



Hazme feliz.

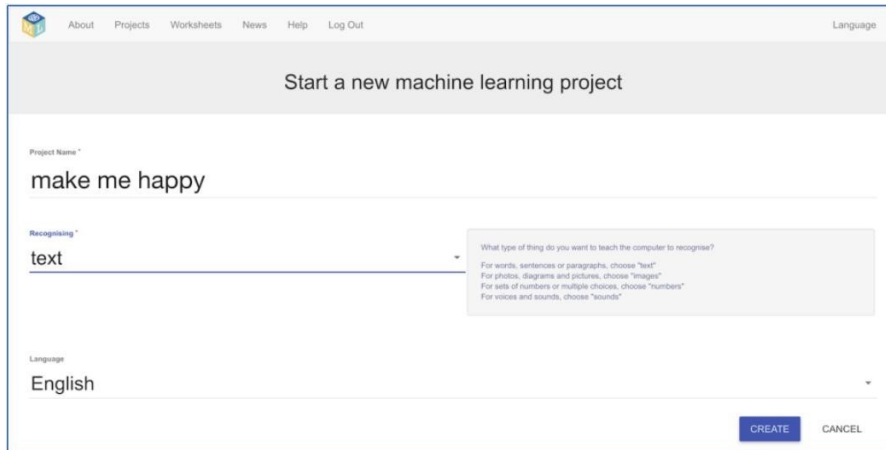
En este proyecto, harás un programa de Python que puede reaccionar a los mensajes que le envíe.

Si lo felicitas, se verá feliz. Si lo insultas, se verá triste. Utilizarás el aprendizaje automático para entrenar al ordenador para reconocer los mensajes de cada tipo, dándole ejemplos de cada uno de ellos.

1. Necesitas dos fotos: una foto de una cara feliz, y una foto de una cara triste. Dibuja o, si lo prefieres, encuentra imágenes en Internet en vez de guardar las dos imágenes en algún lugar de tu ordenador y recordar dónde las colocas. Los necesitarás más tarde.
2. Ve a <https://machinelearningforkids.co.uk/> en un navegador web
3. Haz clic en **Empezar**
4. Haz clic en **Iniciar sesión** y escribe tu nombre de usuario y contraseña
Si no tienes un nombre de usuario, pídele a tu profesor o líder de grupo que te cree uno.
Si no recuerdas tu nombre de usuario o contraseña, pídele a tu profesor o al líder del grupo que te lo restablezca.
5. Pulsa en **Proyectos** en la barra superior.
6. Haz clic en el botón **Añadir un nuevo proyecto**.

7. Nombra a tu proyecto como **Me hace feliz** y ponlo para aprender a reconocer **texto**.

Haz clic en el botón **Crear**



Start a new machine learning project

Project Name *
make me happy

Recognising *
text

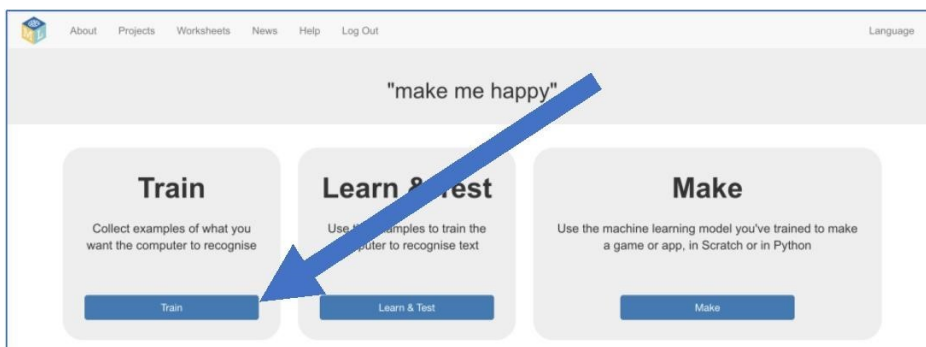
What type of thing do you want to teach the computer to recognise?
For words, sentences or paragraphs, choose "text"
For photos, diagrams and pictures, choose "images"
For sets of numbers or multiple choices, choose "numbers"
For voices and sounds, choose "sounds"

