Tokenisering med regulære uttrykk

Lilja Charlotte Storset
liljacs@ifi.uio.no

Fredrik Aas Andreassen
fredaan@ifi.uio.no

Uttrykket

Inspirasjon: Token kan inneholde mye forskjellig. I stedet for å spesifisere hva token skal kunne bestå av, hva med å spesifisere hva token *ikke* kan bestå av?

```
(?:[A-Z]\.)+|\d{1,3}(?:[\s,.]\d{3})+|[^\w\s]|\b\S+\b[#+-]*
```

Enkle ASCII-forkortelser:

T.V.

Store tall:

3 121 24.601

Enkeltegnsetting:

Punktum, komma, parenteser

Alt som ikke er tomrom med mulighet for et lite utvalg tegnsetting til slutt:

skoleskyss jblack@mail.yahoo.com C++ 142.32.48.231 метрополитен 2022

Implementasjon i Python

```
import re

with open('tekst.txt', encoding='utf-8') as file:
    file_content = file.read()

token_pattern = r'(?:[A-Z]\.)+|\d{1,3}(?:[\s,.]\d{3})+|[^\w\s]|\b\S+\b[#+-]*'

all_tokens = re.findall(token_pattern, file_content)
```

Evaluering

For most languages and particular domains within them there are unusual specific tokens that we wish to recognize as terms , such as the programming languages **C++** and **C#** , aircraft names like **B-52** , or a **T.V.** show name such as **M*A*S*H** – which is sufficiently integrated into popular culture that you find usages such as **M*A*S*H-style** hospitals . Computer technology has introduced new types of character sequences that a tokenizer should probably tokenize as a single token , including email addresses (**jblack@mail.yahoo.com**) , web URLs (**http://stuff.big.com/new/specials.html**) , numeric IP addresses (**142.32.48.231**) , package tracking numbers (**1Z9999W99845399981**) , and more . One possible solution is to omit from indexing tokens such as monetary amounts , numbers , and URLs , since their presence greatly expands the size of the vocabulary . However , this comes at a large cost in restricting what people can search for . For instance , people might want to search in a bug database for the line number where an error occurs . Items such as the date of an email , which have a clear semantic type , are often indexed separately as document metadata (see Section **6.1** , page 110) .

(An Introduction to Information Retrieval, side 24)

Evaluering

I 1960 hadde Norge en befolkning på **3,581** millioner , i 2020 **5 379 000** .

1963 begann die Stadt mit dem Bau eines **U-Stadtbahn-Netzes** . Die erste Teilstrecke konnte ab 1968 genutzt werden . Im Oktober 1966 fanden Proteste gegen die KVB statt , als der Fahrpreis für Schüler und Studenten um mehr als die Hälfte erhöht werden sollte . Rund **10.000** Schüler und Studenten protestierten im Rahmen von insgesamt dreitägigen Aktionen am Rudolfplatz [6] , bevor die Protestaktion von der Polizei gewaltsam aufgelöst wurde . In Zusammenarbeit mit den Clouth-Gummiwerken entwickelten die Kölner Verkehrs-Betriebe 1972 das Kölner Ei , ein elastisches Schienenlager , das oft bei schotterlosem Oberbau verwendet wird .

(https://de.wikipedia.org/wiki/K%C3%B6lner_Verkehrs-Betriebe#Geschichte)

Evaluering

Tar du med potetgull til festen den **31.12.2022** , **@** Rune ?

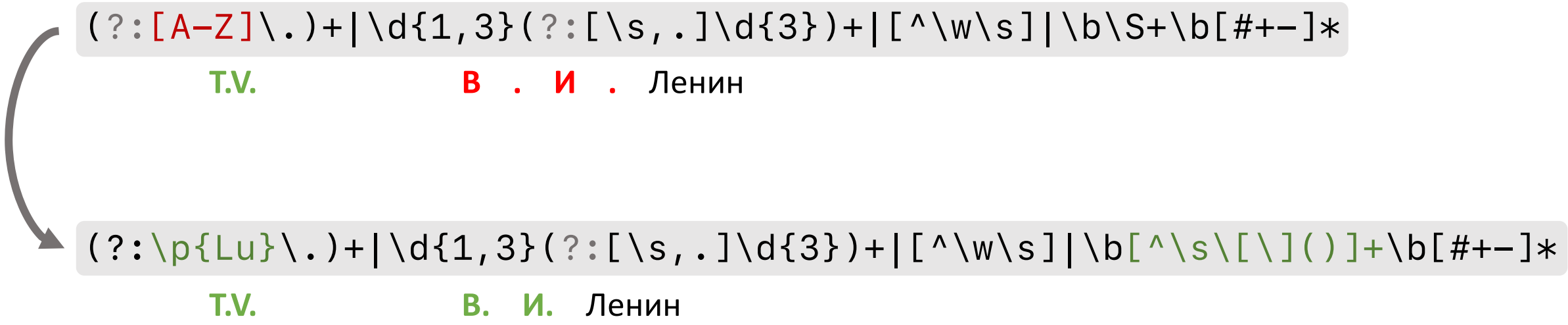
Петербу́ргский метрополите́н (до июля 1992 года — Ленинградский ордена Ленина метрополитен имени **В . И .** Ленина) — скоростная внеуличная транспортная система Санкт-Петербурга и Ленинградской области [8] . Открыт 15 ноября 1955 года , став вторым метрополитеном по дате открытия в СССР после московского , открытого 15 мая 1935 года . Петербургский метрополитен эксплуатирует ГУП « Петербургский метрополитен » (полное название — Санкт-Петербургское государственное унитарное предприятие « Петербургский метрополитен ») .

(https://ru.wikipedia.org/wiki/петербургский_метрополитен)

Støtte for enkle Unicode-forkortelser

`\w` gjenkjenner Unicode, men inkluderer siffer og minuskler (også kalt "små bokstaver").

Dette kan forstyrre gjenkjenning av desimaltall på utenlandsk format, med punktum i stedet for komma, og gjenkjenne ord bestående av én bokstav i slutten av setninger som forkortelser.



Ny implementasjon i Python

```
import regex

with open('tekst.txt', encoding='utf-8') as file:
    file_content = file.read()

token_pattern = r'(?:\p{Lu}\.)+|\d{1,3}(?:[\s,.]\d{3})+|[^\w\s]|\b[^\s\[\]()]+\b[#+-]*'

all_tokens = regex.findall(token_pattern, file_content)
```

`pip install regex`