Rule-based Named-Entity Recognition for Polish

Jakub Piskorski
Language Technology Lab
DFKI GmbH
piskorsk@dfki.de

Abstract

Although considerable work on named-entity recognition for English and few other major languages exists, research on this topic with regard to Slavonic languages has been almost neglected. In this paper, we present an attempt towards constructing a named-entity recognition system for Polish on top of SProUT, a novel multi-lingual NLP platform, we discuss the encountered difficulties, and present preliminary evaluation results.

1 Introduction

Named-entities (NE) constitute significant part of natural language texts and are widely exploited in various NLP applications, such as Information Extraction, Text Mining, Question Answering and Machine Translation. Named-entity recognition (NER) is a well-established task in the NLP community (Appelt and Israel, 1999). While there has been a bulk of research centered around the development of NER systems for English and a few other major languages, relatively few efforts have been undertaken for fulfilling this task for Slavonic languages[1]. Initial attempts at the integration of research activities on this topic were presented at a recent IESL workshop held in conjunction with the RANLP 2003 conference. Some ongoing work on adapting the famous information extraction platform GATE (Cunningham et al., 2002) for the NER task for Bulgarian and Russian were presented in (Paskaleva et al., 2002), (Bontcheva et al., 2003), and (Khoroshevsky, 2003).

In this paper, we present a NER engine for Polish, built on top of SProUT (Shallow Text Processing with Unification and Typed Feature Structures) - a novel general purpose multi-lingual information extraction platform (Becker et al., 2002; Drożdżyński et al., 2004). Polish is a West Slavonic language and, analogously to other languages in the group, it exhibits a highly inflectional character and has a relatively free word-order (Świdziński and Saloni, 1998). Due to these specifics and general lack of linguistic resources for Polish, construction of a NER system for Polish is an intriguing and challenging task.

The rest of the paper is organized as follows. Firstly, in section 2, we introduce SProUT and its particularities. Section 3 takes an insight into setting up and fine-tuning SProUT to the processing of Polish. The NE-grammar development and its evaluation are described in section 4. Finally, we finish off with some conclusions in section 5.

2 SProUT

Analogously to the widely-known GATE system, SProUT is equipped with a set of reusable Unicode-capable online processing components for basic linguistic operations, including tokenization, sentence splitting, morphological analysis, gazetteer lookup, and reference matching. Since typed feature structures (TFS) are used as a uniform data structure for representing the input and output by each of these processing resources, they can be flexibly combined into a pipeline that produces several streams of linguistically annotated structures, which constitute an input for the shallow grammar interpreter, applied at the next stage.

The grammar formalism used in SProUT is a blend of very efficient finite-state techniques and unification-based formalisms which are known to

[1] Slavonic languages constitute a large group of the Indoeuropean language family and are further split into West, East and South Slavonic subgroups.

guarantee transparency and expressiveness. To be more precise, a grammar in SProUT consists of pattern/action rules, where the LHS of a rule is a regular expression over TFSs with functional operators and coreferences, representing the recognition pattern, and the RHS of a rule is a TFS specification of the output structure. Coreferences known from the unification-based formalisms express structural identity, create dynamic value assignments, and serve as means of information transport into the output descriptions. Functional operators provide a gateway to the outside world, and they are primarily utilized in two ways. Firstly, they are deployed for forming the output of a rule (e.g., concatenation of strings, converting complex number expressions into their corresponding numeric values) and, secondly, they can act as predicates that produce Boolean values, which can as well be utilized for introducing complex constraints in the rules.[2] Furthermore, grammar rules can be recursively embedded, which in fact provides grammarians with a context-free formalism. The following rule for the recognition of Prepositional Phrases (PPs) gives an idea of the syntax of SProUT grammar formalism:

```
pp :>  morph &   [ POS Prep
                   SURFACE #prep,
                   INFL [CASE #c ]]
       (morph &  [ POS Det,
                   INFL [ CASE #c,
                          NUMBER #n,
                          GENDER #g ]] ) ?
       (morph &  [ POS Adjective,
                   INFL [ CASE #c,
                          NUMBER #n,
                          GENDER #g ]] ) *
       (morph &  [ POS Noun,
                   SURFACE #noun1
                   INFL [ CASE #c,
                          NUMBER #n,
                          GENDER #g ]]
       (morph &  [ POS Noun,
                   SURFACE #noun2
                   INFL [ CASE #c,
                          NUMBER #n,
                          GENDER #g ]] ?
-> phrase & [CAT pp,
              PREP #prep,
              AGR agr & [ CASE #c,
                          NUMBER #n,
                          GENDER #g]
              CORE_NP #core_np]],
where #core_np=Append(#noun1," ",#noun2).
```

