POLITECHNIKA POZNAŃSKA
WYDZIAŁ ELEKTRYCZNY, INFORMATYKA
SEMESTR VII, GRUPA TI-L1

Task 2: Broader Named Entity Identification and Linking

Autorzy:
Joanna Chojnacka
Anna Zdrojewska
Konrad Kurcaba

Prowadzący:
dr Jarosław Bąk

9 stycznia 2019

1. Charakterystyka ogólna

Niniejsza dokumentacja zawiera wymagania związane ze stworzeniem aplikacji komputerowej do zadania z konkursu OKE2018 Challenge: *Broader Named Entity Identification and Linking*. Dokument zawiera informacje dotyczące estetyki oraz funkcjonalności programu. Całość została podzielona na kilka rozdziałów. W tym została opisana idea projektu.

1.1. Wybór tematu

Naszym zadaniem była identyfikacja rozszerzonej listy typów podmiotów, została ona udostępniona przez organizatora konkursu. Oto wymagane klasy i podklasy:

Klasy	**Podklasy**
Activity	Game, Sport
Agent	Employer, Organisation, Person
Award	Decoration, NobelPrize
Disease	
EthnicGroup	
Event	Competition, PersonalEvent
Language	ProgrammingLanguage
MeanOfTransportation	Aircraft, Train
PersonFunction	PoliticalFunction, Profession
Place	
Species	Animal
Work	Artwork

Zadanie obejmuje identyfikację elementów w zdaniach i ujednoznacznienie zidentyfikowanych podmiotów do wiedzy z bazy DBpedii.

Naszym celem była poprawa umiejętności programistycznych oraz rozwój w zakresie sieci semantycznych. Zagadnienia poznane podczas wykładu mogliśmy zastosować w praktyce i przeanalizować działanie mechanizmów.

1.2. Podział prac

- Joanna Chojnacka:
  - Wykorzystanie framework'u JavaFX do utworzenia interfejsu użytkownika,
  - Testowanie aplikacji,
  - Praca nad dokumentacją.
- Anna Zdrojewska:
  - Usunięcie stopwords z zapytań oraz odczytywanie danych wejściowych przy pomocy NIF Library,
  - Implementacja funkcjonalności,
  - Praca nad dokumentacją.
- Konrad Kurcaba:
  - Wykorzystanie Apache Jena do zapytań RDF,
  - Implementacja funkcjonalności,
  - Praca nad dokumentacją.

2. Użyte narzędzia i środowiska

2.1. Java SE 8

Java to obiektowy język programowania cieszący się dużą popularnością wśród programistów. Wybraliśmy go, ponieważ znamy i lubimy jego strukturę. Java zapewnia również wysoką wydajność, co przy tym projekcie miało duże znaczenie, ze względu na parsowanie zapytań, jak ich wysyłanie - pozwoliło to poprawić efektywność działań naszego programu.

2.2. JavaFX

JavaFX pozwala na tworzenie aplikacji z nowoczesnym wyglądem, zachowując wydajność i większą czytelność kodu. Modyfikacja wyglądu jest możliwa za pomocą stylów CSS, co pozwala na szybsze i łatwiejsze zmiany poszczególnych cech. Podstawowym narzędziem do tworzenia aplikacji w JavaFX jest Scene Builder, którego wykorzystaliśmy, aby zoptymalizować pracę. Za jego pomocą został wygenerowany gotowy kod XML w czytelnej formie.

2.3. NIF Library

Format NIF (ang. NLP Interchange Format) jest to format bazujący na RDF/OWL. NIF Library to biblioteka umożliwiająca odczytywanie zapytań w tym formacie oraz RDF-xml,

JSON-LD, Turtle, NTriples. Okazała się niezbędna dla projektu ze względu na łatwość implementacji.

2.4. Apache Jena

Darmowy framework do budowy semantycznych sieci bazujący na języku Java. Jego struktura składa się z wielu różnych interfejsów API, które współpracują ze sobą, aby przetwarzać dane RDF. Jena umożliwia również tworzenie zapytań SPARQL oraz budowanie ontologii.

2.5. Maven

Narzędzie do zautomatyzowanego budowania projektów. Umożliwia uzyskiwanie dostępu do zewnętrznych zasobów takich jak Apache Jena czy NIF Library.