Language
English

CREATE CANCEL

8. Ahora deberías ver **Me hace feliz** en la lista de tus proyectos.
Haz clic en él.

9. Necesitas ejemplos para entrenar el ordenador. Haz clic en el botón **Entrenar**.



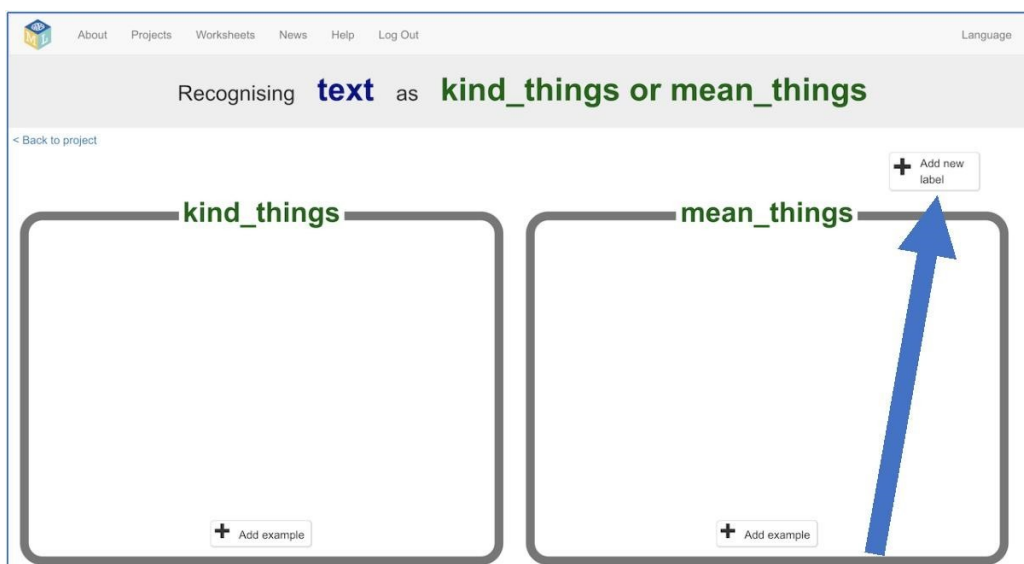
"make me happy"

Train
Collect examples of what you want the computer to recognise
Train

Learn & Test
Use examples to train the computer to recognise text
Learn & Test

Make
Use the machine learning model you've trained to make a game or app, in Scratch or in Python
Make

10. Haz clic en **Añadir etiqueta** y llámalo **cosas amables**. Haz eso de nuevo, y crea un segundo grupo llamado **cosas mezquinas**.



