

Warning

This page is located in archive. Go to the latest version of this [course pages](#). Go the latest version of [this page](#).

Spam filter - krok 3

Vytvořte sadu tříd a funkcí potřebných k posouzení kvality filtru.

Testy [\[/b211/courses/b4b33rph/cviceni/spam/unit_testing\]](#) ke kroku 3:

- samostatně [test3_quality.zip](#) [\[/b211/_media/courses/b4b33rph/cviceni/spam/test3_quality.zip\]](#) nebo
- společně se všemi předchozími testy [test3_all.zip](#) [\[/b211/_media/courses/b4b33rph/cviceni/spam/test3_all.zip\]](#).

Příprava

Prostudujte si:

- V článku [Binary Classification](http://en.wikipedia.org/wiki/Special:Search/Binary_Classification) [\[http://en.wikipedia.org/wiki/Special:Search/Binary_Classification\]](http://en.wikipedia.org/wiki/Special:Search/Binary_Classification) najděte a pochopte význam zkratk TP, FP, TN, FN.

Na kus papíru napište:

- co podle vás tyto zkratky znamenají pro problém filtrování spamu
- co budeme potřebovat, abychom tyto statistiky mohli napočítat

Matice záměn

Úkol:

- V modulu `confmat.py` vytvořte třídu `BinaryConfusionMatrix`.
- Třída bude obalovat čtveřici statistik TP, TN, FP, FN nutných pro hodnocení filtru.
- Při vytváření třídy bude nutné specifikovat parametry `pos_tag` a `neg_tag`, tedy jaké hodnoty se mají považovat za pozitivní a jaké za negativní (bude tedy použitelná obecně, nikoli jen pro kódy 'SPAM' a 'OK').
- Při testování spam filtru bude `pos_tag="SPAM"` a `neg_tag="OK"`

- Po vytvoření instance budou všechny čtyři hodnoty nastaveny na 0.
- Třída bude mít metodu `as_dict()` , která vrátí matici záměn ve formě slovníku s položkami `tp`, `tn`, `fp`, `fn` .
- Třída bude mít metodu `update(truth, prediction)` , která zvýší hodnotu příslušného čítače (TP, TN, FP, nebo FN) o 1 na základě porovnání hodnot proměnných `truth` a `prediction` s hodnotami `pos_tag` a `neg_tag` . Bude-li hodnota některého z parametrů `truth` nebo `prediction` různá od `pos_tag` i `neg_tag` , metoda vyhodí výjimku `ValueError` .
- Třída bude mít metodu `compute_from_dicts(truth_dict, pred_dict)` , která napočítá statistiky TP, TN, FP a FN ze dvou slovníků: v prvním bude správná klasifikace emailů, ve druhém klasifikace emailů predikovaná filtrem.

K čemu nám to bude:

- Třída `BinaryConfusionMatrix` bude tvořit základ pro hodnocení úspěšnosti filtru.
- Třidu bude možno používat následujícím způsobem:

```
>>> cm1 = BinaryConfusionMatrix(pos_tag=True, neg_tag=False)
>>> cm1.as_dict()
{'tp': 0, 'tn': 0, 'fp': 0, 'fn': 0}
>>> cm1.update(True, True)
>>> cm1.as_dict()
{'tp': 1, 'tn': 0, 'fp': 0, 'fn': 0}

>>> truth_dict = {'em1': 'SPAM', 'em2': 'SPAM', 'em3': 'OK', 'em4': 'OK'}
>>> pred_dict = {'em1': 'SPAM', 'em2': 'OK', 'em3': 'OK', 'em4': 'SPAM'}
>>> cm2 = BinaryConfusionMatrix(pos_tag='SPAM', neg_tag='OK')
>>> cm2.compute_from_dicts(truth_dict, pred_dict)
>>> cm2.as_dict()
{'tp': 1, 'tn': 1, 'fp': 1, 'fn': 1}
```

Třída tedy bude mít alespoň 3 veřejné metody: `as_dict()` , `update()` a `compute_from_dicts()` .

as_dict()	Vrátí matici záměn ve formě slovníku.
Vstup:	Nic.
Výstup:	Slovník obsahující klíče tp, tn, fp, fn a jejich hodnoty.
Efekty:	Nic.

update(truth, pred)	Zvyš hodnotu jednoho ze 4 čítačů podle hodnot <code>truth</code> a <code>pred</code> .
Vstup:	Skutečná a predikovaná třída.
Výstup:	Nic.
Efekty:	Zvýšení hodnoty jednoho z čítačů TP, TN, FP, FN, nebo výjimka <code>ValueError</code> .

compute_from_dicts(truth_dict, pred_dict)	Napočítej celou matici záměn na základě hodnot uložených ve slovnících.
Vstup:	Slovníky obsahující skutečné a predikované třídy pro množinu objektů (emailů).
Výstup:	Nic.
Efekty:	Prvky matice se nastaví na počty pozorovaných TP, TN, FP, FN.

Pozn.: Můžete očekávat, že slovníky budou mít stejnou množinu klíčů. Rozmyslete si ale, jak byste chtěli, aby se metoda chovala v případě, že bychom jí předali slovníky s různými množinami klíčů.

Funkce "quality_score()"

Úkol:

- Vytvořte funkci `quality_score(tp, tn, fp, fn)` v modulu `quality.py` .
- Funkce spočte skóre kvality z předaných hodnot matice záměn (viz [Spam filter: Specifikace \[/b211/courses/b4b33rph/cviceni/spam/specifikace#mira_kvality_filtru\]](#)).

<code>quality_score(tp, tn, fp, fn)</code>	Vypočti skóre kvality pro danou matici záměn.
Vstupy	Čtveřice hodnot TP, TN, FP, FN.
Výstupy	Číslo mezi 0 a 1 udávající naši míru kvality predikce.

Funkce "compute_quality_for_corpus()"

Následující specifikace pro funkci `compute_quality_for_corpus()` jsou povinné a musíte jim vyhovět!

Úkol:

- V modulu `quality.py` vytvořte funkci `compute_quality_for_corpus(corpus_dir)`, která ohodnotí kvalitu predikcí (filtru) na základě údajů v souborech `!truth.txt` a `!prediction.txt` v zadaném korpusu.
- Skutečnou i predikovanou klasifikaci umíte načíst ve formě slovníků pomocí `read_classification_from_file()`.
- Matici záměn pro daný korpus z načtených slovníků umíte vytvořit pomocí metody `compute_from_dicts()` třídy `BinaryConfusionMatrix`.
- Skóre kvality z prvků matice záměn umíte spočítat funkcí `quality_score()`.

K čemu nám to bude:

- Funkce bude sloužit k hodnocení úspěšnosti filtru.
- unit test k této funkci předpokládá nastavení `pos_tag="SPAM"` a `neg_tag="OK"`, které odpovídá finálnímu spam filtru.

<code>compute_quality_for_corpus(corpus_dir)</code>	Vypočti kvalitu predikcí v korpusu.
Vstupy	Adresář ohodnoceného korpusu (obsahující soubory <code>!truth.txt</code> a <code>!prediction.txt</code>).
Výstupy	Číslo mezi 0 a 1 udávající naši míru kvality predikce.

courses/b4b33rph/cviceni/spam/krok3.txt · Last modified: 2019/11/27 12:56 by xposik

Copyright © 2024 CTU in Prague | Operated by IT Center of Faculty of Electrical Engineering |

Bug reports and suggestions Helpdesk CTU