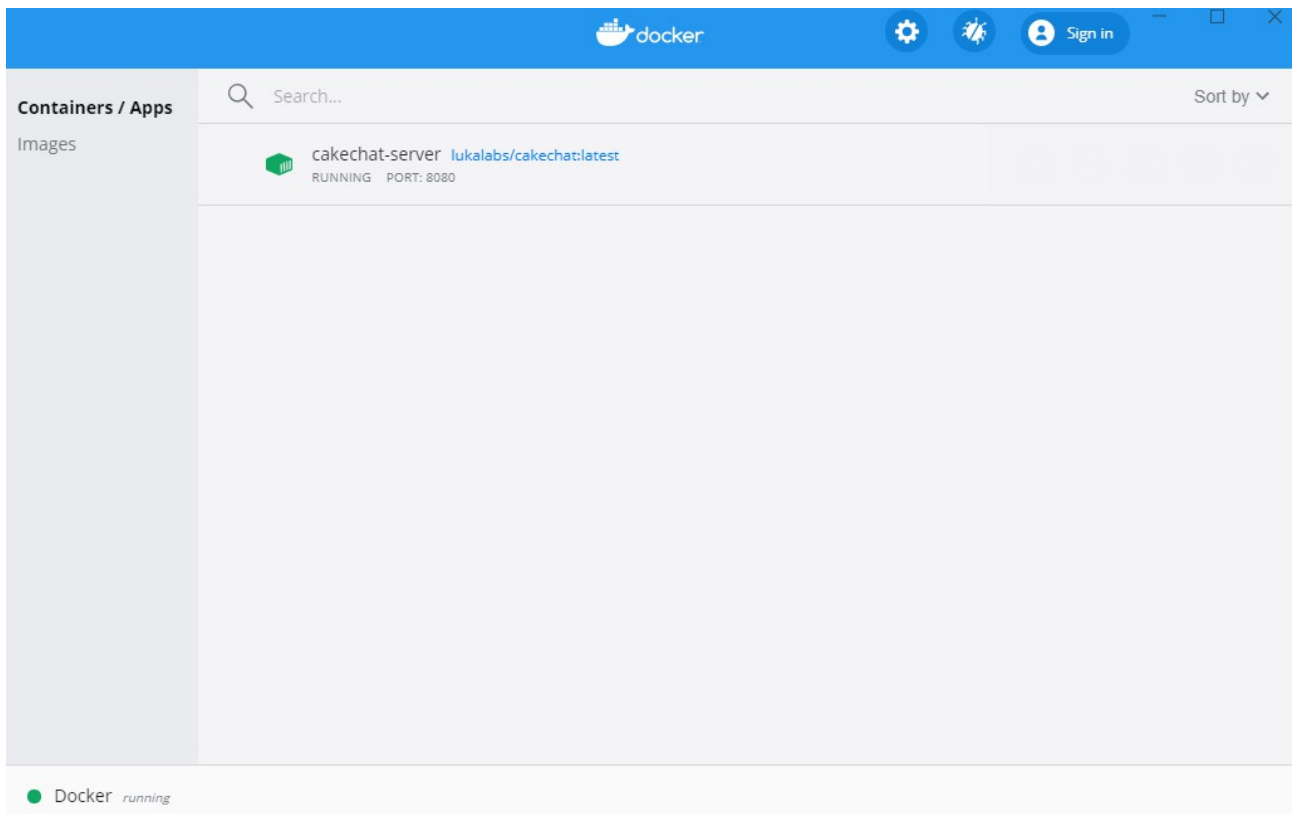
<https://stackoverflow.com/questions/28077049/regex-matching-emoticons>

Una vez identificado, para categorizar vamos a usar unos array con los emoticonos y emojis para cada emoción, ayudandonos de categorizaciones como:

<https://unicode.org/emoji/charts/full-emoji-list.html>

Lo primero ha sido instalar el chatbot en entorno de pruebas, primero instalando el backend de cakechat en un entorno definido en el git usando docker.

La instalación ha ido correctamente siguiendo los pasos indicados y el back funciona perfectamente.



