Politechnika Wrocławska

Projekt

Projektowanie systemów z dostępem w języku naturalnym

Opracowanie aplikacji wspomagającej tworzenie korpusu z tekstami w języku polskim.

Authors:
Rafał Pieniążek
Jakub Pomykała

Supervisor:
Dr inż. Dariusz Banasiak

18 grudnia 2017

Spis treści

1 Dodatek do opisu parametrów **2**
1.1 Tryb TXT 2
1.2 Tryb JSON 2

2 Dodatek do implementacji parserów **3**
2.1 Ekstrakcja tekstu z artykułu 3
2.2 Ekstrakcja autora 5
2.3 Ekstrakcja adresów URL 6

3 Dodatek do wniosków **7**

1 Dodatek do opisu parametrów

1.1 Tryb TXT

Tryb TXT zawiera jedynie treść samego artykułu bez dodatkowych metadanych.

1.2 Tryb JSON

W trybie JSON zostanie zapisana cała klasa *Article* przedstawiona poniżej.

```
public class Article {
        private String source;
        private String title;
        private String body;
        private HashMap<String, String> metadata;
    //...
}
```

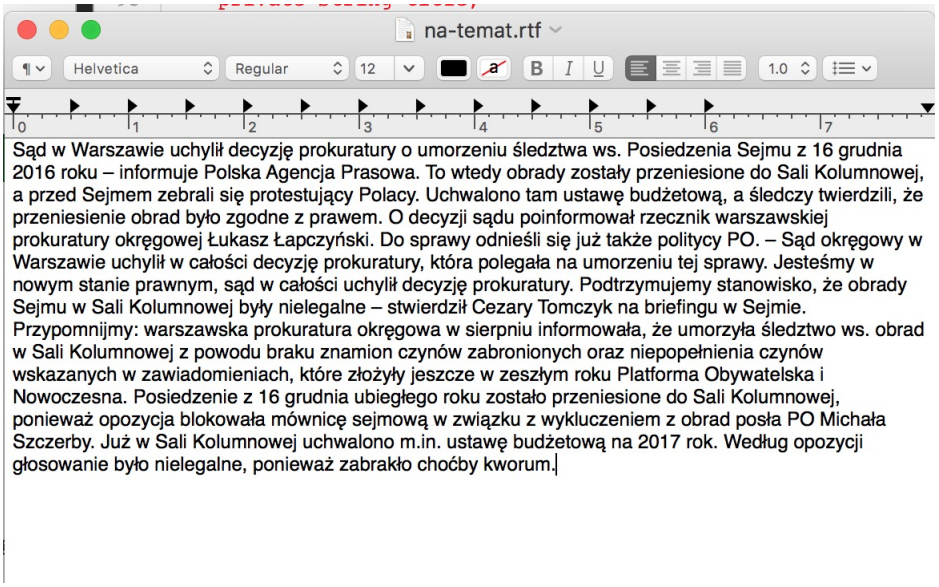
Przykłas wynikowego pliku *json*.

```
{
  "source" : "https://niebezpiecznik.pl/post/zatrzymania-w-sprawie-wycieku
  -z-bazy-pesel-komornik-naduzywal-uprawnien-na-rzecz-windykatora/",
  "title" : "Komornik z windykatorem naduzywali dostepu do bazy PESEL.
  Prokuratura i ABW zamknely kramik",
  "body" : "Cztery osoby zatrzymane (...)",
  "metadata" : {
    "date" : "7:58 14/12/2017",
    "author" : "Marcin Maj"
  }
}
```

2 Dodatek do implementacji parserów

2.1 Ekstrakcja tekstu z artykułu

Wszystkie akapity tekstu zostały połączone w jeden tekst



W przypadku strony https://niebezpiecznik.pl metoda *parseLink* przyjmuje URL artykułu i na jego podstawie dokonuje ekstrakcji poszczególnych metadanych.

```
private Article parseLink(String articleUrl) {
    try {
        Article article = new Article(articleUrl);
        Document doc = JsoupConnector.connectThrowable(articleUrl, SLEEP_TIME);
        article.setTitle(doc.select(".postcontent").select("h1").text());
        article.getMetadata().put("author", getAuthor(doc));
        article.getMetadata().put("keywords", getKeywords(doc));
        article.getMetadata().put("date", getDate(doc));
        article.setBody(getBody(doc));

        return article;
    } catch (Exception e) {
        return null;
    }
}
```

Usuwane są wszystkie teksty które znajdują się w tagach *blockquote* wskazujące na obecność cytatów w tekście.

```
private String getBody(Document doc) {
    doc.select("blockquote").remove();
    return doc.select(".postcontent").select("p").text();
}
```

getDate przy pomocy wyrażenia regularnego wyłuskuje datę napisania artykułu.