2.6. Eclipse IDE Photon

Eclipse to bezpłatne środowisko programistyczne przeznaczone dla języka Java. Pozwala na sprawne zarządzanie projektem i zapewnia dużą czytelność kodu.

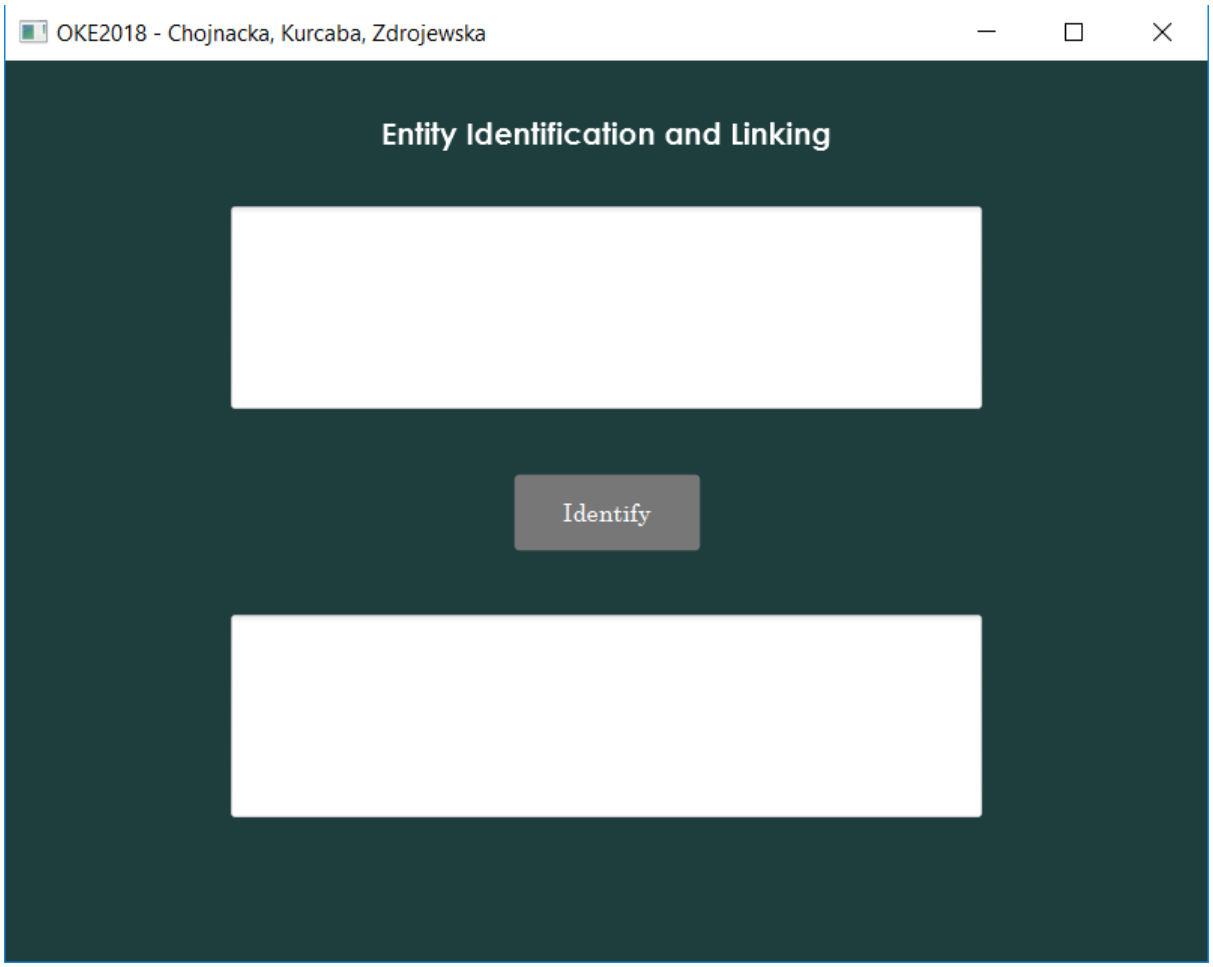
3. Realizacja zadania

3.1. Opis działania

W pierwszym kroku następuje odczyt danych z pliku w formacie NIF. Dalej zostają one przekazane do parsera, który pobiera interesujący nas tekst do analizy. Dla zoptymalizowania programu dodaliśmy implementację umożliwiającą usunięcie stopwords. Wykorzystaliśmy w tym celu listę angielskich stopwords dostępnych pod adresem (https://gist.github.com/sebleier/554280 [dostęp 8.01.2019]). Następnie przetworzony tekst jest przekazywany do identyfikacji. Wykorzystywany jest tu mechanizm polegający na wyodrębnieniu wycinka składającego się z trzech słów. Sprawdzana jest każda możliwa kombinacja wyrazów. Tak powstałe zapytanie wysyłane jest do DBpedii. Jeżeli zostanie uzyskany odpowiadający typ to zwrócony rekord zapisywany jest do mapy. Dodatkowo znaleziony element jest wycinany z tekstu. Dalej następuje kolejna iteracja po tekście wejściowym, ale tym razem wyodrębniany wycinek jest zmniejszany o jeden. Całość jest powtarzana, aż do iteracji po jednym słowie. W końcowym działaniu mapa jest przekształcana na tekst wyjściowy, działanie wątku się kończy i zostaje pobrany string wynikowy wyświetlony w interfejsie.

