Wiadomości wstępne, podstawowe trudności w PJN, wyrażenia regularne

mgr inż. Dawid Wisniewski - Przetwarzanie języka naturalnego

29 February 2020

Laboratoria 1 i

1. Jak zaliczyć?
2. Po co mi to?
3. Konkrety część 1: wyrażenia regularne

Laboratoria 1 ii

Składowe oceny:

Typ	Punkty jednostkowe	Ilość	Punkty w sumie
Zadania	5	10	50
Projekt	15/20/30	≤ 1	0-30

Projekt jest nieobowiązkowy.

10 spotkań zadaniowych, reszta konstulacje projektowe (projekty nieobowiązkowe, dlatego konstulacje dla chętnych).

Wymagana obecność do zaliczenia: 8/10 zajęć zadaniowych, reszta dla chętnych.

Laboratoria 1 iii

Zadania na zajęciach, jeśli ktoś nie zdąży można skońćzyć w domu i wysłć je a mailem dwisniewski@cs.put.poznan.pl lub pokazać na kolejnych zajęciach laboratoryjnych bez strat punktów. Wysłąnie/pokazanie po tym terminie skutkuje podzieleniem uzyskanych punktów przez 2.

Laboratoria 1 iv

10 laboratoriów praktycznych z użyciem Pythona + Jupyter Notebook:

- Lab1: **Szybkie wyszukiwanie/operowanie na tekście - wyrażenia regularne**. (NLTK, SpaCy)
- Lab2: **Klasyfikacja tekstów (Tworzenie reprezentacji BagOfWords, Tokenizacja, Lematyzacja, Stemming, normalizacja TF-IDF, SVM, Naiwny Bayes)** (Pandas, Numpy, sklearn, NLTK)
- Lab3: **NGramy (Reprezentacja NGram vs BagOfWords, detekcja języka, generowanie tekstu)**. (sklearn, NumPy, Pandas, NLTK)
- Lab4: **Embeddingi jako niskowymiarowa alternatywa dla BagOfWords/NGram (podobieństwo w przestrzeni embeddingów, embeddingi do klasyfikacji) + Poprawianie literówek z użyciem odległości edycyjnej** (sklearn, NumPy, Pandas, NLTK)
- Lab5: **Sieci neuronowe (Przypomnienie z SI, sieci jako sekwencja operacji na macierzach, sieć implementowana bez użycia frameworków)**. (NumPy, sklearn, matplotlib)
- Lab6: **Tworzenie zasobów (Crawling/Scraping danych z sieci, tworzenie zasobów)**. (BeautifulSoup)
- Lab7: **Sieci rekurencyjne w przetwarzaniu tekstu (RNN od podstaw bez użycia frameworków, idea historii w RNN)**. (Numpy)
- Lab8: **Detekcja sentymentu z użyciem zaawansowanych architektur sieci (GRU/LSTM/CNN)**. (Keras lub PyTorch)
- Lab9: **Wykrywanie encji nazwanych i fraz rzeczownikowych.** (NLTK, SpaCy, sklearn, pycrfsuite)
- Lab10: **Sumaryzacja poprzez wyszukiwanie zdań kluczowych (Key-sentence extraction), modelowanie tematów (topic modelling - LDA), Ekstrakcja informacji (Drzewo zależnościowe, rozbiór gramatyczny zdań)**. (SpaCy, NLTK, gensim)

Laboratoria 1 v

Mamy około 13 spotkań (Czyli 3 konstultacyjne).

Tydzień nieparzysty / Tydzień parzysty	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
I połowa semestru:					
1					
2	2-3-2020	3-3-2020	4-3-2020	5-3-2020	6-3-2020
3	9-3-2020	10-3-2020	11-3-2020	12-3-2020	13-3-2020
4	16-3-2020	17-3-2020	18-3-2020	19-3-2020	20-3-2020
5	23-3-2020	24-3-2020	25-3-2020	26-3-2020	27-3-2020
6	30-3-2020	31-3-2020	1-4-2020	2-4-2020	3-4-2020
7	6-4-2020	7-4-2020	8-4-2020	9-4-2020	**8.**
8	**8.**	**8.**	~~15-4-2020~~	16-4-2020	17-4-2020
9	~~20-4-2020~~	~~21-4-2020~~	~~22-4-2020~~	23-4-2020	24-4-2020
II połowa semestru:					
10	27-4-2020	28-4-2020	29-4-2020	30-4-2020	**9.**
11	4-5-2020	5-5-2020	6-5-2020	7-5-2020	8-5-2020
12	11-5-2020	12-5-2020	13-5-2020	14-5-2020	15-5-2020
13	18-5-2020	19-5-2020	20-5-2020	21-5-2020	22-5-2020
14	25-5-2020	26-5-2020	27-5-2020	28-5-2020	29-5-2020
15	1-6-2020	2-6-2020	3-6-2020	4-6-2020	5-6-2020
16	8-6-2020	9-6-2020	10-6-2020	**10.**	12-6-2020
17	~~15-6-2020~~	~~16-6-2020~~			

