Narzędzia do rozpoznawania

i wizualizacji struktur informacyjnych

30.05.2016

dr inż. Michał Marcińczuk

michal.marcinczuk@pwr.edu.pl

1. Wstęp

Niniejszy dokument opisuje następujące narzędzia:

1. **g419-spatial** — narzędzie do przetwarzania tekstów w celu rozpoznania i oznaczenia struktur informacyjnych wyrażeń przestrzennych. Docelowo narzędzie zostanie zintegrowane z systemem *nlprest*, który obsługuje zdalne przetwarzanie tekstów poprzez wywołania REST. Narzędzie zostało zaimplementowane w taki sposób, aby możliwa była integracja z systemem *nlprest*.

2. **ieviewer** — webowe narzędzie do wizualizacji struktur informacyjnych wyrażeń przestrzennych. Jego zadaniem jest przetworzenie

2. Narzędzie g419-spatial

2.1. Realizacja

Narzędzie *g419-spatial* zostało wykonane jako dwa niezależne moduły systemu Liner2: *g419-spatial-core* i *g419-spatial-cli*. Pierwszy moduł, tj. *g419-spatial-core* zawiera niezbędne struktury danych, klasy pomocnicze oraz klasę SpatialRelationRecognizer realizującą zadanie rozpoznawania wyrażeń przestrzennych (zob. 2.3). Drugi moduł, tj. *g419-spatial-cli*, jest modułem pomocniczym pozwalającym na uruchomienie narzędzia bezpośrednio z linii poleceń (zob. 2.2).

2.2. Uruchamianie narzędzia z linii komend

Do uruchomienia narzędzia w konsoli służy skrypt `spatial-cli`.

```
./spatial-cli pipe
```

Po uruchomieniu skryptu bez parametrów zostanie wyświetlony opis dostępnych trybów:

```
*----------------------------------------------------------------------------*
* Tools for spatial expression recognition and related
*
* Authors: Michał Marcińczuk (2015)
* Contact: michal.marcinczuk@pwr.wroc.pl
*
*          G4.19 Research Group, Wrocław University of Technology
*
*----------------------------------------------------------------------------*


[Option error] Missing required options: M, w

usage: ./liner2-cli pipe [options]
 -f,--input_file <filename>      read input from file
 -i,--input_format <format>      input format [iob, ccl, plain, plain:maca,
plain:wcrft, tei,
                                 batch:{format}]
 -M,--malt <filename>            path to maltparser model
 -o,--output_format <filename>   output format [iob, ccl, arff, tokens,
tuples, tei,
                                 batch:{format}]
 -t,--output_file <filename>     path to an output file
 -v,--verbose                    print help
 -w,--wordnet <path>             path to a folder with a wordnet in Princeton
format
```

Przykład wywołania narzędzia w celu przetworzenia dokumentu:

```
./spatial-cli pipe -i tei -o json-frames -f sample_input_tei
```