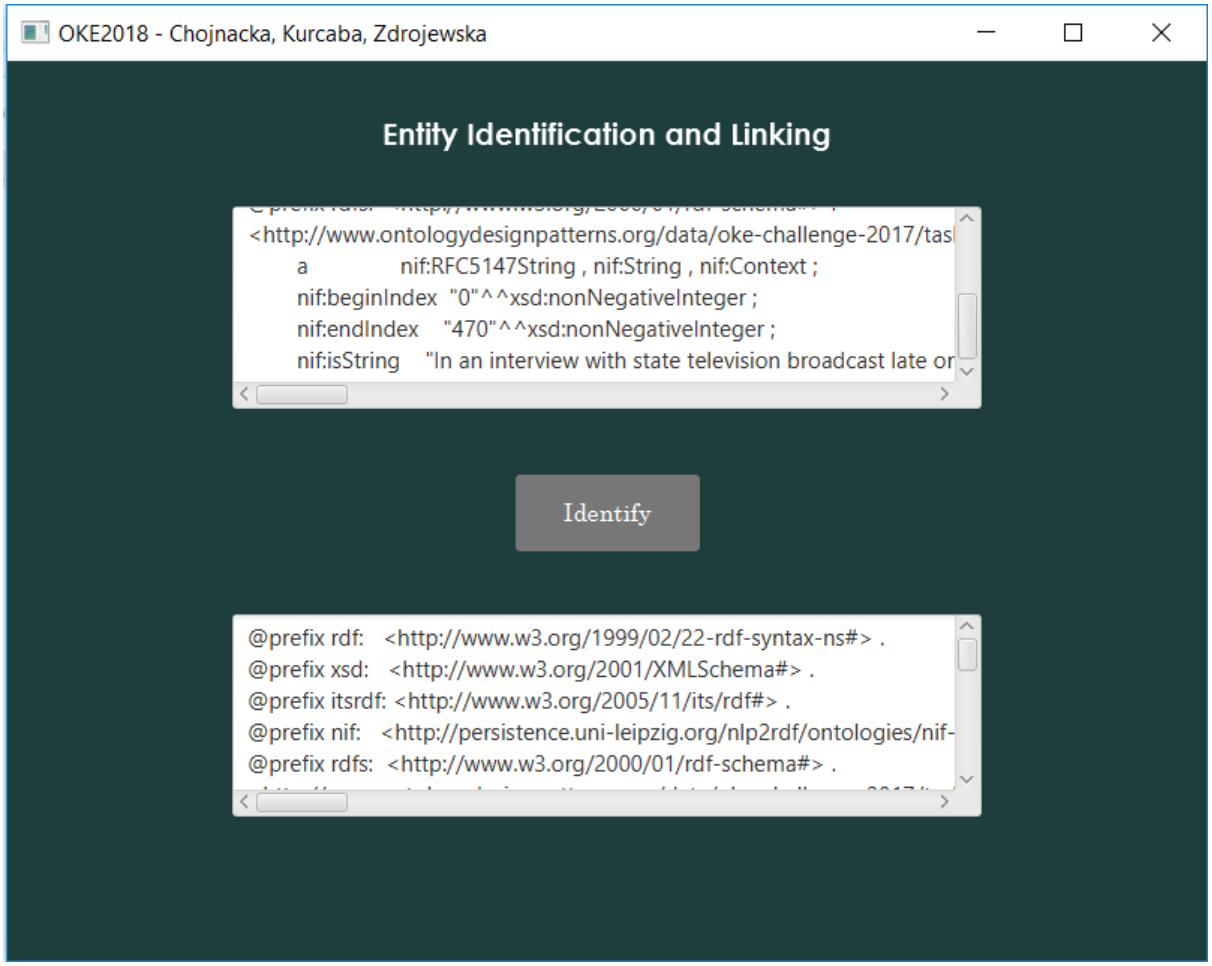
3.2. Instrukcja obsługi

Aplikacja zawiera dwa pola tekstowe – wejściowe oraz wyjściowe. Pomiędzy umieszczony jest przycisk, który pełni rolę wykonywania działań w tle.



***Ilustracja nr 1** – Interfejs aplikacji.*

Aby rozpocząć działanie aplikacji wystarczy nacisnąć przycisk pomiędzy oknami.



***Ilustracja nr 2** – Działanie.*

W polu wejściowym zostanie umieszczone zapytanie, natomiast w polu wyjściowym otrzymamy rezultat.

3.3 Testy

Zapytanie 1
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> . @prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> . @prefix dbpedia: <http://dbpedia.org/resource/> . @prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> . <http://example.com/example-jb1#char=0,55> a nif:RFC5147String , nif:String , nif:Context ; nif:beginIndex "0"^^xsd:nonNegativeInteger ; nif:endIndex "55"^^xsd:nonNegativeInteger ; nif:isString "**Michael Jackson and Donald Trump never met in New York.**"^^xsd:string .

Rezultat

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/example-jb1#char=0,55>
      a               nif:RFC5147String , nif:String , nif:Context ;
      nif:beginIndex      "0"^^xsd:nonNegativeInteger ;
      nif:endIndex        "55"^^xsd:nonNegativeInteger ;
      nif:isString        "Michael Jackson and Donald Trump never met in New
York."^^xsd:string .
<http://example.com/example-jb1#char=20,32>
      a               nif:RFC5147String , nif:String , nif:Context ;
      nif:anchorOf        "Donald_Trump"@en ;
