Współczesne Narzędzia Obliczeniowe
Laboratorium 5, 6
Filtr antyspamowy

Zadania do wykonania (Lab 5):

1. Napisać skrypt w Pythonie który przeskanuje folder w poszukiwaniu plików tekstowych zawierających maile, wczytać je i sparsować je do postaci listy **klas** (2 pkt)
2. Umożliwić wystąpienia polskich znaków, konwertując kodowanie tekstu na ASCII (1 pkt)
3. Stworzyć **słowniki** (zmienne słownikowe, wewnątrz programu) spamu i hamu na podstawie liczby wystąpień danego słowa w mailach (1 pkt) i policzyć prawdopodobieństwa warunkowe słów $P(word_i|SPAM)$ i $P(word_i|HAM)$ (1 pkt)

Zadania do wykonania (Lab 6):

1. Przetestować wykrywanie spamu (na podstawie słowników z poprzedniego zadania) na pliku example.txt - jakie jest prawdopodobieństwo, że dana wiadomość jest spamem (1 pkt)
2. W pliku dict.xml znajduje się słownik wyrazów wraz z prawdopodobieństwami im przypisanymi, należy wczytać wyrazy do słowników (czystych) i przetestować wykrywanie spamu - example.txt (1 pkt)
3. Zmienić sposób obliczania prawdopodobieństwa - wygładzanie Lapace'a $k = 2$, $possible_{types} = 2$ (SPAM $\vee$ HAM) (1 pkt), połączenie obu słowników - dict.xml i z poprzedniego zadania (1 pkt)
4. Skonstruować inteligentny słownik - odmiana wyrazów (1 pkt)

Trochę prawdopodobieństwa ((3) przy założeniu zdarzeń niezależnych):

$$P(SPAM) = \frac{count(massages\ in\ SPAM)}{count(massages)} \tag{1}$$

Wygładzanie Laplace'a

$$P(SPAM) = \frac{count(massages\ in\ SPAM) + k}{count(massages) + k * possible_{types}} \tag{2}$$

$$P(word_i|SPAM) = \frac{count(word_i\ in\ SPAM)}{count(words\ in\ SPAM) + k * possible_{types})} \tag{3}$$

$$P(massage|SPAM) = \prod_i P(word_i|SPAM) \tag{4}$$

$$P(SPAM|massage) = \frac{P(massage|SPAM) * P(SPAM)}{P(massage|SPAM) * P(SPAM) + P(massage|HAM) * P(HAM)} \tag{5}$$

Przydatne polecenia - słowniki:

```
>>> knights = {'gallahad': 0.9, 'robin': 0.000005, 'lancelot' : 0.1}
>>> knights
{'gallahad': 0.9, 'robin': 5e-06, 'lancelot': 0.1}

>>> knights = dict(gallahad=0.9, robin=0.000005, lancelot=0.1)          # konstruktor
>>> knights
{'gallahad': 0.9, 'robin': 5e-06, 'lancelot': 0.1}

>>> knights['robin']
5e-06
>>> knights['abraham'] = 0.0000000001            # dodawanie elementu
>>> knights
{'gallahad': 0.9, 'abraham': 1e-10, 'robin': 5e-06, 'lancelot': 0.1}

>>> del knights['abraham']                       # usuwanie elementu
>>> knights
{'gallahad': 0.9, 'robin': 5e-06, 'lancelot': 0.1}


>>> knights['robin'] = 0.2
>>> knights
{'gallahad': 0.9, 'robin': 0.2, 'lancelot': 0.1}


>>> for k, v in knights.iteritems():             # iterkeys() -tylko po kluczach
...     print k, v                               # itervalues() -tylko po wartosciach
gallahad 0.9
robin 0.2
lancelot 0.1
```

Przydatne polecenia - inne:

```
>>> import os
>>> os.listdir('.')                      # klasyczny dir

>>> import glob
>>> glob.glob('./*.txt')                 # filtrowanie listy plikow

>>> import codecs                        # otwieranie plikow z innym kodowaniem
>>> f = codecs.open('file','r','UTF-8')
```