2.3. Przetworzenie dokumentu z poziomu kodu

Poniższy kod prezentuje wykorzystanie klasy SpatialRelationRecognizer w celu przetworzenia dokumentu.

```
package g419.spatial.action;

import org.apache.commons.cli.CommandLine;
import org.apache.commons.cli.DefaultParser;
import g419.corpus.io.reader.AbstractDocumentReader;
import g419.corpus.io.reader.ReaderFactory;
import g419.corpus.io.writer.AbstractDocumentWriter;
import g419.corpus.io.writer.WriterFactory;
import g419.corpus.structure.Document;
```

```
import g419.lib.cli.Action;
import g419.lib.cli.CommonOptions;
import g419.liner2.api.tools.parser.MaltParser;
import g419.spatial.tools.SpatialRelationRecognizer;
import g419.toolbox.wordnet.Wordnet3;

public class ActionPipe extends Action {

   private String inputFilename = null;
   private String inputFormat = null;
   private String outputFilename = null;
   private String outputFormat = null;
   private String maltparserModel = null;
   private String wordnetPath = null;

   public ActionPipe() {
      super("pipe");
      this.setDescription("recognize spatial expressions");
      this.options.addOption(CommonOptions.getInputFileNameOption());
      this.options.addOption(CommonOptions.getInputFileFormatOption());
      this.options.addOption(CommonOptions.getOutputFileFormatOption());
      this.options.addOption(CommonOptions.getOutputFileNameOption());
      this.options.addOption(CommonOptions.getMaltparserModelFileOption());
      this.options.addOption(CommonOptions.getWordnetOption(true));
   }

   @Override
   public void parseOptions(String[] args) throws Exception {
      CommandLine line = new DefaultParser().parse(this.options, args);
      parseDefault(line);
      this.inputFilename = line.getOptionValue(CommonOptions.OPTION_INPUT_FILE);
      this.inputFormat = line.getOptionValue(CommonOptions.OPTION_INPUT_FORMAT);
      this.outputFilename = line.getOptionValue(CommonOptions.OPTION_OUTPUT_FILE);
      this.outputFormat = line.getOptionValue(CommonOptions.OPTION_OUTPUT_FORMAT);
      this.maltparserModel = line.getOptionValue(CommonOptions.OPTION_MALT);
      this.wordnetPath = line.getOptionValue(CommonOptions.OPTION_WORDNET);
   }

   @Override
   public void run() throws Exception {
      Wordnet3 wordnet = new Wordnet3(this.wordnetPath);
      MaltParser malt = new MaltParser(this.maltparserModel);
      SpatialRelationRecognizer recognizer = new SpatialRelationRecognizer(malt, wordnet);

      AbstractDocumentReader reader = ReaderFactory.get().getStreamReader(this.inputFilename,
this.inputFormat);
      AbstractDocumentWriter writer = null;

      if ( this.outputFilename == null ){
         writer = WriterFactory.get().getStreamWriter(System.out, this.outputFormat);
      }
      else{
         writer = WriterFactory.get().getStreamWriter(this.outputFilename, this.outputFormat);
      }

      Document document = null;
      while ( ( document = reader.nextDocument() ) != null ){
         System.out.println("=======================================");
         System.out.println("Document: " + document.getName());
         System.out.println("=======================================");
         recognizer.recognizeInPlace(document);
         writer.writeDocument(document);
      }
      writer.close();
      reader.close();
   }
}
```

2.4. Format wejściowy TEI

Dokument wejściowy powinien zawierać następujące dane:

1. Analizę morfologiczną — dane mogą być uzyskane przy pomocy narzędzia *wcrft*[1].

2. Jednostki identyfikacyjne — dane mogą być uzyskane przy pomocy narzędzia *Liner2*[2].

3. Grupy nominalne — dane mogą być pozyskane przy pomocy narzędzia *Spejd*[3] i *gramatyki NKJP*[4].

4. Frazy składniowe — dane mogą być pozyskane przy pomocy narzędzia *Iobber*[5].

Powyższe dane mogą być zapisane w formacie TEI. Struktura tego formatu jest opisana na stronie http://nlp.ipipan.waw.pl/TEI4NKJP/.

2.5. Format wyjściowy json-frames

Rozpoznane wyrażenia przestrzenne zapisywane są dodawane do dokumentu w postaci struktur *frames*. Struktura *frame* zawiera zbiór atrybutów oraz zbiór nazwanych anotacji. Wymagane atrybuty to identyfikator (id) i typ ramy (type). Wyrażenia przestrzenne mają przypisany typ "spatial". Zbiór nazwanych anotacji zawiera: "trajector", "spatial_indicator", "landmark" i "region" (opcjonalnie). Każda anotacja jest reprezentowana jako identyfikator anotacji (id) oraz zbiór atrybutów. Dla anotacji "trajector" i "landmark" jest to atrybut "sumo", która zawiera listę konceptów SUMO[6] przypisanych do głowy anotacji.

