Utworzenie uniwersalnej, otwartej, repozytoryjnej platformy hostingowej i komunikacyjnej dla sieciowych zasobów wiedzy dla nauki, edukacji i otwartego społeczeństwa wiedzy

Raport 25.13 Demonstracyjna wersja programu wyszukującego nazwy własne w tekście.

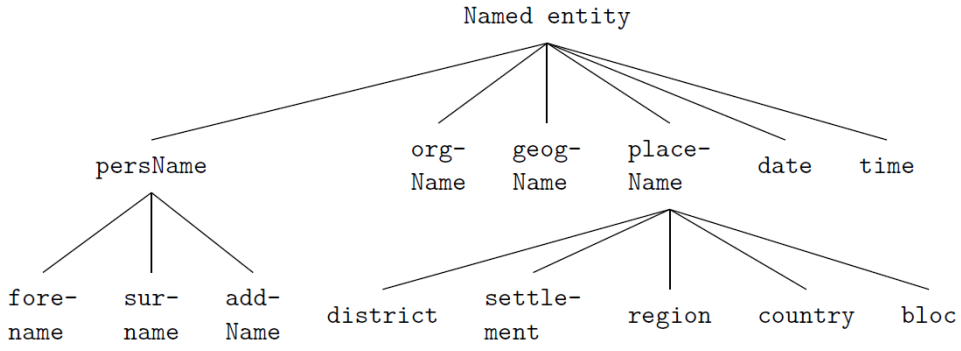
Michał Lenart, Dominika Rogozińska, Agnieszka Mykowiecka

1. Specyfikacja zadania

Celem raportowanego zadania była demonstracyjna wersja funkcjonalności pozwalającej na rozpoznawanie w tekście nazw własnych (osób, nazw geograficznych i nazw organizacji). Funkcjonalność taka może być przydatna przy ustaleniu jakich osób czy rejonów dotyczą konkretne teksty bądź przy wyszukiwaniu wszystkich dokumentów, które dotyczą wybranych osób. w wersji demonstracyjnej narzędzie funkcjonujące jako service internetowy zaznacza w podanym tekście nazwy określonych typów.

2. Narzędzie Nerf

Nerf (http://zil.ipipan.waw.pl/Nerf) jest narzędziem statystycznym służącym do oznaczania nazw własnych w tekście. Nerf rozpoznaje jednostki nazewnicze, korzystając z wytrenowanego wcześniej modelu opartego na algorytmie Linear Chain Condiitonal Random Fields. Dane uczące użyte do wytrenowania dostarczonego modelu pochodzą z Narodowego Korpusu Języka Polskiego (nkjp.pl). Typy jednostek nazewniczych są reprezentowane w ontologii, więc nazwa własna może mieć zarówno typ, jak i podtypy. Hierarchia zdefiniowanych typów nazw własnych przedstawiona jest na poniższym rysunku (za *Narodowy korpus języka polskiego*, A. Przepiórkowski, Mirosław Bańko, Rafał L. Górski i Barbara Lewandowska-Tomaszczyk (red.) Wydawnictwo Naukowe PWN):



Przykładowy wynik działania narzędzia Nerf dla zdania “Ala Kotowska ma kota.”, w którym występuję jedna nazwa osoby składająca się z imienia i nazwiska to:

<persName><persName.forename>Ala</persName.forename>

<persName.surname>Kotowska</persName.surname></persName> ma kota.

2.1 Udostępnienie funkcjonalności na potrzeby platformy Synat

Narzędzie Nerf zaimplementowane jest w języku Haskell podczas gdy większość oprogramowania platformy Infona oparta jest o Jawę. Początkowo planowane było udostępnienie funkcjonalności rozpoznawania nazw własnych poprzez stworzenie odpowiedniego wrappera. Z pewnością najbardziej wydajne byłoby użycie natywnej biblioteki JNI. Niestety podczas jej tworzenia pojawiły się techniczne problemy, których pokonanie okazało się niemożliwe w dostępnym czasie. W szczególności stwierdzono:

- Niewystarczające wsparcie dla tego typu zadania – biblioteki w Haskellu są domyślnie zainstalowane w wersjach uniemożliwiających dynamiczne linkowanie (wymagane przez JNI). W szczególności wykryto problemy programu cabal (The Haskell Common Architecture for Building Applications and Libraries) w rozwiązywaniu zależności.
- Kompilator w wersji dołączonej do oficjalnej dystrybucji Haskella zawiera błąd praktycznie uniemożliwiający linkowanie (http://stackoverflow.com/questions/23800358/calling-haskell-from-java-dynamic-linking-error-relocation, https://ghc.haskell.org/trac/ghc/ticket/8696)

W celu zapewnienia wsparcia dla rozpoznawania nazw własnych z poziomu Javy został opracowany moduł uruchamiający program Nerf jako podproces i pobierający wyniki ze standardowego wyjścia.