Recognising text as kind_things or mean_things

< Back to project

kind_things

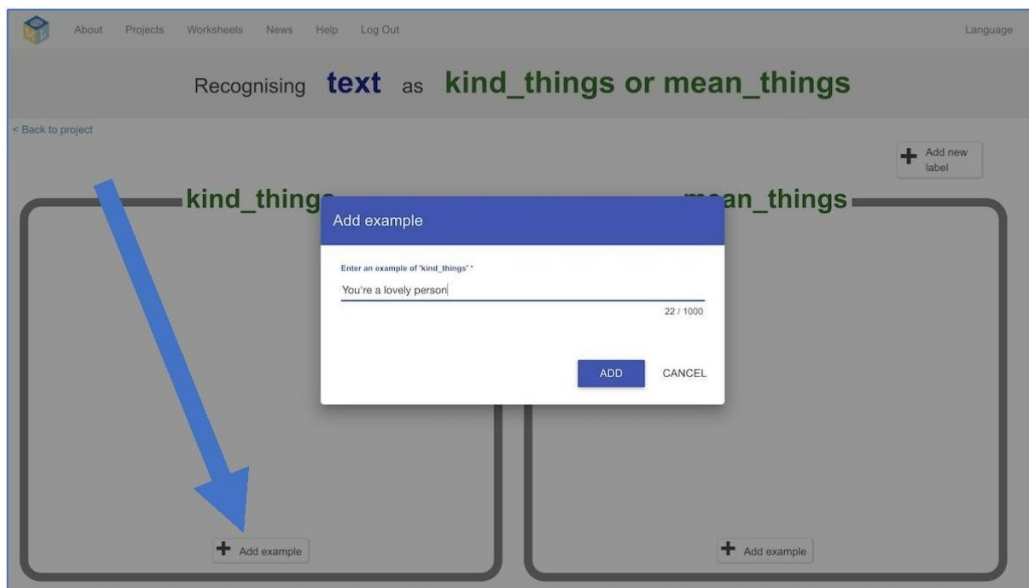
mean_things

+ Add new label

+ Add example

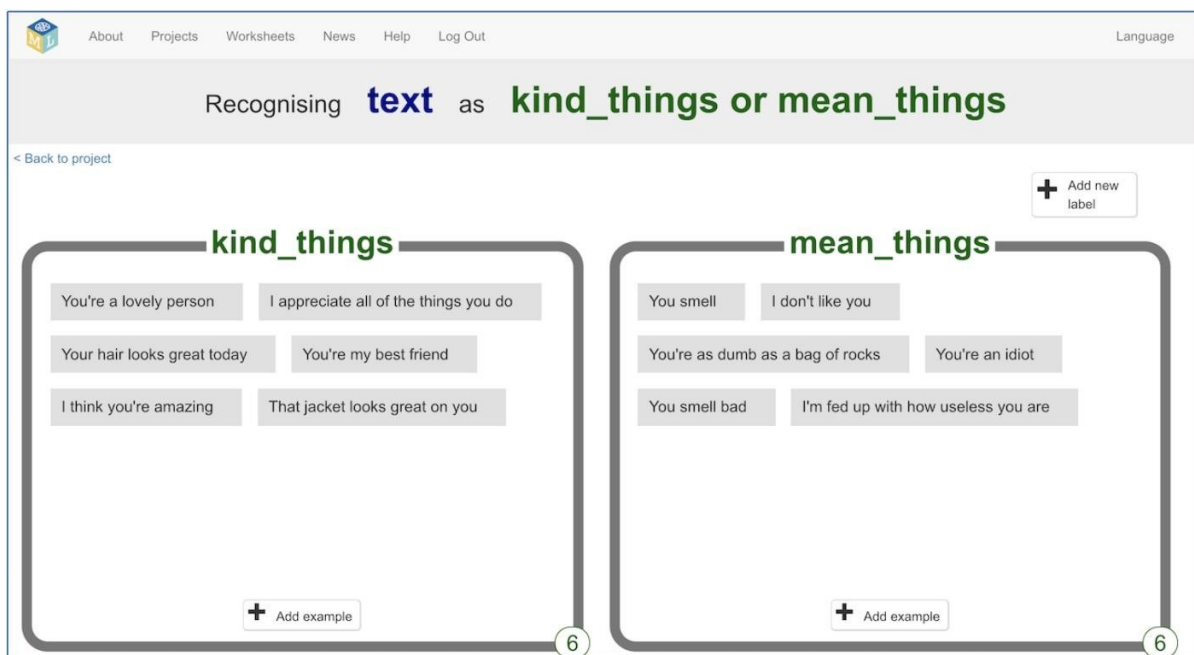
+ Add example

- 11.** Pulsa el botón **Añade un ejemplo** en el grupo de "cosas amables" y añade frases o palabras que sean cumplidos bonitos.



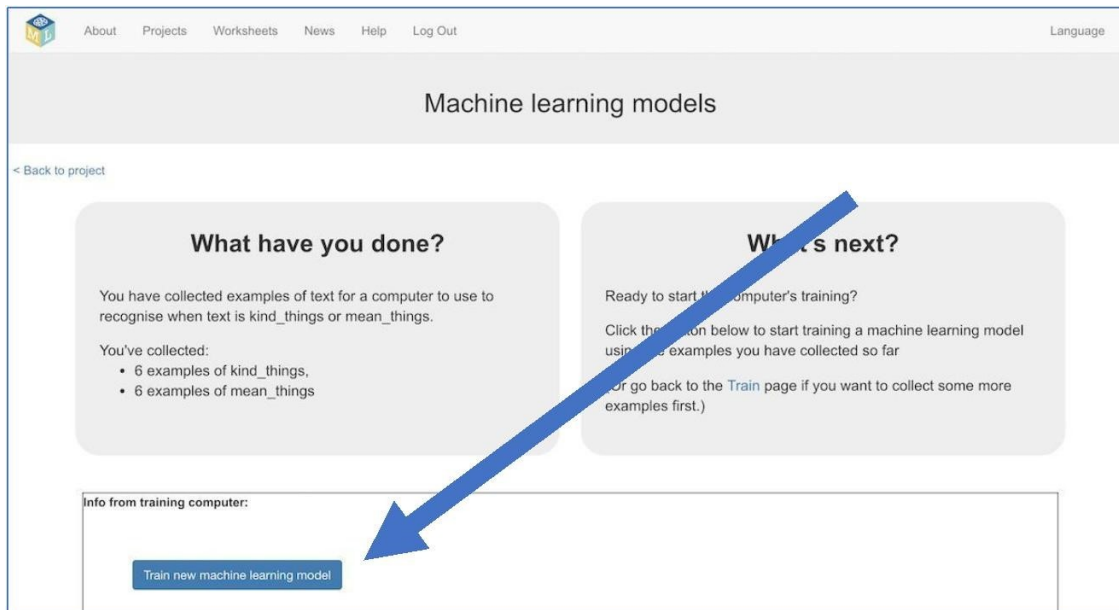
- 12.** Pulsa el botón **Añade un ejemplo** en el grupo de "cosas mezquinas" que sean los insultos más crueles que conozcas.