Format *json-frames* zapisany jest w formacie JSON i zawiera następujące dane:

1 http://nlp.pwr.wroc.pl/redmine/projects/wcrft/wiki
2 https://clarin-pl.eu/dspace/handle/11321/231
3 http://zil.ipipan.waw.pl/Spejd
4 http://clip.ipipan.waw.pl/LRT?action=AttachFile&do=view&target=gramatyka_Spejd_NKJP_1.0.zip
5 http://nlp.pwr.wroc.pl/redmine/projects/iobber/wiki
6 http://www.adampease.org/OP/

- lista tokenów — każdy token reprezentowany jest jako tablica następujących elementów: identyfikator, forma ortograficzna, forma bazowa, tag morfologiczny, informacja o istnieniu spacji po tokenie.

- lista anotacji — każda anotacja opisana jest następującymi atrybutami: identyfikator, lista identyfikatorów tokenów wchodzących w skład anotacji, forma tekstowa anotacji.

- lista ram — każda rama opisana jest następującymi atrybutami: identyfikator, typ, lista nazwanych anotacji (para nazwa-anotacja).

Poniżej znajduje się przykładowy dokument w formacie *json-frames*.


```
{"frames":
  [{"slots":{
      "debug":{
        "attributes":{
          "schema":"w,we#w1",
          "pattern":"\u003cFirstNG|...|PrepNG\u003e"}},
      "trajector":{
        "attributes":{
          "sumo":"PsychologicalAttribute, CompoundSubstance, CorpuscularObject,
Artifact"},
          "id":"a28"},
      "landmark":{
        "attributes":{
          "sumo":"GroupOfPeople, City"},
          "id":"a24"},
      "region":{},
      "spatial_indicator":{"id":"a14"}},
      "id":"x",
      "type":"spatial"}],
  "annotations":[
    {"tokens"["t1"],
     "id":"a1",
     "text":"Toronto",
     "type":"Noun",
     "category":"word"}
    ,
    {"tokens":["t2"],
     "id":"a2",
     "text":"Dominion",
     "type":"Noun",
     "category":"word"}
    ,(...)],
  "tokens":[
   ["t1","Toronto","Toronto","subst:sg:nom:n","0"]
   ,["t2","Dominion","dominion","subst:sg:nom:n","0"]
   ,["t3","Centre","centre","subst:sg:nom:n","0"]
   ,["t4","Toronto","Toronto","subst:sg:nom:n","0"]
   ,(...)]
}
```

3. Narzędzie ieviewer

3.1. Opis

Narzędzie do wizualizacji zostało zaimplementowane w technologii PHP z wykorzystanie dynamicznych elementów napisanych w JavaScript. Do uruchomienia aplikacji wymagany jest serwer www Apache2, biblioteka PEAR z modułem HTTP_Session2.

3.2. Dostęp

Narzędzie ieviewer dostępne jest pod adresem http://inforex.clarin-pl.eu/ieviewer

3.3. Funkcje ieviewer

Narzędzie ievierwer udostępnia następujące funkcje:

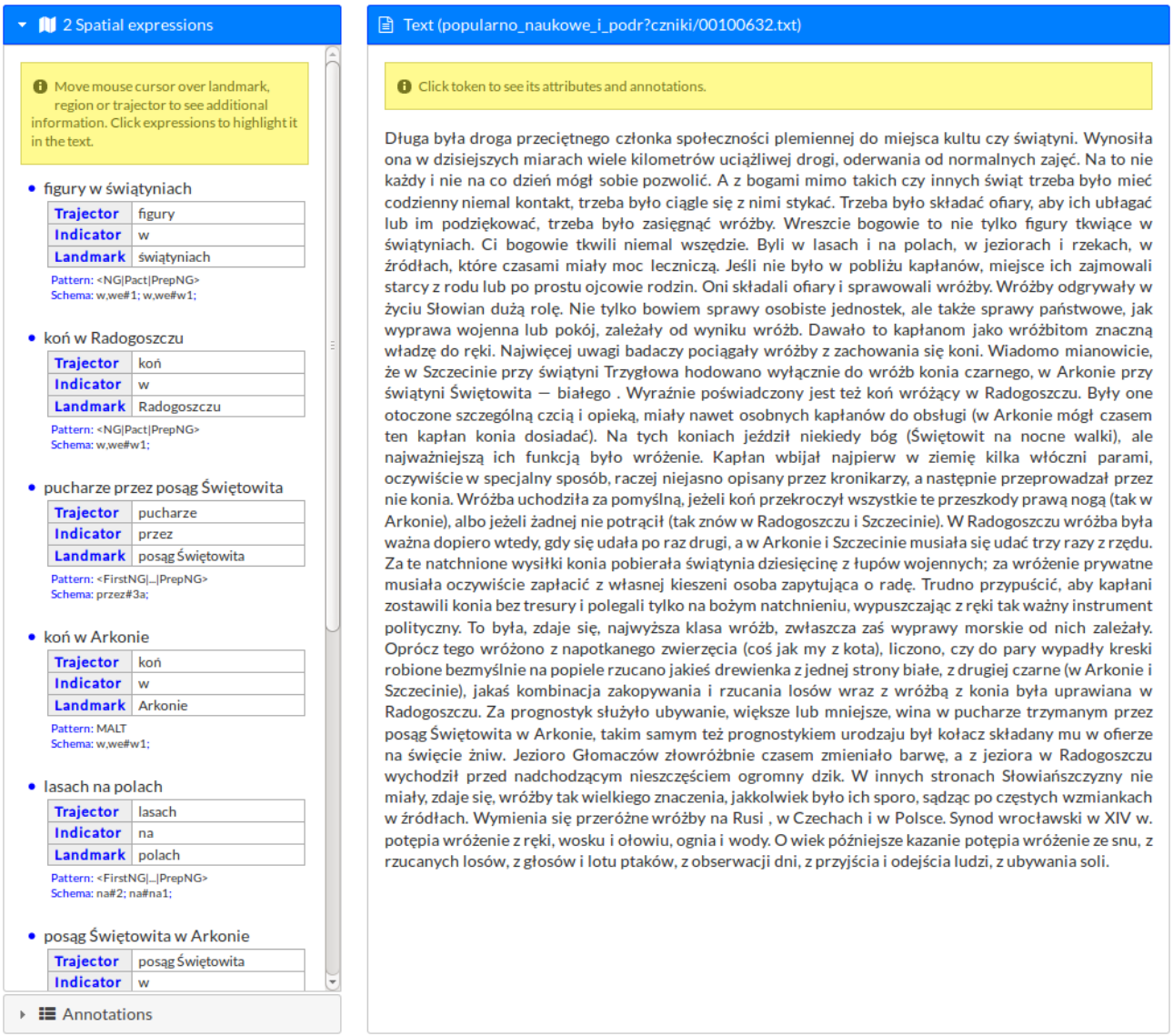
1. Widok dokumentu z możliwością wyświetlenia informacji dla każdego tokenu. Informacja o tokenie zawiera: formę ortograficzną, formę bazową, tag morfologiczny, listę anotacji zawierających danych token.