The first TFS matches a preposition. Then, one or zero determiners are matched. They are followed by zero or more adjectives. Finally, one or two noun items are consumed. The variables `#c`, `#n`, `#g` establish coreferences expressing the agreement in case, number, and gender for all matched items (except for the initial preposition item which solely agrees in case with the other items). The RHS of the rule triggers the creation of a TFS of type phrase, where the surface form of the matched preposition is transported into the corresponding slot via the variable `#prep`. A value for the attribute `CORE_NP` is created through a concatenation of the matched nouns (variables `#noun1` and `#noun2`). This is realized via a call to a functional operator called `Append` on the RHS of the rule. The formal specification of the grammar formalism is presented in (Drożdżyński et al., 2004.)

Grammars consisting of such rules are compiled into extended finite-state networks with rich label descriptions (TFSs). Consequently, the grammar interpreter uses a unifiability operation on TFS as the equality test while traversing such networks, whereas the construction of fully-fledged output structures is carried out through unification of the TFSs representing the matched items with a TFS-representation of the appropriate rule (Becker et al., 2002). Since fully specified TFSs usually do not allow for minimization and efficient processing of such networks, a handful of methods going beyond standard finite-state techniques have been deployed to remedy this problem. One of the speed-up techniques consists of sorting all outgoing transitions of a given state via a computation of a transition hierarchy under subsumption, which potentially reduces the number of time-consuming unification operations performed by the grammar interpreter. A further option allows for additional calibration of transition hierarchies, which exhibit a somewhat flat character, through an introduction of user-definable artificial transitions (Krieger & Piskorski, 2004).

3 Adopting SProUT to the Processing of Polish

Since SProUT provides some linguistic resources for the processing components for Germanic and Roman languages, we could exploit these re-

[2] SProUT comes with a set of circa 20 predefined functional operators.

sources in the process of fine-tuning SProUT to processing Polish w.r.t. NER.

3.1 Tokenization

The tokenizer in SProUT is Unicode compatible and allows for fine-grained token classification.[3] We adopted the available tokenizer resources by extending the character set with some specific Polish characters and adjusting some of circa 30 predefined token classes (e.g., currency sign, email, words containing both lowercase and uppercase characters, complex structures including hyphens etc.). For instance, the class `WORD-WITH-APOSTROPHE` for Polish defines all strings containing at least one apostrophe, whereas its counterpart for English or French are restricted to a proper subset for appropriate handling of contractions like "*it's*".

3.2 Morphological processing

SProUT comes with *Morfeusz*, a morphological analyzer for Polish developed by M. Woliński which uses a rich tagset based on both morphological and syntactic criteria (Przepiórkowski and Woliński, 2003). It is capable of recognizing circa 1,800,000 Polish contemporary word forms. Some work has been accomplished in order to infer additional implicit information (e.g., tense) hidden in the tags generated by *Morfeusz*. The following TFS exemplifies the result produced by the morphology component for the word *urzędzie* (office – locative and vocative form).

```
[ SURFACE ‶urzędzie′
  STEM ‶urząd″
  POS noun
  INFL [ CASE_NOUN loc_voc
         NUMBER_NOUN singular
         GENDER_NOUN masc3 ] ]
```

3.3 Gazetteer

The task of the gazetteer is the detection of full names (e.g., locations) and keywords (e.g., company designators) based on static lexica. Since extensive gazetteers constitute an essential resource in a rule-based NER system, some work has focused on acquisition of such resources. Apart from adapting a subset of circa 50,000 gazetteer entries for Germanic languages (mainly first names, locations, organizations, and titles), which appear in Polish texts as well, we acquired additional language-specific resources from various Web sources. The current status of the types and amount of NEs collected so far is depicted in the table in figure 1.