      nif:beginIndex      "20"^^xsd:nonNegativeInteger ;
      nif:endIndex        "32"^^xsd:nonNegativeInteger ;
      itsrdf:taIdentRef   dbpedia:Donald_Trump .
<http://example.com/example-jb1#char=50,54>
      a               nif:RFC5147String , nif:String , nif:Context ;
      nif:anchorOf        "York"@en ;
      nif:beginIndex      "50"^^xsd:nonNegativeInteger ;
      nif:endIndex        "54"^^xsd:nonNegativeInteger ;
      itsrdf:taIdentRef   dbpedia:York .
<http://example.com/example-jb1#char=46,54>
      a               nif:RFC5147String , nif:String , nif:Context ;
      nif:anchorOf        "New_York"@en ;
      nif:beginIndex      "46"^^xsd:nonNegativeInteger ;
      nif:endIndex        "54"^^xsd:nonNegativeInteger ;
      itsrdf:taIdentRef   dbpedia:New_York .
<http://example.com/example-jb1#char=0,15>
      a               nif:RFC5147String , nif:String , nif:Context ;
      nif:anchorOf        "Michael_Jackson"@en ;
      nif:beginIndex      "0"^^xsd:nonNegativeInteger ;
      nif:endIndex        "15"^^xsd:nonNegativeInteger ;
      itsrdf:taIdentRef   dbpedia:Michael_Jackson .
```

Zapytanie 2

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#>.
<http://example.com/example-jb2#char=0,46>
      a               nif:RFC5147String , nif:String , nif:Context ;
      nif:beginIndex      "0"^^xsd:nonNegativeInteger ;
      nif:endIndex        "46"^^xsd:nonNegativeInteger ;
```

nif:isString "**La Toya is an older sister of Michael Jackson.**"^^xsd:string .