- 13.** Repite los pasos 9 y 10 hasta que tenga al menos seis ejemplos de cada uno



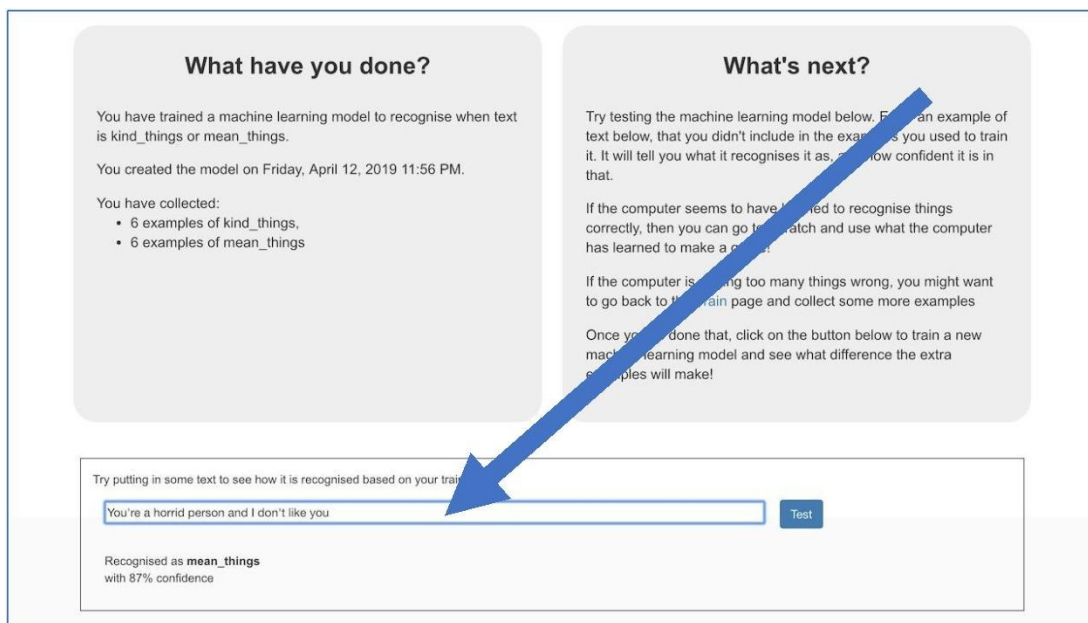
- 14.** Haz clic en **Volver al proyecto** y luego haz clic en el botón **Aprender & Probar**.

- 15.** Haz clic en el botón **entrenar nuevo modelo**. Mientras hayas recopilado suficientes ejemplos, el ordenador debería empezar a aprender a reconocer los mensajes de los ejemplos que le ha dado.



- 16.** Espera a que se complete el entrenamiento. Esto puede tardar unos minutos.

- 17.** Una vez completado el entrenamiento, se mostrará un recuadro de prueba.