TYPE	AMOUNT	FEATURES
large companies	1211	CONCEPT, TYPE, SUBTYPE
federal government organizations	65	CONCEPT, TYPE
higher schools	68	CONCEPT, LOCATION
cities	1680	CONCEPT, COUNTRY
countries	215	CONCEPT, CASE, FULL-NAME, CAPITAL, CONTINENT
geographical regions	286	CONCEPT, TYPE
first names	350	CONCEPT, GENDER, CASE

Figure 1. Language-specific gazetteer entries

SProUT allows for associating gazetteer entries with a list of arbitrary attribute-value pairs. Due to the highly inflectional nature of Polish (e.g., nouns and adjectives decline in seven cases), this specific feature of the gazetteer comes particularly in handy. In this context, some of our efforts concentrated on manual and semi-automatic production of all orthographic and morphological variants for the subset of the acquired gazetteer resources. For instance, we implemented a brute-force algorithm which generates full declension of first names. The created gazetteer entries were additionally enriched with semantic tags and some basic morphological information, e.g., for the word form '*Argentyny*' (genitive form for *Argentyna*) the following entry has been created:

```
Argentyny  | concept:Argentyna
   | full-name:Republika Argentyńska
   | case:genitiv | capital:Buenos Aires
   | continent:South America
```

The tags used for each particular NE class are listed in the `FEATURE` column (c.f. figure 1). Note that the values in the column labeled `AMOUNT` refer only to the number of the different main (canoni-

[3] One of the major aims in the process of developing SProUT was a strong decomposition of linguistic analysis into clear-cut components. As a consequence of this, the context information is disregarded during token classification and sentence boundary detection constitutes a stand-alone module.

cal) forms (CONCEPT tag) in the given category. The actual number of entries is circa 10 times as high (e.g., for the 215 countries (and independent regions) there are 1727 entries).

Since producing all variant forms is a laborious job, and because the process of creating new names is very productive, a further way of establishing a better interplay between the gazetteer and the morphology module was achieved through an extension of the gazetteer processing module so as to accept lemmatized tokens as input. This solution is beneficial in case of single-word NEs covered by the morphological component. However, since declension of multi-word NEs in Polish is very complex, and frequently some of the words they comprise of are unknown, the next technique for boosting the gazetteer exploits the grammar formalism itself by introducing SProUT rules for the extraction, lemmatization and generation of diverse variants of the same NE from the available text corpora. The following rule demonstrates the idea.

```
org :> (morph & [ SURFACE #key,
                  STEM "urząd" & #stem,
                  INFL #infl]) |
       (morph & [ SURFACE #key,
                  STEM "komisja" & #stem,
                  INFL #infl]) |
       (morph &  [SURFACE #key,
                  STEM "komitet" & #stem,
                  INFL #infl]) |
       @seek(pl_np_gen) & [SURFACE #rest]
   -> gaz_entry & [  ENTRY #entry,
                     TYPE gaz_org,
                     SUBTYPE #stem,
                     CONCEPT #main,
                     INFL #infl],
where  #entry=ConcWithBlanks(#key,#rest),
       #main=ConcWithBlanks(#stem,#rest).
```

This rule identifies diverse morphological forms of keywords, such as *urząd* ’office’, *komisja* ‘commission’, or *komitet* ‘comitee’ followed by a genitive NP (realized by the seek statement). The RHS of the rule generates a gazetteer entry, where the functional operator ConcWithBlanks simply concatenates all its arguments and inserts blanks between them. For instance, the above rule matches all variants of the phrase *Urząd Ubezpieczeń Zdrowotnych* (Health Insurance Office). It is important to notice that in this particular type of constructions, only the keyword undergoes declension (*urząd*), whereas the rest remains unchanged. So even if the interpreter fails to recognize a genitive NP due to potential failure of the morphological unit, we could relax the overall rule by extending the call to the rule for genitive NPs with an alternative call to a rule which maps a sequence of capitalized words and conjunctions. This automatic lemmatization of unknown multi-words turned out to further boost the power of the gazetteer.