Recibe inputs y devuelve el output adecuado, que va cambiando cada vez.

```
docker exec -it 4e0e4abb4bdef3254d9434c7b6958244ba79b3ed9690d27d8e2dd506cd4f9548 /bin/sh
ly calculated model
[02.11.2020 18:57:49.682][INFO][125][cakechat.dialog_model.inference_model.InferenceCakeChatModel][246] Restored model weights from /root/cakechat/results/nn_models/reverse_cakechat_v2.0_keras_tf_6648fddd592a/model
[02.11.2020 18:57:49.683][INFO][125][cakechat.dialog_model.inference_model.InferenceCakeChatModel][141] Loaded model: /root/cakechat/results/nn_models/reverse_cakechat_v2.0_keras_tf_6648fddd592a/model
[02.11.2020 18:57:49.683][INFO][125][cakechat.utils.w2v.model][66] Getting w2v model
[02.11.2020 18:57:50.309][INFO][125][cakechat.utils.s3.bucket][19] Getting file w2v_models/train_processed_dialogs_window10_voc50000_vec128_sgTrue.bin from AWS S3 and saving it as /root/cakechat/data/w2v_models/train_processed_dialogs_window10_voc50000_vec128_sgTrue.bin
[02.11.2020 18:57:56.470][INFO][125][cakechat.utils.s3.bucket][21] Got file w2v_models/train_processed_dialogs_window10_voc50000_vec128_sgTrue.bin from S3
[02.11.2020 18:57:56.473][INFO][125][cakechat.utils.w2v.model][53] Loading model from /root/cakechat/data/w2v_models/train_processed_dialogs_window10_voc50000_vec128_sgTrue.bin
[02.11.2020 18:57:56.755][INFO][125][cakechat.utils.w2v.model][55] Model "train_processed_dialogs_window10_voc50000_vec128_sgTrue.bin" has been loaded.
[02.11.2020 18:57:56.756][INFO][125][cakechat.utils.w2v.model][83] Successfully got w2v model

# python tools/test_api.py -f 127.0.0.1 -p 8080 \
  -c "Hi, Eddie, what's up?" \
  -c "Not much, what about you?" \
  -c "Fine, thanks. Are you going to the movies tomorrow?"> > >
Using TensorFlow backend.
{'response': 'Yes. If i go, you should.'}
#
```

Ya que aún estamos probando el backend, he cogido un modelo de chatbot muy simple de frontend para enviar la request por ajax:

<https://hielotemplated.neocities.org/>

El problema es que al enviar el post tal y como se indica desde el repositorio, da error porque el header externo al container no está en la whitelist del container para aceptar requests.

```
✖ Access to XMLHttpRequest at 'http://127.0.0.1:8080/cakechat_api/v1/actions/get_response' from origin 'https://hielote(index):1
mplatad.neocities.org' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested
resource.
error (index):25
✖ ▶ POST http://127.0.0.1:8080/cakechat_api/v1/actions/get_response net::ERR_FAILED jquery.min.js:2
>
```

Debido a la inexperiencia con docker ha sido imposible solucionar esto, pero en un servidor del cual se tenga control es fácil de resolver, por eso se ha empezado la implementación del servidor de prod con antelación.

Después de buscar muchos servidores en los que te den control absoluto del servidor mediante ssh para poder instalar lo que quiera el usuario, se ha acabado optando por usar el servicio gratuito para estudiantes de AWS, que aunque limitado puede servir.

<https://aws.amazon.com/es/education/awseducate/>

Se dispone de un servidor t2.micro, que mediante el uso va consumiendo créditos, pero de aproximadamente 0,01 créditos por hora, y se disponen de 100 créditos durante un año. Este servidor se apaga cuando no está en uso para que no consuma créditos de más.

Una vez el servidor ha sido montando y la conexión por ssh es correcta, queda instalar el chatbot. El problema viene de que en el git está preparado para usarse desde docker, por lo que para ponerlo en un servidor hay que buscarse la vida.

Para la instalación, lo lógico sería seguir los pasos que sigue docker en la instalación, por lo que se ha procedido a ejecutar los comandos del fichero de docker versión cpu en orden:

<https://github.com/lukalabs/cakechat/blob/master/dockerfiles/Dockerfile.cpu>

Pese a que se han tenido que instalar cosas adicionales, porque daba error de versión o falta de módulos, la instalación se ha hecho correcta, pero al ejecutar el mismo comando que en docker da error. El repositorio ha sido ubicado como en la ubicación de docker: root/cakechat. El problema es que este directorio en el servidor ubuntu 18 es una carpeta restringida, por lo que no debería ser lo óptimo.