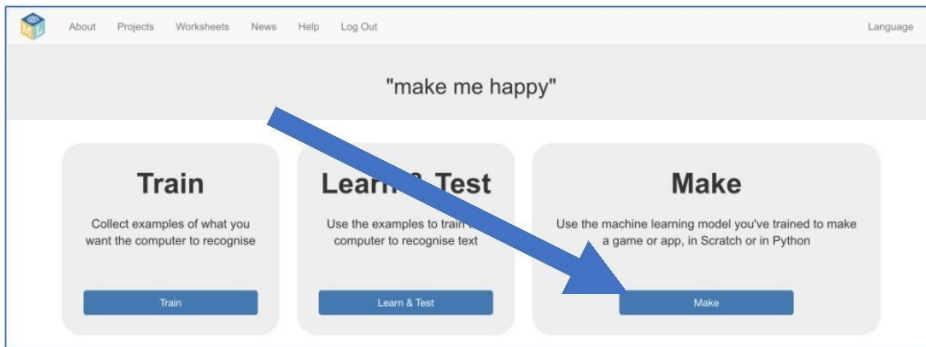


Intenta probar el modelo de aprendizaje automático para ver lo que ha aprendido el sistema. Teclea algo bueno, y pulsa Intro. Debería ser reconocida como una palabra amable. Escribe algo malo y pulsa Intro. Debería reconocerse como una palabra mezquina.

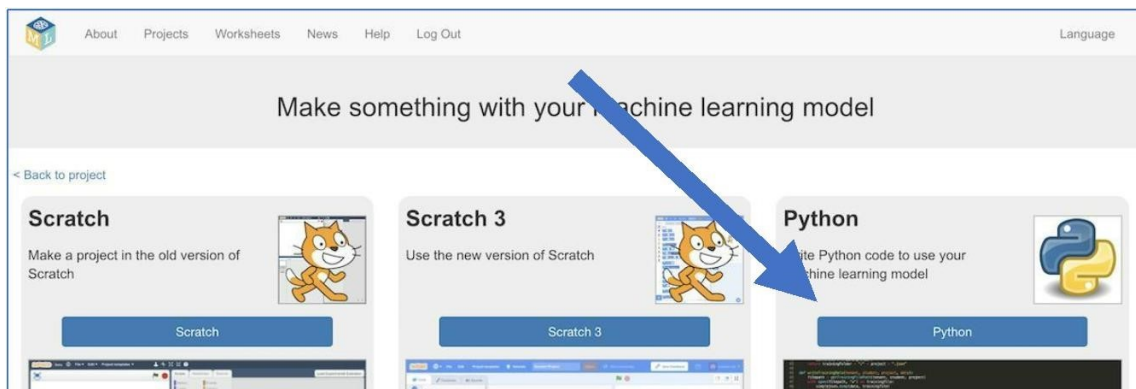
Prueba con ejemplos de que no has mostrado el sistema antes. Si no estás contento con la forma en que el ordenador reconoce los mensajes, regresa y agrega algunos ejemplos más.

18. Pulsa el enlace **Volver al Proyecto**

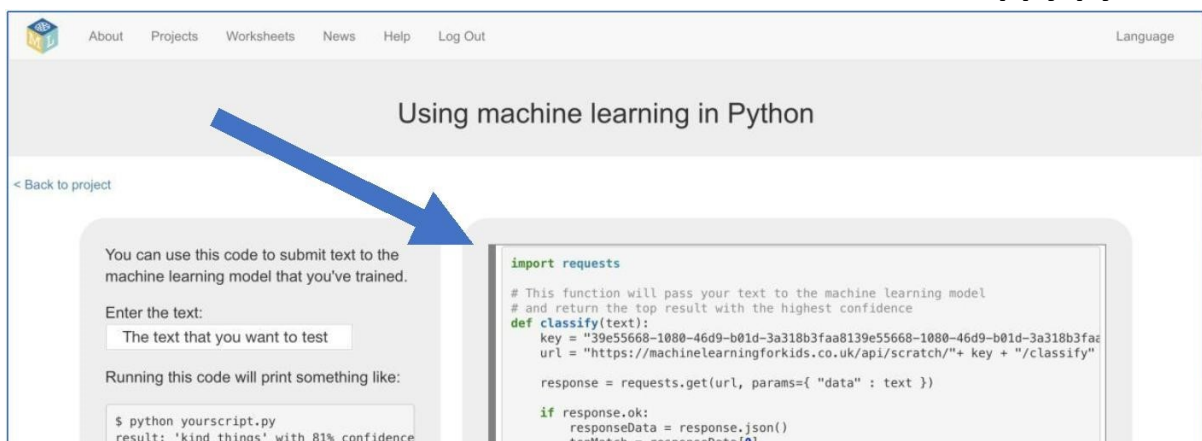
19. A continuación, usaremos Python. Haz clic en el botón **Crear**