Punkty a oceny:

Zakres punktów	Ocena
<0-24>	2.0
<25-32>	3.0
<33-40>	3.5
<41-49>	4.0
<50-55>	4.5
<55-∞)	5.0

Laboratoria 1 vii

Dlaczego zajmować się przetwarzaniem języka?

- W sieci mamy całkiem dużo tekstu.
- Bardzo dynamiczny rozwój (GPT-2 od OpenAI).
- Komercyjna potrzeba automatyzacji procesów i zrozumienia użytkowników.

Laboratoria 1 viii

Trudności związane z przetwarzaniem języka:

Język naturalny jest wysoce niejednoznaczny.

- Co oznacza Apple w wyrażeniu 'Apple is great'? Owocem? Nazwą firmy?
- Świetny telefon, WCALE nie popsuł mi się miesiąc po zakupie – czy to zdanie zawiera w sobie pozytywne czy negatywne emocje? Czy to ironia?
- Marcin pojechał z Bartkiem na uczelnię, on zawsze się spóźnia. - On, czyli kto?
- Czy nie jesteś zajęty? - Tak/Nie.
- Pojechałem do Żabki po bułki.

Laboratoria 1 ix

Co miał na myśli autor mówiąc: I made her duck?

- Upiekłem/am dla niej kaczkę.
- Upiekłem/am kaczkę należącą do niej.
- Stworzyłem/am (z papieru?) kaczkę, którą teraz posiada.
- Spowodowałem/am, że zrobiła unik.
- Machnąłem różdżką i zamieniłem ją w kaczkę.

Laboratoria 1 x

Wyrażenia regularne - co to?

Specjalnie interpretowany **ciąg znaków** pozwalający na **szybkie wyszukiwanie** w tekście zadanych **wzorców**.

Wyszukiwanie określonego adresu e-mail lub wyszukiwanie wszystkiego co wygląda jak adres e-mail w dokumentach.

Wyrażenia regularne - do czego mi się to przyda?

Wyrażenia regularne to dobre narzędzie gdy:

1. trzeba szybko przeszukać plik(i) w poszukiwaniu wzorców.
2. trzeba usunąć pewne fragmenty plików.
3. trzeba zinterpretować zawartość plików.

Wyrażenia regularne dostępne w wielu narzędziach

Przykład

```
cat pracownicy_PP.txt | grep "Wi.* "
```

Laboratoria 1 xii

Czy wyrażenia regularne są trudne?

Skądże.

Laboratoria 1 xiii

Some people, when confronted with a problem, think, I know, I'll use regular expressions. Now they have two problems. - Jamie Zawinski

W późniejszym czasie powstały też różne wariacje na ten temat:

Some people see a problem and think 'I know, I'll use Java!' Now they have a ProblemFactory.

Some people, when confronted with a problem, think 'I know, I'll use multithreading'. Nothhw tpe yawrve o oblems.

Some people, when confronted with a problem, think, 'I know, I'll use AI'. Now the problem says they are they problem.

Proste wyrażęnie regularne: ’Adam’

Laboratoria 1 xv

’Trochę’ gorsze wyrażenie - to wykrywające wszystkie maile i tylko maile, zgodnie ze standardem RFC.

http://www.ex-parrot.com/pdw/Mail-RFC822-Address.html

Wyrażenia regularne i

Kto jest największym dzbanem w Polsce?

Poszukajmy wyrażeń dzbanopodobnych: dzban, dzbanie, dzbanom, dzbn (jeśli ktoś ma klawiaturę bez 'a'), dzbababan, ...

- dzban $\rightarrow$ **dzban**, pod**dzban**, nie**dzban**owa

- dzban $\rightarrow$ **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b $\rightarrow$ **dzban**

- dzban $\rightarrow$ **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b $\rightarrow$ **dzban**
- dzb.n $\rightarrow$ **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**
- dzb[a-c]n → **dzban**, **dzbbn**, **dzbcn**

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**
- dzb[a-c]n → **dzban**, **dzbbn**, **dzbcn**
- dzba?n → **dzban**, **dzbn**

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**
- dzb[a-c]n → **dzban**, **dzbbn**, **dzbcn**
- dzba?n → **dzban**, **dzbn**
- dzba*n → **dzbn**, **dzban**, **dzbaaan**, **dzbaaaaaan**. ...