El error viene de que hay que inicializar el sistema para que prepare la base de datos de respuesta con el comando:

```
python tools/fetch.py
```

Al ejecutarlo, da el siguiente error, cosa que en docker no:

AWS:


```

Last login: Sun Nov 1 15:19:04 2020 from 95.22.6.63
ubuntu@ip-172-31-31-168:~$ sudo su
root@ip-172-31-31-168:/home/ubuntu# cd /root/cakechat/
root@ip-172-31-31-168:~/cakechat# python tools/fetch.py
Using TensorFlow backend.
2020-11-02 19:17:38.899272: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this Tensor
[02.11.2020 19:17:40.509] [INFO] [20708] [cakechat.utils.files_utils][169] Loading /root/cakechat/results/tensorboard/steps
[02.11.2020 19:17:40.613] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][276] Initializing NN model
[02.11.2020 19:17:40.613] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][282] Building tokens_emb
[02.11.2020 19:17:40.626] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][282] Building tokens_emb
[02.11.2020 19:17:40.636] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][298] Building condition_emb
[02.11.2020 19:17:40.653] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][316] Building utterance_enc
[02.11.2020 19:17:44.694] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][336] Building context_encod
[02.11.2020 19:17:46.369] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][355] Building decoder...
[02.11.2020 19:17:49.124] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][442] Building seq2seq...
[02.11.2020 19:17:50.117] [INFO] [20708] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][278] NN model is initialize

2020-11-02 19:17:50.283692: W tensorflow/core/framework/allocator.cc:122] Allocation of 7077888 exceeds 10% of system memory.
2020-11-02 19:17:50.285996: W tensorflow/core/framework/allocator.cc:122] Allocation of 7077888 exceeds 10% of system memory.
2020-11-02 19:17:50.287005: W tensorflow/core/framework/allocator.cc:122] Allocation of 7077888 exceeds 10% of system memory.
2020-11-02 19:17:50.287963: W tensorflow/core/framework/allocator.cc:122] Allocation of 7077888 exceeds 10% of system memory.
2020-11-02 19:17:50.288894: W tensorflow/core/framework/allocator.cc:122] Allocation of 7077888 exceeds 10% of system memory.
Killed
root@ip-172-31-31-168:~/cakechat# █

