NLP

Part 2

Understand how a computer attempts to process human language

A decision tree is an easy to understand decision making tool that allows us to visualise how a computer is understanding us.

Take a look at this example. We have used an array of emotions because emotion and sentiment analysis and intent recognition play a large part in Natural Language Processing.

Sentiment analysis classifies text into three groups, Positive, Neutral and Negative. Let's use a decision tree to help teach our robot which emotions are positive and which are negative.

In this example we start the tree with the question, "Is it a positive emotion?" We chose this question because we were looking for the highest level dichotomy that we could find when it comes to emotions.

Can you think of a different question that could start this tree?
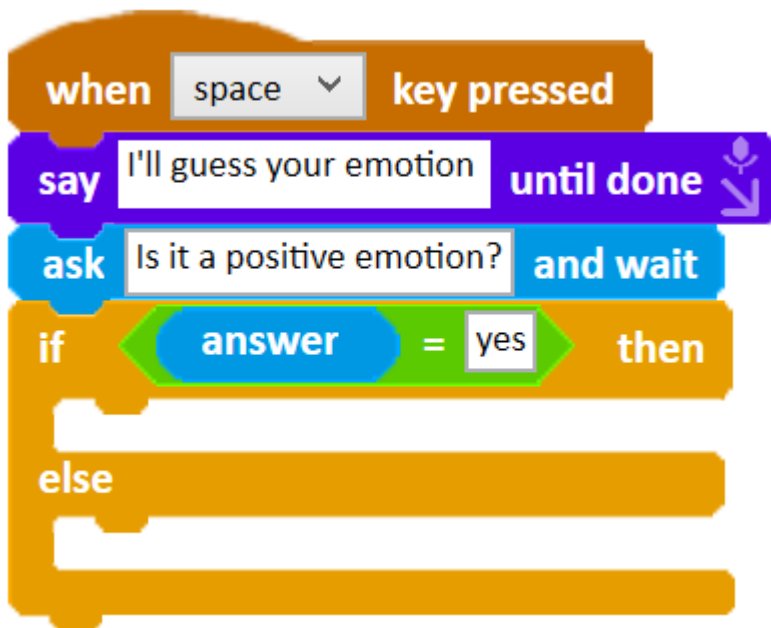
From here, the program then branches down the two routes of positive or negative emotions with further yes/no questions until it makes a guess at your emotion.

Try running the program and see whether it guesses your emotion correctly.

```
when space key pressed
say "I'll guess your emotion" until done
ask "Is it a positive emotion?" and wait
if < answer = yes > then
    ask "Does it make you giddy?" and wait
    if < answer = yes > then
        ask "Does it make your heart beat faster?" and wait
        if < answer = yes > then
            say "Love" until done
        else
            say "Joy" until done
    else
        ask "Are you thinking about the future?" and wait
        if < answer = yes > then
            say "Hope" until done
        else
            say "Pride" until done
else
    ask "Does it make you red in the face?" and wait
    if < answer = yes > then
        ask "Have you been treated unfairly?" and wait
        if < answer = yes > then
            say "Anger" until done
        else
            say "Shame" until done
    else
        ask "Does it make you want to flee?" and wait
        if < answer = yes > then
            say "Fear" until done
        else
            say "Sadness" until done
```

Each **if _ then / else** block separates answers containing the word "yes" and all other answers. You could add an **Or** operator block to include alternative positive affirmations such as "yeah"

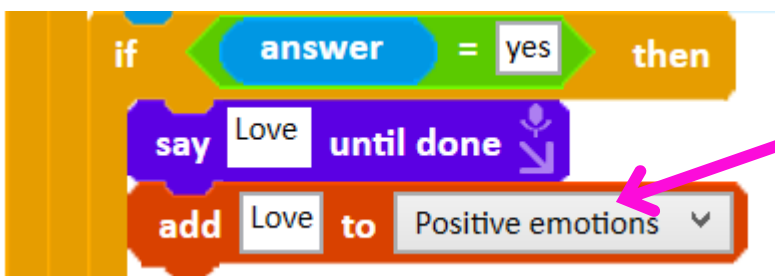Can you think of any other emotions you could add to this program?

What questions would you ask to find out how someone was feeling?

Here is a diagram of the decision tree that we just used.

Now let's use our decision tree to train our robot about emotions.