3.4 Reference matcher

Finally, the task of the reference matcher is to find identity relations between entities previously recognized in the text. Note that this component runs after grammar interpretation. It takes as input the output structures generated by the interpreter, potentially containing user-defined information on variants of the recognized entities for certain NE classes, and performs an additional pass through the text, in order to discover mentions of previously recognized entities[4]. The variant specification is done by defining additional attributes, e.g., VARIANT, on the RHS of grammar rules, which contain a list of all variant forms. For instance, for the entity *Dyrektor Prof. Stanisław Kowalski* ‘director prof. Stanisław Kowalski’ the VARIANT slot might include three forms*: Dyrektor Kowalski, Prof. Kowalski*, and *Dyrektor Prof. Kowalski*, obtained by simply concatenating some of the constituents of the full name. Interestingly, the size of the contextual frame (e.g., a paragraph) for tracking entity mentions is parametrizable.

As we have implicitly mentioned in the previous section, the essential information for creation of variants comes from the correct lemmatization of proper names, which is a challenging task with regard to Polish, especially for multi-word names. Let us briefly address lemmatization of person names. In general, both first name and surname of a person undergo declension. Lemmatization of first names is handled by the gazetteer which provides the main forms (at least for the frequently used Polish first names), whereas lemmatization of surnames is in some degree a more complex task. Firstly, we have implemented a range of rough sure-fire rules, e.g., rules that convert suffixes like {*-skiego*, *-skim*, *-skiemu*} into the main-form suffix *-ski*, which covers a significant part of the surnames. Secondly, for surnames which do not match any of the sure-fire rules, slightly more so-

[4] Current version of this component does not handle pronominal entity mentions.

phisticated rules are applied that take into account several factors including: the part-of-speech of the surname (e.g., noun, adjective, or unknown), gender of the surname (in case it is provided by the morphology), and even contextual information, such as the gender of the preceding first name (possibly provided by the gazetteer). For instance, if the gender of the first name is feminine (e.g., *Stanisława*), and the surname is a masculine noun (e.g., *Grzyb 'mushroom'*), then the surname does not undergo declension (e.g. main form: *Stanisława Grzyb* vs. accusative form: *Stanisławę Grzyb*). If in the same context the first name is masculine (e.g., *Stanisław*), then the surname would undergo declension (e.g. nom: *Stanisław Grzyb* vs. acc:*Stanisława Grzyba*). On the other hand, if the surname is an adjective it always declines. No later than now, can we witness how useful the inflectional information for the first names provided by the gazetteer is. A maze of similar lemmatization rules was derived from the bizarre proper name declension paradigm presented in (Grzenia, 1998). Nevertheless, in sentences like, e.g., *Powiadomiono wczoraj wieczorem G. Busha o ataku* '[They have informed] [yesterday] [evening] [G. Bush] [about] [the attack]', correctly inferring the main form of the surname *Busha* would at least involve a subcategorization frame for the verb *powiadomić* 'to inform' (it takes accusative NP as argument). Since subcategorization lexica are not provided, such cases are not covered at the moment.

The lemmatization component is integrated in SProUT simply via a functional operator. Hence, any extensions or adaptations to processing other languages w.r.t. lemmatization are straightforward. Lemmatization of organization names is done implicitly in the grammar rules as far as it is feasible (see Sections 3.3 and 4.1).

4 Named-entity Recognition

4.1 Grammar development

Within the highly declarative grammar paradigm of SProUT, we have developed grammars for recognition of MUC-like NE types (Chinchor and Robinson, 1998), including: persons, locations, organizations, temporal expressions, and quantities from financial texts. This task was accomplished with the visual grammar development environment provided by SProUT (see figure 3).

In the first step, to avoiding starting from scratch, we tried to recycle some of the existing NE-grammars for German and English by simply substituting crucial keywords with their Polish counterparts. As NEs mainly consists of nouns and adjectives, major changes focused on replacing the occurrences of the attribute `SURFACE` with the attribute `STEM` (main form) and specifying some additional constraints to control the inflection. Contrary to German and English, the role of morphological analysis in the process of NER for Polish is essential, since even rules for identifying such simple entities as time spans involve morphological information. This observation is exemplified with the following rule for matching expressions like *od stycznia do lutego 2003* 'from January till February 2003', where genitive forms of month names are required.

```
time_span :> token & [SURFACE "od"]
             (@seek(pl_month)
              & [ STEM #start,
                  INFL [ CASE_NOUN gen,
                         NUMBER_NOUN sg]])
             token & [SURFACE "do"]
             (@seek(pl_month)
              & [ STEM #end,
                  INFL [CASE_NOUN gen,
                        NUMBER_NOUN sg]])
             gazetteer & [GTYPE gaz_year,
                          CONCEPT #year]
-> timex & [ FROM [ MONTH #start,
                    YEAR #year],
             TO [ MONTH #end,
                  YEAR #year]].
```