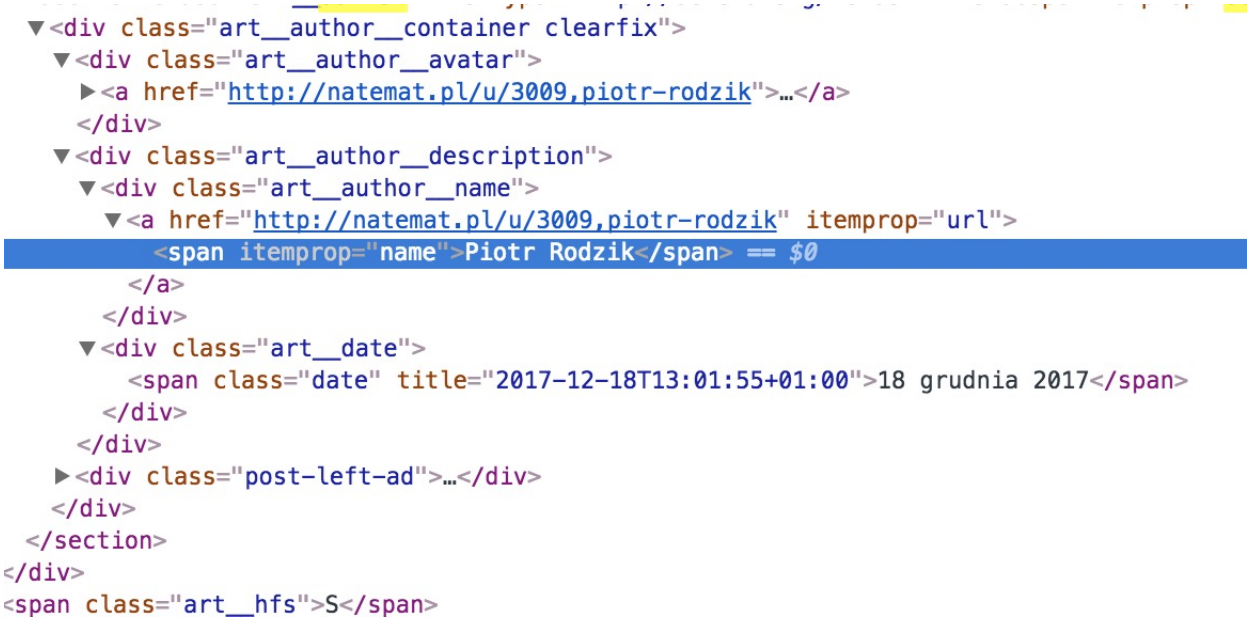
```
private String getDate(Document doc) {
    String postmeta = doc.select(".dolna-ramka").select("span").text();
    Pattern pattern = Pattern.compile("(.+) w kategorii");
    Matcher matcher = pattern.matcher(postmeta);
    if (matcher.find()) {
        return matcher.group(1);
    }
    return null;
}
```

W przypadku strony http://niebezpiecznik.pl autor jest podany wprost w tagu *a* z ustawionym atrybutem *rel=author*.

```
private String getAuthor(Document doc) {
    return doc.select("a[rel=author]").text();
}
```

2.2 Ekstrakcja autora

W kodzie HTML poniżej znajduje się informacja o imieniu i nazwisku autora artykułu ze strony natemat.pl.

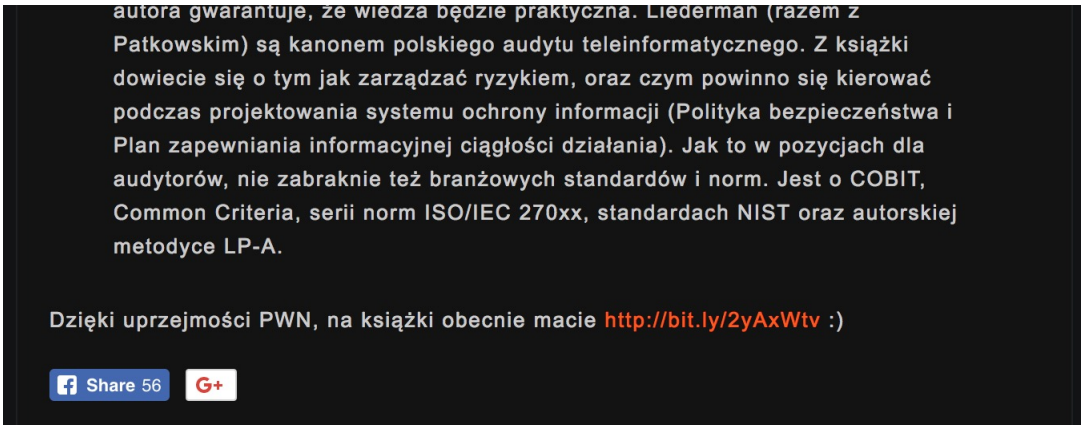


Poniższy kod z pobranego dokumentu strony wyodrębnia imię i nazwisko autora.

```
private String getAuthor(Document doc) {
                Element authorElement = doc.select(".art__author__name").first();
                String author = authorElement.text().trim();
                if (author.contains("Partnerem")) {
                        return null;
                }
                authorElement.remove();
                return author;
        }
```

2.3 Ekstrakcja adresów URL

Na poniższym rysunku znajduje się przykład adresu URL, który zostanie usunięty.



Poniżej znajduje się framentu kodu źródłowego strony.

Funkcja *removeLinks* przyjmuje jako parametr tekst z którego będa usuwane linki URL.

```
private String removeLinks(String text) {
    if(text == null){
        throw new IllegalArgumentException("Article body cannot be null");
    }
    return text.toLowerCase().replace("http.?://\\S+", "").trim();
}
```

Zmienna *String page* zawiera cały tekst artykułu z którego zostaną usunięte wszystkie adresy url, które zaczynają się od *http://* oraz. *https://*.

Wszystkie wykonywane operacje na pobranych tekstach to:

- Usuwanie linków
- Uswanie cytatów
- Usuwanie kodów źródłowych
- Usuwanie linków do obrazków

3 Dodatek do wniosków

Lista stron z któych zostały pobrane informacje:

- https://niebezpiecznik.pl - ponad 1 tysiąc artykułów
- https://zaufanatrzeciastrona.pl ponad 2 tysiące artykułów
- https://natemat.pl - ponad 16 tysięcy artykułów
- https://www.purepc.pl ponad 1 tysiąc artykułów