

Twitter-Daten sammeln und mit Elasticsearch analysieren

Nobutake Kamiya (Universität Zürich, Universitätsbibliothek)

Was ich hier anbiete:

- Wie man Twitter-Daten sammelt (Python)
- Einführung der Tools für Datenanalyse (Elasticsearch und weitere Plugins)
- Denkanstoß für Handhabung der Daten / Datenmanagement

Das Ziel ist hier

Der Link führt zu meinem lokalen Server. Deshalb funktioniert er nur von meinem PC...

Was ich hier **NICHT** anbiete:

- Eine wissenschaftliche Erkenntnis
- Fachspezifische Forschungsmethode

Tweets sammeln

Tweets sammeln : Methode

- Twitter API v2 - Academic Research
- Python

Übersicht von Twitter API)

Essential	Elevated	Academic Research
500'000 tweets / Monat	2 Mio. Tweets / Monat	10 Mio. Tweets / Monat
Full-Archive-Suche nicht möglich	Full-Archive-Suche nicht möglich	Full-Archive-Suche und Full-Archive-Count
XX	XX	Advanced search operator

Python-Code

Die gesamten Codes sind [hier](#) zu finden.

Python-Code : Query 1

Datum und Uhrzeit von Japanischer Zeit zu UTC (koordinierte Weltzeit)

```
jst_st = datetime.datetime(2021, 3, 1, 0, 0, 0, 0, datetime.timezone(timedelta(hours=+9)))
jst_et = datetime.datetime(2021, 4, 1, 0, 0, 0, 0, datetime.timezone(timedelta(hours=+9)))

utc_st = jst_st.astimezone(timezone.utc)
utc_st = utc_st.isoformat()
utc_et = jst_et.astimezone(timezone.utc)
utc_et = utc_et.isoformat()
```

Python-Code : Query 2

Suchbedingungen

```
query_params = {'query': '(オリンピック lang:ja -is:retweet) OR (五輪 lang:ja -is:retweet)',  
                'tweet.fields': 'created_at,id,text,public_metrics',  
                'start_time': utc_st,  
                'end_time': utc_et,  
                'max_results': 500}
```

Weitere Regeln siehe [hier](#)

Data cleaning

Beispiel: [Ein Tweet von der IOC \(japanisch\)](#).



Python-Code : Data cleaning

```
# User-Name (Erwähnungen), URL, Hash-tags und Zeilenumbrüche aus dem Tweets entfernen
tw_text = re.sub(r'@\w+', '', ['text'])
tw_text = re.sub(r'(http|https)://[0-9a-zA-Z\./]+', '', tw_text)
tw_text = re.sub(r'#+?(\s|$)', '', tw_text)
tw_text = re.sub(r'\n', '', tw_text)
# Kana soll immer in Fullwidth-Zeichen dargestellt werden
tw_text = mojimoji.han_to_zen(tw_text, kana=True, digit=False, ascii=False)
# Digit und Ascii sollen immer in Halfwidth-Zeichen dargestellt werden
tw_text = mojimoji.zen_to_han(tw_text, kana=False, digit=True, ascii=True)
```

Besonderheiten bei der Twitter-Daten

- "Academic researchers are permitted to distribute an unlimited number of Tweet IDs and/or User IDs [...]", aber mehr nicht.
[Developer terms \(unter "content redistribution"\)](#).
- Tweets können gelöscht werden!

Tweets sammeln - Zusammenfassung

- Nutzungsbedingungen der Daten kennenlernen!
- Codes für die Query und für Data cleaning ebenso publizieren
- Das Datum der Ausführung dokumentieren

Elasticsearch (ES)

Was ist Elasticsearch?

- Eine Suchmaschine basierend auf Lucene
- Nutzung der Standarddistribution ist kostenlos
- Verwendet auch in [次世代デジタルライブラリー](#) von der NDL

Plugins für Japanisch

- [Kuromoji](#)
- [ICU \(International Components for Unicode\)](#)
- [Sudachi \(die neueste Version für ES 5.6?\)](#)

In diesem Beispiel werden **Kuromoji** und **ICU** für Analyzer verwendet

Einstellung des Analyzers

- Konfiguration kann in JSON-Format geschrieben werden
- Char_filter (Normalisierung der Schriftzeichen, fakultativ)
- Tokenizer (Worttrennung [z.B. N-gram], notwendig, nur ein Tokenizer anwendbar)
- Token_filter (Wörter [z.B. Stopwörter] werden nach bestimmten Regeln gefiltert, fakultativ)

Einstellung mit dem Beispielsatz

"そのオリンピック選手は身長196㍑という長身だった。人々はおどろいた。二〇〇〇人がオリンピックを観戦しながらコンピューターをつかっていた"

Char_filter - Einstellung

```
{
  "settings": {
    "analysis": {
      "analyzer": {
        "my_kuromoji_analyzer": {
          "type": "custom",
          "char_filter" : ["icu_normalizer", "kuromoji_iteration_mark"],
          "tokenizer": "keyword"
        }
      }
    }
  }
}
```

Der "Keyword"-Tokenizer gibt den Satz so zurück (s. [hier](#))

Ergebnis

そのオリンピック選手は身長**196センチ**という長身だった。**人人**はおどろいた。二〇〇〇人がオリンピックを観戦しながらコンピューターをつかっていた

Tokenizer-Einstellung

```
{
  "settings": {
    "analysis": {
      "analyzer": {
        "my_kuromoji_analyzer": {
          "type": "custom",
          "char_filter" : ["icu_normalizer", "kuromoji_iteration_mark"],
          "tokenizer": "kuromoji_tokenizer"
        }
      }
    }
  }
}
```

Ergebnis

[Ergebnis in JSON-Format](#)

Token_filter-Einstellung 1

```
[...]
  "filter": ["kuromoji_baseform", "kuromoji_part_of_speech", "kuromoji_stemmer", "ja_stop", "kuromoji_number", "synonym_filter"]
    },
    "filter": {
      "synonym_filter": {
        "type": "synonym",
        "synonyms": ["オリンピック,五輪"]
      }
    }
  }
}
```


Token_filter-Einstellung 2

Einige Token_filter von Kuromoji sind verwendet

- kuromoji_baseform
- kuromoji_part_of_speech

Zusätzlich ist noch "synonym_filter" eingesetzt.

- "オリンピック" und "五輪"

Ergebnis

[Hier](#)

Die endgültige Einstellung

...Falls man sich dafür interessiert... [Hier](#)

Elasticsearch - Zusammenfassung

- Mit ES kann man die große Datenmenge behandeln und analysieren
 - Durch API kann man weiter die Daten verarbeiten z.B. für Netzwerkanalyse
 - [Kibana](#) (ein Plugin von ES) erlaubt eine einfache GUI-Verarbeitung
- Hier sollte man auch die Konfiguration bekannt machen

Denkanstoß - 1

- Behandlung der großen Menge von japanischen Texten ist durchaus möglich
- Für die wissenschaftlichen Kommunikation ist die Nachvollziehbarkeit wichtig
- Open Data-Gedanken ist deshalb sehr wichtig

Denkanstoß - 2

- Falls die Rohdaten nicht publiziert werden kann, sollten mindestens die Codes für Datensammlung und Verarbeitung publiziert werden
- Für die wiss. Bibliothekare wäre interessant, die Information über anwendbaren Datenquellen und derer Nutzungsbedingungen zu wissen/vermitteln

Vielen Dank!