Pokemon Classifier Exercise

One area of research in linguistics is *sound symbolism*, which studies the correlation between sounds and their meaning. *Pokémonastics* looks at sound symbolism specifically in Pokémon names. (There is quite a bit of linguistic research in this area)!

Past experiments with novel Pokémon have shown that speakers of several languages (including English, Japanese, French, Spanish, Portuguese, German, Italian, Korean, and Mandarin) make consistent judgements about Pokémon characteristics based on the sounds and/or other linguistic characteristics in their names. Let's test this!

Sound symbolism in Pokémon names

Download the Pokemon corpus ("pokemon.csv"). This is a file you can open in Excel or another spreadsheet program, adapted from a much larger set of data by Andrew Lamont, available at https://lingbuzz.net/lingbuzz/007137.

First, as a group, look through the Pokémon corpus. Try to find at least 3 characteristics that you might be able to guess if you were given a Pokémon's name. Think of how certain phonemes or linguistic characters make you *feel* (e.g. *What kind of Pokémon might be associated with long names? What kind of Pokémon might have more* [w]*s?*). Note: You do not actually have to have any knowledge of Pokémon to be able to complete this task.

1. ____
2. ____
3. ____

Step 1: Text

Let's walk step-by-step through creating a Pokémon classifier! First, we need our text. Choose one of the following subsets of Pokémon, based on type:

□ Flying

□ Dark

□ Fairy

Load the group of Pokémon you want to work with by first downloading the corresponding `type.pickle` file (where `type` is the type you have chosen). Then load it into Python using the following:

```
import pickle
type = pickle.load(open("type.pickle", "rb"))
```

To give the impression of randomized text, we will shuffle our list of Pokémon.

```
import random
random.shuffle(type)
```

Step 2: Features

Now we need features to train our classifier on. This will be the linguistic aspects of the Pokémons' names that you predict will help you decide "this is likely a (insert type) Pokémon".

Create a function that will take a word and give you the feature you are working with. An example is given below. (Note: Your function does not have to look at the initial segment).

```
def initialb(pokemon):
    if pokemon[0] == "b":
        initialb = True
    if pokemon[0] != "b":
        initialb = False
    return {'initialb': initialb}
```

Using your function, create a set of features to give the classifier.

```
featuresets = [(initialb(pokemon), pokemonType) for (pokemon, pokemonType)
                                               in type]
```

Step 3: Training data

Now we need to train our classifier! We will use the first 500 Pokémon to train the classifier.

```
    train_set = featuresets[:500]
```

We will run the built-in NLTK Naïve Bayes classifier function on this training data.

```
import nltk
classifier = nltk.NaiveBayesClassifier.train(train_set)
```

Pick a Pokémon and see if your classifier categorizes it correctly! How does it do when you try different Pokémon?

```
classifier.classify(initialb('Squirtle'))
```

Step 4: Testing data

Now let's test on "unseen" Pokémon, or the remaining 500.

```
test_set = featuresets[500:]
```

How accurate is the classifier? Above 50%? Below 50%?

```
print(nltk.classify.accuracy(classifier, test_set))
```

What are the most informative features in your classifier? What do they tell you about sound symbolism in the Pokémon names, if anything?

```
classifier.show_most_informative_features(5)
```

References

Kawahara, S., & Kumagai, G. (2019). Expressing Evolution in Pokémon Names: Experimental Explorations. *Journal of Japanese Linguistics*.

Kawahara, S., Noto, A., & Kumagai, G. (2018). Sound symbolic patterns in Pokémon names. *Phonetica*, *75*.