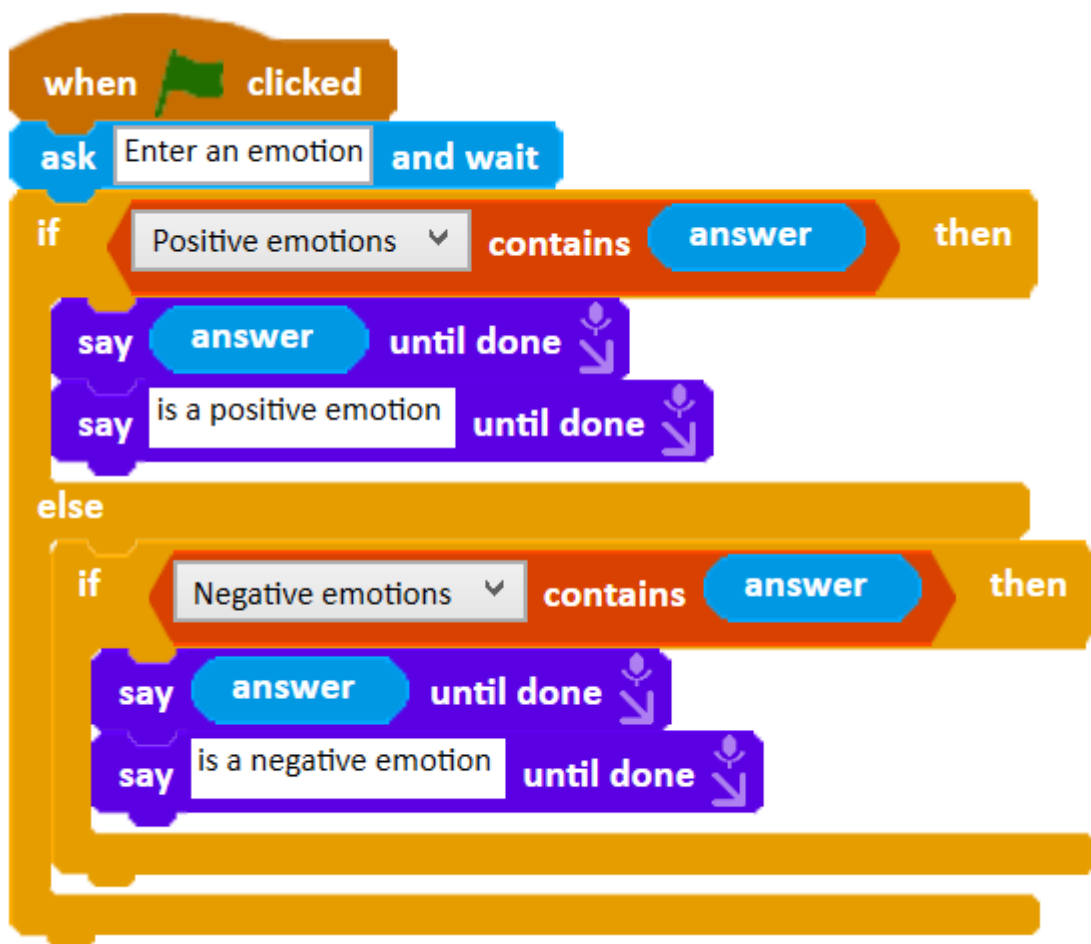Create two lists, one for positive emotions and one for negative.

Make sure you add to the right list

Put an **add _ to _** block below each emotion and run the program again.

When you reach an emotion on the decision tree, that emotion will now be added to either the positive or negative emotion list.

Positive emotions: Love

By doing this you are building up a dataset for the robot to refer to.

```
when 🚩 clicked
ask Enter an emotion and wait
if  Positive emotions ∨  contains  answer  then
    say  answer  until done
    say  is a positive emotion  until done
else
    if  Negative emotions ∨  contains  answer  then
        say  answer  until done
        say  is a negative emotion  until done
```

The robot can now recognise emotions as either positive or negative.

Can you see how this kind of emotion recognition is useful?

Businesses are using the emotion and sentiment analysis aspects of NLP to sort through massive amounts of data, gaining insights into customer's likes and dislikes.

Try making a list for neutral emotions and add branches to the decision tree that lead to neutral emotions.