Rezultat

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/example-jb2#char=0,46>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:beginIndex      "0"^^xsd:nonNegativeInteger ;
    nif:endIndex        "46"^^xsd:nonNegativeInteger ;
    nif:isString        "La Toya is an older sister of Michael Jackson."^^xsd:string .
<http://example.com/example-jb2#char=30,45>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:anchorOf        "Michael_Jackson"@en ;
    nif:beginIndex      "30"^^xsd:nonNegativeInteger ;
    nif:endIndex        "45"^^xsd:nonNegativeInteger ;
    itsrdf:taIdentRef   dbpedia:Michael_Jackson .
```

Zapytanie 3

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#>.
<http://example.com/example-jb3#char=0,118>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:beginIndex      "0"^^xsd:nonNegativeInteger ;
    nif:endIndex        "118"^^xsd:nonNegativeInteger ;
    nif:isString        "At Poznan University of Technology a lot of nice people are
working, especially Adam Malysz who is teaching ski jumps."^^xsd:string .
```

Rezultat

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/example-jb3#char=0,118>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:beginIndex      "0"^^xsd:nonNegativeInteger ;
    nif:endIndex        "118"^^xsd:nonNegativeInteger ;
    nif:isString        "At Poznan University of Technology a lot of nice people are
working, especially Adam Małysz who is teaching ski jumps."^^xsd:string .
<http://example.com/example-jb3#char=44,55>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:anchorOf        "Nice_People"@en ;
```

nif:beginIndex "44"^^xsd:nonNegativeInteger ;
nif:endIndex "55"^^xsd:nonNegativeInteger ;
itsrdf:taIdentRef **dbpedia:Nice_People** .
<http://example.com/example-jb3#char=80,91>
a nif:RFC5147String , nif:String , nif:Context ;
nif:anchorOf "Adam_Małysz"@en ;
nif:beginIndex "80"^^xsd:nonNegativeInteger ;
nif:endIndex "91"^^xsd:nonNegativeInteger ;
itsrdf:taIdentRef **dbpedia:Adam_Małysz** .

Zapytanie 4

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#>.
<http://example.com/example-jb4#char=0,146>
a nif:RFC5147String , nif:String , nif:Context ;
nif:beginIndex "0"^^xsd:nonNegativeInteger ;
nif:endIndex "146"^^xsd:nonNegativeInteger ;
nif:isString "**Elvis married Priscilla in 1967 at the Aladdin Hotel in Las Vegas, Nevada. They had a daughter called Lisa and she was born in Memphis, Tennessee.**"^^xsd:string .

Rezultat

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/example-jb4#char=0,146>
a nif:RFC5147String , nif:String , nif:Context ;
nif:beginIndex "0"^^xsd:nonNegativeInteger ;
nif:endIndex "146"^^xsd:nonNegativeInteger ;
nif:isString "**Elvis married Priscilla in 1967 at the Aladdin Hotel in Las Vegas, Nevada. They had a daughter called Lisa and she was born in Memphis, Tennessee.**"^^xsd:string .

<http://example.com/example-jb4#char=136,145>
a nif:RFC5147String , nif:String , nif:Context ;
nif:anchorOf "Tennessee"@en ;
nif:beginIndex "136"^^xsd:nonNegativeInteger ;
nif:endIndex "145"^^xsd:nonNegativeInteger ;
itsrdf:taIdentRef **dbpedia:Tennessee** .
<http://example.com/example-jb4#char=56,65>
a nif:RFC5147String , nif:String , nif:Context ;
nif:anchorOf "Las_Vegas"@en ;

nif:beginIndex "56"^^xsd:nonNegativeInteger ;
nif:endIndex "65"^^xsd:nonNegativeInteger ;
itsrdf:taIdentRef **dbpedia:Las_Vegas** .
<http://example.com/example-jb4#char=47,52>
a nif:RFC5147String , nif:String , nif:Context ;
nif:anchorOf "Hotel"@en ;
nif:beginIndex "47"^^xsd:nonNegativeInteger ;
nif:endIndex "52"^^xsd:nonNegativeInteger ;
itsrdf:taIdentRef **dbpedia:Hotel** .
<http://example.com/example-jb4#char=67,73>
a nif:RFC5147String , nif:String , nif:Context ;
nif:anchorOf "Nevada"@en ;
nif:beginIndex "67"^^xsd:nonNegativeInteger ;
nif:endIndex "73"^^xsd:nonNegativeInteger ;
itsrdf:taIdentRef **dbpedia:Nevada** .

Zapytanie 5

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#>.
<http://example.com/example-jb5#char=0,203>
a nif:RFC5147String , nif:String , nif:Context ;
nif:beginIndex "0"^^xsd:nonNegativeInteger ;
nif:endIndex "203"^^xsd:nonNegativeInteger ;
nif:isString "**The Googleplex is the corporate headquarters complex of Google and its parent company Alphabet Inc.. It is located at 1600 Amphitheatre Parkway in Mountain View, California, United States, near San Jose.**"^^xsd:string .

Rezultat

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/example-jb5#char=0,203>
a nif:RFC5147String , nif:String , nif:Context ;
nif:beginIndex "0"^^xsd:nonNegativeInteger ;
nif:endIndex "203"^^xsd:nonNegativeInteger ;
nif:isString "**The Googleplex is the corporate headquarters complex of Google and its parent company Alphabet Inc.. It is located at 1600 Amphitheatre Parkway in Mountain View, California, United States, near San Jose.**"^^xsd:string .
<http://example.com/example-jb5#char=4,10>
a nif:RFC5147String , nif:String , nif:Context ;
nif:anchorOf "Google"@en ;
nif:beginIndex "4"^^xsd:nonNegativeInteger ;

