(Advanced) Text mining

Sunday, May 5, 2019 11:17 AM

Sometimes, data is embedded in text with little structure.
How do we extract data from unstructured text?

Consider the following problem:

```
budget = """My airfare was 300.00. My
hotel cost 200.00 for one night. My
food cost was 100.00."""
```

How do we sum up these numbers?
The numbers are embedded in an *unstructured string.*

Steps in solving this problem

Notice what is constant in the solution. The numbers are "numbers". Everything else is a regular word.
Use this constancy to transform the string to an intermediate representation.
Study that representation. What can I do with it?

First, let's break it into words, by splitting at spaces.

```
words = budget.split(' ')
```

Then words is:

```
['My',
 'airfare',
 'was',
 '300.00.',
 'My',
```

'hotel',
'cost',
'200.00',
'for',
'one',
'night.',
'My',
'food',
'cost',
'100.00.']

We note that there are numbers and non-numbers. We can do something exceedingly clever:

Try to convert everything to a number.

For everything that is a number, add to the results.

This results in the following code:

```
costs = []
for w in words:
    try:
        number = float(w)
        costs.append(number)
    except Exception as e:
        print(e)
```

The `try: … except … :` syntax tries to do something that may fail, e.g., `float(w)`. If w is a number, it works, and if not, it fails. If it fails, we print an error message.

This prints:

```
could not convert string to float: 'My'
could not convert string to float: 'airfare'
could not convert string to float: 'was'
could not convert string to float: '300.00.'
could not convert string to float: 'My'
could not convert string to float: 'hotel'
could not convert string to float: 'cost'
could not convert string to float: 'for'
could not convert string to float: 'one'
could not convert string to float: 'night.'
could not convert string to float: 'My'
could not convert string to float: 'food'
could not convert string to float: 'cost'
could not convert string to float: '100.00.'
```

Oops! Some numbers didn't convert!

200.00 converted fine.

numbers ending in a . did not convert!

Let's drop the . from the end of each word.

```
costs = []
for w in words:
    if w.endswith('.'):
        w = w[:-1]
    try:
        number = float(w)
        costs.append(number)
    except Exception as e:
        print(e)
```

The special syntax `w[:-1]` is w without its last character.

The special syntax `w.endswith('.')` is True if '.' is the

last character.

The combination

```
if w.endswith('.'):
    w = w[:-1]
```

removes the last character if it is a '.'

Running this produces:

```
could not convert string to float: 'My'
could not convert string to float: 'airfare'
could not convert string to float: 'was'
could not convert string to float: 'My'
could not convert string to float: 'hotel'
could not convert string to float: 'cost'
could not convert string to float: 'for'
could not convert string to float: 'one'
could not convert string to float: 'night'
could not convert string to float: 'My'
could not convert string to float: 'food'
could not convert string to float: 'cost'
```

and costs is:

```
[300.0, 200.0, 100.0]
```

Thus we can compute the sum as before from this.

This is an important example of python style. **To test whether you can do something, simply try to do it** and -- if it works -- use the value. If not, do something else.