20. Haz clic en el botón **Python**.



21. Copia el código de esqueleto que se visualiza en el editor de texto. Guarda el archivo como un archivo denominado **makemehappy.py**



22. Suprime la mitad inferior del código

Todo desde “#CHANGE THIS” hasta el final del archivo, resaltado a continuación. Suprímelo, ya que no lo necesitarás.

```
makemehappy.py •
1  import requests
2
3  # This function will pass your text to the machine learning model
4  # and return the top result with the highest confidence
5  def classify(text):
6      key = "83d7dbcb-aa09-4df3-9bf9-237217b6127cbfbf1eba-d721-487c-9e9f-f86d6a1b8962"
7      url = "https://machinelearningforkids.co.uk/api/scratch/"+ key + "/classify"
8
9      response = requests.get(url, params={ "data" : text })
10
11      if response.ok:
12          responseData = response.json()
13          topMatch = responseData[0]
14          return topMatch
15      else:
16          response.raise_for_status()
17
18  # CHANGE THIS to something you want your machine learning model to classify
19  demo = classify("The text that you want to test")
20
21  label = demo["class_name"]
22  confidence = demo["confidence"]
23
24
25
26  # CHANGE THIS to do something different with the result
27  print ("result: '%s' with %d%% confidence" % (label, confidence))
28
```

23. Estarás usando una biblioteca de Python llamada Pillow para mostrar tus fotos.

Si no lo tienes instalado, tienes que hacer esto ahora. Pide a tu profesor o líder de grupo que te ayude. Se pueden encontrar instrucciones en

<https://pillow.readthedocs.io/en/stable/installation.html>

24. Añade “from PIL import Image” en la parte superior del Código.

```
makemehappy.py x
1  from PIL import Image
2  import requests
3
4  # This function will pass your text to the machine learning model
5  # and return the top result with the highest confidence
6  def classify(text):
7      key = "45ea3e2a-8fa1-4045-8307-da370882296d8e63ee85-e2a4-4989-94a7-a2b06228
8      url = "https://machinelearningforkids.co.uk/api/scratch/"+ key + "/classify
9
10     response = requests.get(url, params={ "data" : text })
```

25. Añade esto al final de tu código.

```
input = raw_input (" ¿Qué es lo que quieres decirme? > ") reconocida = classify (input) label = reconocida
["class_name"]
```

```
if label == "kind_things": print ("¡ Eres tan bonita!") img = Image.open (" happy.png") img.show () de otro modo:
```

```
print ("¡ Eres tan malo!") img = Image.open (" sad.png") img.show ()
```

Te pedirá que escribas un mensaje.

El mensaje que escribas se almacenará en una variable llamada "input". Esta variable se dará al modelo de aprendizaje automático que has entrenado para que puedas tratar de reconocerlo. Si el modelo de aprendizaje automático lo reconoce como un cumplido, su programa mostrará un mensaje de "Eres tan agradable" y mostrará tu foto de una cara feliz.

Tiene que actualizar **-¿Por que no?** y **entristpng** para que coincida con los nombres de tus fotos de caras.

26. Ejecuta el programa con el mandato: **python makemehappy.py**

¿Qué has hecho?

Has realizado un programa de Python que solicita la entrada de los usuarios. A continuación, utilizarás el modelo de aprendizaje automático que ha capacitado para reconocer si lo que el usuario ha puesto ha sido un cumplido o un insulto. Esto se describe a menudo como un "análisis de opinión".

Cuanto mayor sea el número de ejemplos que se le dan, mejor deberá obtener un reconocimiento correcto de los mensajes

Ideas y xtensiones

Ahora que has terminado, ¿por qué no pruebas alguna de estas ideas?
¿O creas tú una propia?

Grabar una respuesta

En lugar de cambiar la forma en que miran, haz la respuesta de personaje, en base a lo que reconoce en el mensaje!

Intentar un carácter diferente

En vez de la cara de una persona, ¿por qué no intentar algo diferente, como un animal?
¡Por ejemplo, podrías hacer un perro que mueve la cola si dices algo amable de él!

Análisis de opinión del mundo real

¿Se puede pensar en ejemplos en los que es útil poder entrenar a un ordenador para reconocer la emoción por escrito?