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**
- dzb[a-c]n → **dzban**, **dzbbn**, **dzbcn**
- dzba?n → **dzban**, **dzbn**
- dzba*n → **dzbn**, **dzban**, **dzbaaan**, **dzbaaaaaan**. ...
- dzba+n → **dzban**, **dzbaaan**, **dzbaaaaaan**. ...

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**
- dzb[a-c]n → **dzban**, **dzbbn**, **dzbcn**
- dzba?n → **dzban**, **dzbn**
- dzba*n → **dzbn**, **dzban**, **dzbaaan**, **dzbaaaaaan**. ...
- dzba+n → **dzban**, **dzbaaan**, **dzbaaaaaan**. ...
- dz(ba)+n → **dzban**, **dzbaban**, **dzbababababan**. ...

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**
- dzb[a-c]n → **dzban**, **dzbbn**, **dzbcn**
- dzba?n → **dzban**, **dzbn**
- dzba*n → **dzbn**, **dzban**, **dzbaaan**, **dzbaaaaaan**. ...
- dzba+n → **dzban**, **dzbaaan**, **dzbaaaaaan**. ...
- dz(ba)+n → **dzban**, **dzbaban**, **dzbababababan**. ...
- dz(ba){1,7}n → jak wyżej, gdzie **ba** powtórzone od 1 do 7 razy

- dzban → **dzban**, pod**dzban**, nie**dzban**owa
- \bdzban\b → **dzban**
- dzb.n → **dzban**, **dzbbn**, **dzbZn**, **dzb1n**, **dzb,n**
- dzb[abc]n → **dzban**, **dzbbn**, **dzbcn**
- dzb[a-c]n → **dzban**, **dzbbn**, **dzbcn**
- dzba?n → **dzban**, **dzbn**
- dzba*n → **dzbn**, **dzban**, **dzbaaan**, **dzbaaaaaan**. ...
- dzba+n → **dzban**, **dzbaaan**, **dzbaaaaaan**. ...
- dz(ba)+n → **dzban**, **dzbaban**, **dzbababababan**. ...
- dz(ba){1,7}n → jak wyżej, gdzie **ba** powtórzone od 1 do 7 razy
- dzban(ek|uszek|owy|em) → **dzbanek**, **dzbanuszek**, **dzbanowy**, **dzbanem**

Zachłanność dopasowań

Wyrażenia regularne domyślnie starają się dopasować najdłuższy możliwy podciąg spełniający warunki opisane w wyrażeniu

Zachłanność dopasowań

Wyrażenia regularne domyślnie starają się dopasować najdłuższy możliwy podciąg spełniający warunki opisane w wyrażeniu

Wyrażenie regularne: a.*a, tekst wejściowy: analfabeta

Dopasowanie **analfabeta** - najdłuższy ciąg zaczynający się na a i kończący na a z doowlną ilością dowolnych znaków pomiędzy.

Zachłanność dopasowań

Wyrażenia regularne domyślnie starają się dopasować najdłuższy możliwy podciąg spełniający warunki opisane w wyrażeniu

Wyrażenie regularne: a.*a, tekst wejściowy: analfabeta

Dopasowanie **analfabeta** - najdłuższy ciąg zaczynający się na a i kończący na a z doowlną ilością dowolnych znaków pomiędzy.

Działanie niezachłanne (poszukiwanie najkrótszych ciągów): a.*?a
Dopasowanie: **ana**lf**abeta**

Zachłanność dopasowań

Wyrażenia regularne domyślnie starają się dopasować najdłuższy możliwy podciąg spełniający warunki opisane w wyrażeniu

Wyrażenie regularne: a.*a, tekst wejściowy: analfabeta

Dopasowanie **analfabeta** - najdłuższy ciąg zaczynający się na a i kończący na a z doowlną ilością dowolnych znaków pomiędzy.

Działanie niezachłanne (poszukiwanie najkrótszych ciągów): a.*?a
Dopasowanie: **ana**lf**abeta**

Domyślnie silniki wyrażeń regularnych zwracają listę nienakładających się dopasowań (stąd brak dopasowania an**alfa**beta).

Grupy nazwane

qwert(?P<nazwa>abc)xxx

Zadania

Moodle -> 1 stopień -> Przetwarzanie języka naturalnego ->
Laboratoria 1