2. Lista ram z możliwością podświetlenia całej ramy i jej składowych w dokumencie.

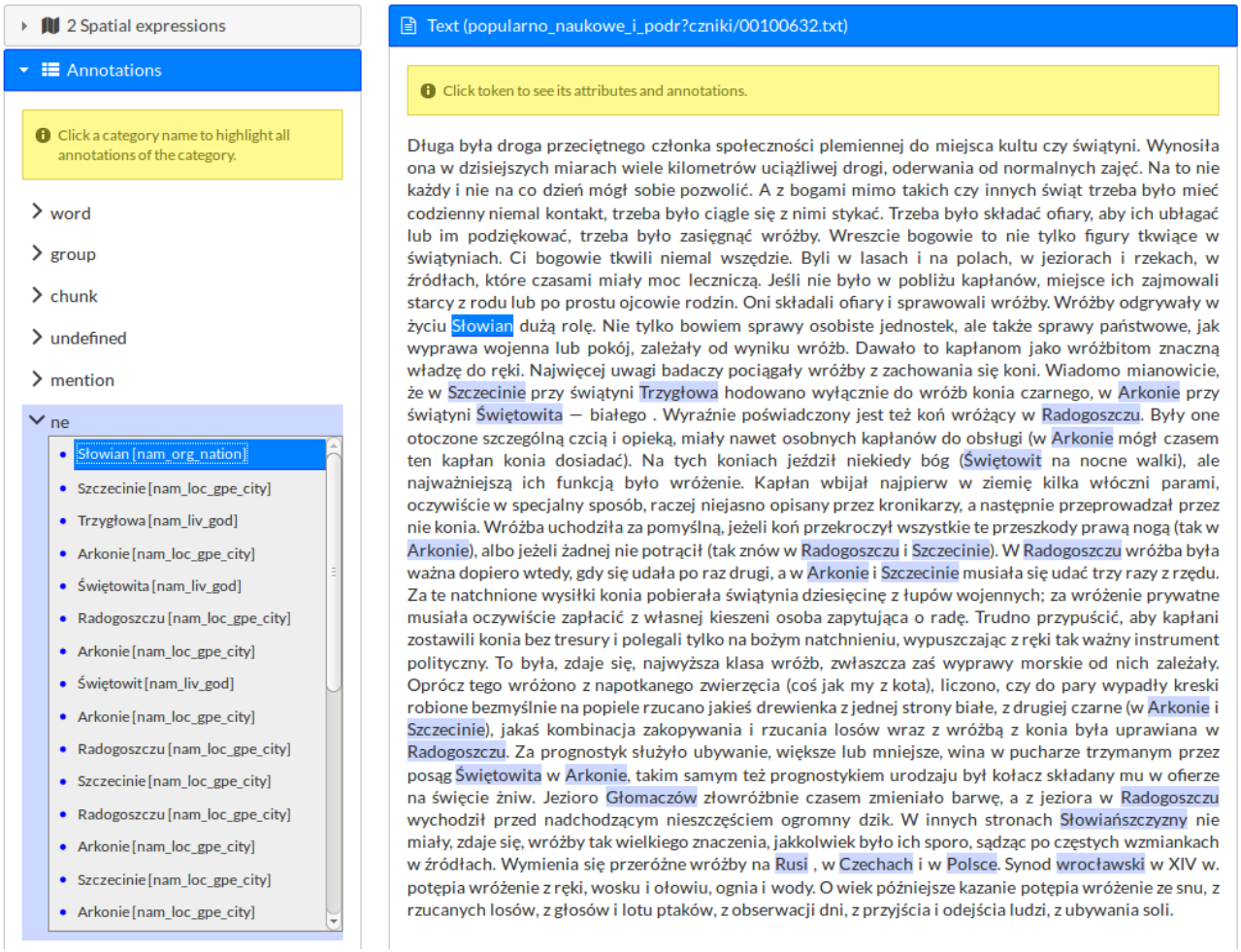
3. Lista anotacji podzielonych na kategorie (word, ne, group, chunk, mention i indefined).