```
    nif:endIndex        "10"^^xsd:nonNegativeInteger ;
    itsrdf:taIdentRef   dbpedia:Google .
<http://example.com/example-jb5#char=78,85>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:anchorOf        "Company"@en ;
    nif:beginIndex      "78"^^xsd:nonNegativeInteger ;
    nif:endIndex        "85"^^xsd:nonNegativeInteger ;
    itsrdf:taIdentRef   dbpedia:Company .
<http://example.com/example-jb5#char=71,77>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:anchorOf        "Parent"@en ;
    nif:beginIndex      "71"^^xsd:nonNegativeInteger ;
    nif:endIndex        "77"^^xsd:nonNegativeInteger ;
    itsrdf:taIdentRef   dbpedia:Parent .
<http://example.com/example-jb5#char=162,172>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:anchorOf        "California"@en ;
    nif:beginIndex      "162"^^xsd:nonNegativeInteger ;
    nif:endIndex        "172"^^xsd:nonNegativeInteger ;
    itsrdf:taIdentRef   dbpedia:California .
<http://example.com/example-jb5#char=174,187>
    a               nif:RFC5147String , nif:String , nif:Context ;
    nif:anchorOf        "United_States"@en ;
    nif:beginIndex      "174"^^xsd:nonNegativeInteger ;
    nif:endIndex        "187"^^xsd:nonNegativeInteger ;
    itsrdf:taIdentRef   dbpedia:United_States .
```

4. Podsumowanie pracy

Wykonanie zadania z konkursu OKE2018 Challenge polegającego na stworzeniu aplikacji umożliwiającej identyfikację podmiotów dało nam okazję do rozwinięcia naszych umiejętności programistycznych, jak i poszerzyło wiedzę teoretyczną. Dotknęliśmy zagadnień związanych z pozyskiwania wiedzy z tekstu dla sieci semantycznej, sprawdzania wydajności systemów pozyskiwania wiedzy, identyfikacji podmiotów, ujednoznacznienia poprzez powiązanie z bazą wiedzy, ekstrakcji relacji oraz wiedzy. Poprawiliśmy sprawność efektywnego i zoptymalizowanego programowania, odświeżyliśmy informację na temat różnych formatów plików, takich jak **.rdf*, **.xml*. Wykonaliśmy dużo pracy przy implementacji w języku Java, tworząc główną aplikację, jak i przy mniejszych częściach np. przygotowaniu tekstu do analizy, gdzie konieczna była implementacja usuwania stopwords.