Spam filter - krok 2

Vytvořte třídu `Corpus`, která bude obalovat adresář s emaily. Vybavte ji metodami, které umožní emaily snadno procházet.

Testy [/b211/courses/b4b33rph/cviceni/spam/unit_testing] ke kroku 2:

- samostatně test2_corpus.zip [/b211/_media/courses/b4b33rph/cviceni/spam/test2_corpus.zip] nebo
- společně se všemi předchozími testy test2_all.zip [/b211/_media/courses/b4b33rph/cviceni/spam/test2_all.zip].

Příprava

- Už byste měli vědět, jak pracovat s textovými soubory.
- Zjištění obsahu adresáře pomocí funkce os.listdir() [http://docs.python.org/py3k/library/os.html?highlight=listdir#os.listdir]
- Vytváření generátoru pomocí funkce `yield` (viz příklad v odstavci 9.10 [http://docs.python.org/py3k/tutorial/classes.html#generators] oficiálního tutoriálu Pythonu)

Korpus

Úkol:

- V modulu `corpus.py` vytvořte třídu `Corpus` obalující adresář se soubory (emaily)

K čemu nám to bude:

- Třída `Corpus` bude užitečná při klasifikaci nových emailů, učení filtru a bude sloužit jako základ pro třídu `TrainingCorpus`, která je jedním z dalších kroků.

Specifikace

Třída `Corpus` (v modulu `corpus.py`) bude obalovat adresář s emaily a umožní nám je snadno procházet. Třída bude mít následující vlastnosti:

- Při inicializaci jí bude předána cesta k adresáři s emaily.
- Třída bude mít metodu `emails()`, která bude generátorem. Tato metoda si bude vědoma toho, že v adresáři s emaily mohou být i soubory s metainformacemi. Název těchto souborů bude vždy začínat znakem `!` (např. `!truth.txt`), proto **všechny soubory začínající vykřičníkem v této metodě ignorujte!!!** Metoda nám umožní používat `Corpus` např. následujícím způsobem:

```
# Create corpus from a directory
corpus = Corpus('/path/to/directory/with/emails')
count = 0
# Go through all emails and print the filename and the message body
for fname, body in corpus.emails():
    print(fname)
    print(body)
    print('------------------------')
    count += 1
print('Finished: ', count, 'files processed.')
```

K výše uvedenému příkladu: Těla některých emailů obsahují unicode znaky - proto používáme **kódování utf-8**, abychom je v řetězci dovedli reprezentovat. **Při výpisu pomocí `print(body)` ale můžete občas dostat výjimku!** Záleží na tom, na jakém systému a v jakém shellu výše uvedený skript spustíte. Konzole, na kterou výpis probíhá, implicitně nějaké kódování používá a často je jiné než utf-8. Nastane pak situace, kdy se snažíme konzoli vnutit znak, který nezná.

Jedno z možných řešení je místo `print(body)` použít k výpisu `print(body.encode())`. Touto metodou se řetězec znaků převede na sekvenci bytů (datový typ `bytes`), která by se měla dát vypsat ať už konzole používá jakékoli kódování. Místo onoho problémového unicode znaku pak ve výpisu uvidíte sekvenci 2 až 4 jiných znaků. Nijak významně to ale výpis nepokazí.

Bug reports and suggestions Helpdesk CTU