3.4. Przykładowe zrzuty ekranu

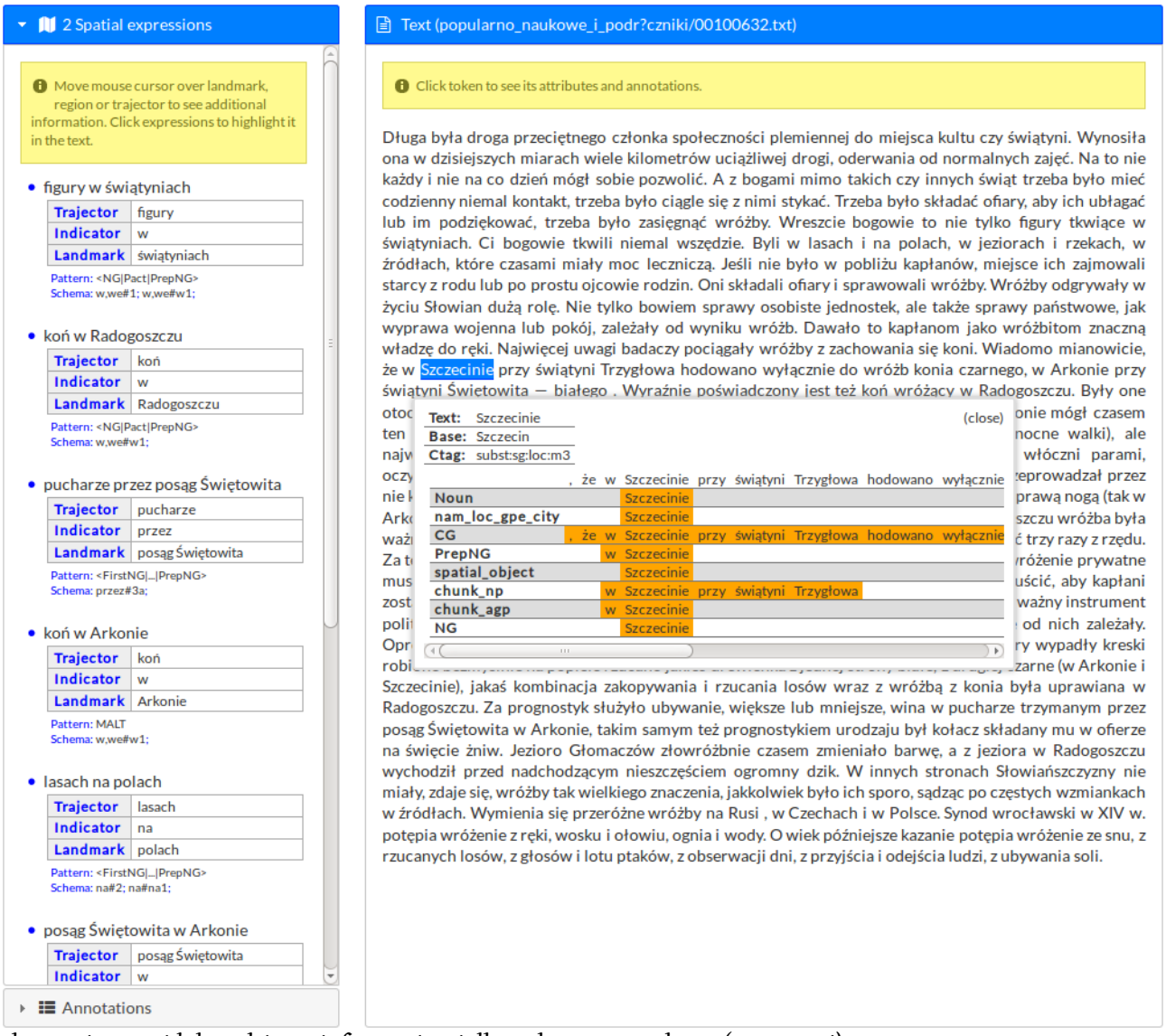
Ilustracja 1: Tooltip z opisem dopasowanego schematu semantycznego do wyrażenia przestrzennego.



Ilustracja 2: Widok listy wyrażeń przestrzennych (po lewej) i widoku dokumentu (po prawej).



Ilustracja 3: Widok listy anotacji podzielonych na kategorie (po lewej).



Ilustracja 4: Widok tooltipa z informacjami dla wskazanego tokenu (po prawej).