As soon as we had addressed the issue of lemmatization, the major part of the rules created so far for the particular NE classes had to be broken down into several rules, where each new rule covers different lemmatization phenomenon. In section 3.4 we have discussed the issue of lemmatization of person names. Due to the fact that organization names are frequently built up of noun phrases, their lemmatization is even more complex and relies heavily on proper recognition of their internal structure. The following fragment of the schema for lemmatization of organization names with some examples visualizes the idea.

`[Adj] [N-key] NP-gen`
(e.g., [*Naczelnej*] [*Izby*] *Kontrolii*)

```
[Adj] [N-key] [Adj] NP-gen
```

(e.g., [*Okręgowy*] [*Komitet*] [*Organizacyjny*]
Budowy Autostrady)

N-key represents nominal keywords such as *ministerstwo* (ministry). The constituents which undergo declension are bracketed. For each rule in such schema a corresponding NER rule has been defined. However, the situation can get even more complicated, since NEs may have potentially more than one internal syntactical structure, which is typical for Polish, since adjectives may either stand before a noun, or they can follow a noun. For instance, the phrase *Biblioteki Głównej Wyższej Szkoły Handlowej* has at least three possible internal structures:

(1) [*Biblioteki Głównej*] [*Wyższej Szkoły Handlowej*] `[of the main library] [of the Higher School of Economics]',

(*2) [*Biblioteki Głównej Wyższej*] [*Szkoły Handlowej*] `[of the main higher library] [of the School of Economics]', and

(*3) [*Biblioteki*] [*Głównej Wyższej Szkoły Handlowej*] `[of the library] [of the Main Higher School of Economics]'.

This poses a serious complicacy in the context of lemmatization, not to mention singular-plural ambiguity of the word *biblioteki* (singular-gen vs. plural-nom-acc), etc. In order to tackle this problem, some experiments proved that an introduction of multiple keywords (e.g., '*Biblioteka Główna*' in the example above) would potentially reduce the number of ambiguities.

Last but not least, there exists another issue which complicates lemmatization of proper names in SProUT. We might easily identify the structure of organization names such as *Komisji Europejskiej Praw Człowieka* (of the European Commission for Human Rights), but the part which undergoes declension, viz. *Komisji Europejskiej* (of the European Commission) can not be simply lemmatized via a concatenation of the main forms of these two words. This is because *Morfeusz* returns the nominal masculine form as the main form for an adjective, which generally differs in the ending from the corresponding feminine form (masc: *Europejski* vs. fem: *Europejska*), whereas the word *Komisja* is a feminine noun. Once again, functional operators were utilized to find a rough workaround and minimize the problem.

Ultimately, somewhat 'more relaxed' rules have been introduced in order to capture entities which could not have been captured by the ones based on morphological features and ones which perform lemmatization. For example, such rules cover sequences of capitalized words and some keywords. Consequently, SProUTs' mechanism for rule prioritization has been deployed in order to give higher preference to rules capable of performing lemmatization, i.e., to filter the matches found by the interpreter. The current grammar consists of 143 rules.

SProUT provides a further mechanism for merging the matches into more informative structures via a sequence of unification operations. However, we have not yet used this option in the context of NER for Polish.

4.2 Evaluation

A corpus consisting of 100 financial news articles from a leading Polish newspaper has been selected for analysis and evaluation purposes. The precision-recall metrics are depicted in the table in figure 2. The results for persons, locations, and organization are somewhat worse due to the problems discussed in the previous sections.

TYPE	PRECISION	RECALL
TIME	81.3	85.9
PERCENTAGE	100.0	100.0
MONEY	97.8	93.8
ORGANIZATIONS	87.9	56.6
LOCATIONS	88.4	43.4
PERSONS	90.6	85.3

Figure 2. Precision-recall metrics

We also evaluated the quality of lemmatization. 79.6% of the detected NEs were lemmatized correctly. We expect to gain recall via providing additional gazetteer resources and improvement of the lemmatization of unknown multi-words.