2.1.1 Opis formatu wyjściowego (JSON)

Dane wyjściowe programu są zapisywane w formacie JSON, służącym do reprezentacji danych w formacie tekstowym (typ String). Obiekt JSON zawiera listę obiektów reprezentujących akapity. Obiekt odpowiadający akapitowi składa się z tekstu akapitu oraz listy występujących w nim jednostek nazewniczych. Obiekt odpowiadający jednostce nazewniczej zawiera indeksy początku i końca (indeksowanie kolejnych znaków tekstu) oraz typ i podtyp jednostki nazewniczej. Przykład tekstu w formacie NerfJSON:

```
[
{
           "text": "Ala Kowalska ma kota",
           "namedEntities": [
           {
               "type": "persName",
               "subtype": "firstName",
               "startIdx": 0,
               "endIdx": 3
           },
           {
               "type": "persName",
               "subtype": "surname",
               "startIdx": 4,
               "endIdx": 12
           },
           {
               "type": "persName",
               "startIdx": 0,
               "endIdx": 12
           }
       ]
    }
]
```

2.1.2 Użycie

Źródła są dostępne w postaci projektu mavenowego o nazwie synat-nerf. W celu zbudowania i projektu należy mieć zainstalowane:

- Java (wersja 7 lub wyższa)
- Maven (wersja 2 lub wyższa)
- Apache Tomcat (potrzebne do serwisu online, testowane na wersji 7.0.32)
- System: Linux (testowane na Ubuntu 13.10)

Podstawowe klasy

- NerfServerInterface - podstawowa klasa, która umożliwia rozpoznawanie nazw własnych. Wczytuje model tylko raz, w czasie wywoływania konstruktora. Posiada dwie metody:

- String execNerfClientToJSON(String text) - wykonaj rozpoznawanie nazw własnych i daj wynik jako JSON
- destroy() - zakończ działanie podprocesu Nerf
- NerfException - wyjątek rzucany wtedy, gdy inicjalizacja klasy NerfServerInterface lub rozpoznawanie nazw własnych się nie powiedzie

3. Webserwis opakowujący Nerf

Poza udostępnieniem modułu bezpośrednio wewnątrz demonstracyjnej wersji oprogorgramowania platformy Infona, na serwerze IPI PAN uruchomiona została zewnętrzna wersja serwisu internetowego z funkcjnalnością rozpoznawania nazw własnych w tekście.

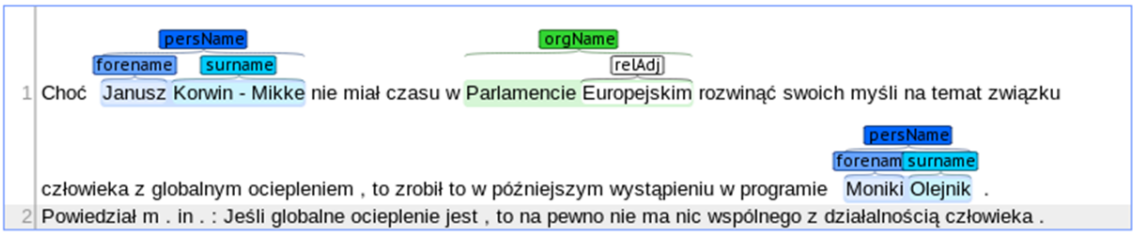
3.1 Użycie poprzez protokół HTTP

Poniższe polecenie umożliwia wywołanie serwisu za pomocą konsoli i uzyskanie wyniku w formacie JSON:

curl -X POST -d “Ala ma kota.” -H “text/plain” http://glass.ipipan.waw.pl:8087/synat-aligner/rest/nerf/annotate

3.2 Użycie przez przeglądarkę

Aby skorzytsać z srwisu wystarczy wejść na adres glass.ipipan.waw.pl:8087/synat-aligner/nerf. Po wybraniu pliku z tekstem zostanie wykonane rozpoznanie nazw własnych, które zostaną pokazane na ekranie w formie graficznej (jak na poniższym rysunku przedstawiającym obraz ekranu).



3.3 Instalacja

Instalacja serwisu odbywa się (tak jak w przypadku innych aplikacji Java EE) przez skopiowanie pliku .war do katalogu webapps danej instalacji serwera Tomcat.