```

Docker:

```

# python tools/fetch.py
Using TensorFlow backend.
2020-11-02 18:55:51.093151: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
[02.11.2020 18:55:51.964] [INFO] [125] [cakechat.utils.files_utils][169] Loading /root/cakechat/results/tensorboard/steps
[02.11.2020 18:55:52.008] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][276] Initializing NN model
[02.11.2020 18:55:52.008] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][282] Building tokens_embedding_model...
[02.11.2020 18:55:52.020] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][282] Building tokens_embedding_model...
[02.11.2020 18:55:52.032] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][298] Building condition_embedding_model...
[02.11.2020 18:55:52.051] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][316] Building utterance_encoder...
[02.11.2020 18:55:57.221] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][336] Building context_encoder...
[02.11.2020 18:56:08.053] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][355] Building decoder...
[02.11.2020 18:56:17.465] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][442] Building seq2seq...
[02.11.2020 18:56:18.652] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][278] NN model is initialized

[02.11.2020 18:57:08.150] [INFO] [125] [cakechat.dialog_model.inference_model.InferenceCakeChatModel][272]

Model weights summary:
layer name                                output shape:                                size:
x_token_embedding/embeddings:0            (50000, 128)                                24.41Mb
utterance_scope/bidir_utterance_encoder/forward_encoder/kernel:0          (128, 2304)                                  1.12Mb
utterance_scope/bidir_utterance_encoder/forward_encoder/recurrent_kernel:0 (768, 2304)                                  6.75Mb
utterance_scope/bidir_utterance_encoder/forward_encoder/bias:0             (2, 2304)                                    0.02Mb
utterance_scope/bidir_utterance_encoder/backward_encoder/kernel:0          (128, 2304)                                  1.12Mb
utterance_scope/bidir_utterance_encoder/backward_encoder/recurrent_kernel:0 (768, 2304)                                  6.75Mb
utterance_scope/bidir_utterance_encoder/backward_encoder/bias:0            (2, 2304)                                    0.02Mb
utterance_scope/utterance_encoder_final/kernel:0                            (1536, 2304)                                 13.50Mb
utterance_scope/utterance_encoder_final/recurrent_kernel:0                 (768, 2304)                                  6.75Mb
utterance_scope/utterance_encoder_final/bias:0                              (2, 2304)                                    0.02Mb
encoder_scope/context_encoder/kernel:0                                     (768, 2304)                                  6.75Mb
encoder_scope/context_encoder/recurrent_kernel:0                           (768, 2304)                                  6.75Mb
encoder_scope/context_encoder/bias:0                                        (2, 2304)                                    0.02Mb
condition_embedding_scope/condition_embedding/embeddings:0                 (5, 128)                                      0.08Mb
y_token_embedding/embeddings:0                                              (50000, 128)                                24.41Mb
decoder_scope/decoder_0/kernel:0                                           (1024, 2304)                                 9.08Mb
decoder_scope/decoder_0/recurrent_kernel:0                                  (768, 2304)                                  6.75Mb
decoder_scope/decoder_0/bias:0                                              (2, 2304)                                    0.02Mb
decoder_scope/decoder_1/kernel:0                                            (768, 2304)                                  6.75Mb
decoder_scope/decoder_1/recurrent_kernel:0                                  (768, 2304)                                  6.75Mb
decoder_scope/decoder_1/bias:0                                              (2, 2304)                                    0.02Mb
decoder_scope/dense_1/kernel:0                                              (768, 50000)                                 146.48Mb
decoder_scope/dense_1/bias:0                                                (50000,)                                     0.19Mb

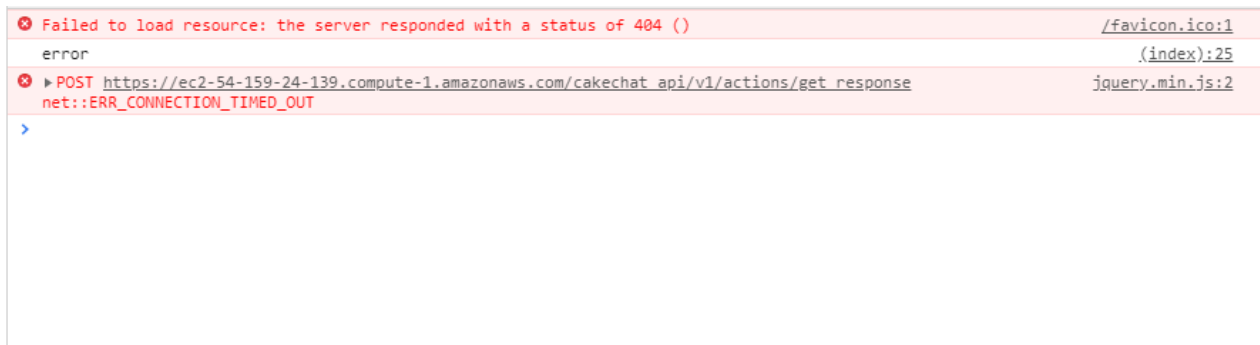
Total network size: 274.4 Mb

```

Quizás este error puede ser debido a la memoria que se usa en el servidor de AWS, quizás es debido a alguna limitación o a la mal instalación de este.

Además, tampoco queda claro como va a entrar la petición por el servidor, ya que la url dada no responde y no queda claro si habría que instalar un servicio de servir información como puede ser apache o nginx.

<https://ec2-3-80-67-245.compute-1.amazonaws.com/>



En cualquier caso habría que revisar los puertos con los que se puede acceder a este servicio y como lo enlaza, ya que la url para enviar el post es:
/cakechat_api/v1/actions/get_response

La parte positiva es que una vez que funcione el servidor y acepte peticiones, tendremos hecho gran parte del trabajo, ya que el resto se trata de modificar el front con jQuery y html/css, cosa relativamente fácil.