Finally, an experiment on adjusting the speed-up option ('sorting transitions') for the grammar interpreter (cf. section 2) yielded for our NE-grammar a reduction of unification calls to 24% in comparison to the number of unifications performed without using this option.

5 Conclusions

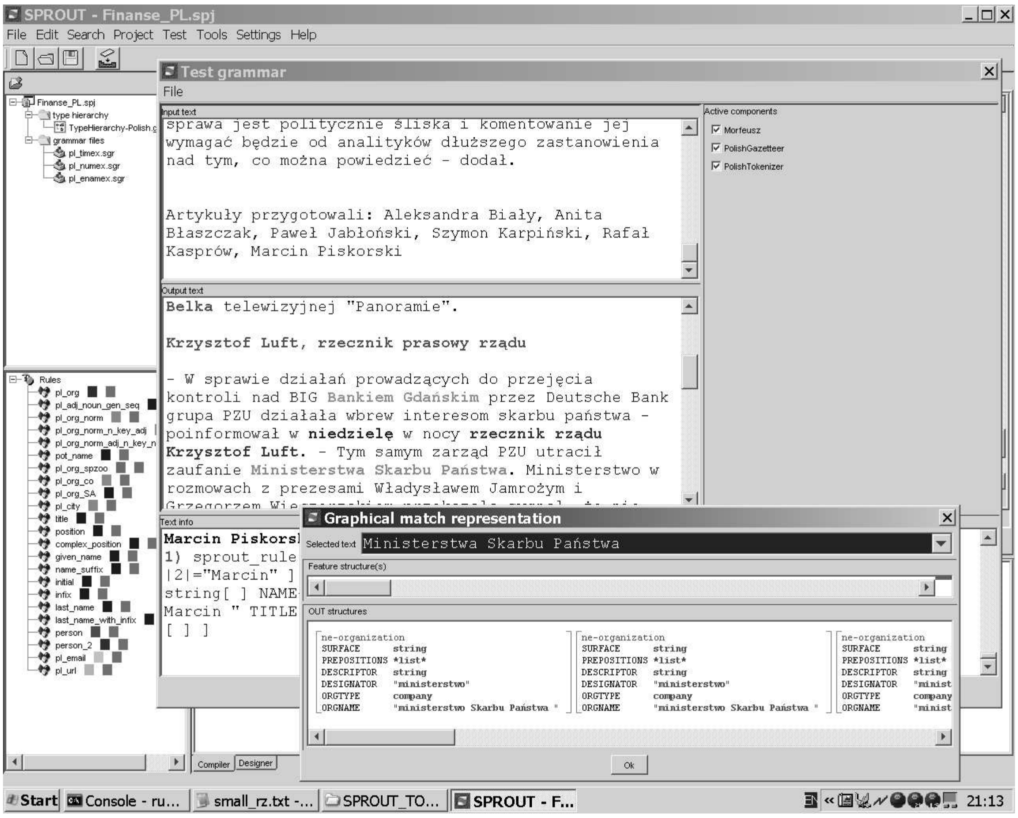
We have presented a preliminary attempt towards constructing a NER system for Polish via adapting and fine-tuning of SProUT, a flexible multi-lingual NLP system, and by introducing some language-specific components which could be easily integrated into SProUT through functional operators. The initial evaluation results turned to be very promising, although the recall values are still far away from the state-of-the-art results obtained for the more studied languages. Further, we pinpointed some peculiarities of Polish, which revealed the indispensability of the morphology component and the need of integrating additional nice-to-have components including lemmatizer for unknown multi-words (Erjavec and Džeroski, 2003), subcategorization lexicon, morphosyntactic tagger (Dębowski, 2003), and morphological generation module, in order to improve the performance of the presented approach, which is probably among the pioneering studies in the context of automatic NER for Polish.

While proximate work will concentrate on improving the overall system, in a parallel line of research an investigation of applying standard machine learning techniques to NER for Polish is envisaged. In particular, corpus annotation work is in the foreground.

6 Acknowledgements

The author is greatly indebted to Witold Drożdżyński and Petr Homola for their contribution to the task of adapting SproUT to Polish. Special thanks go to Anna Drożdżyńska and Marcin Rzepa for their work on acquisition of language specific resources.

The work reported in this paper was supported by the EU-funded project MEMPHIS under grant no. IST-2000-25045 and by additional non-financed personal effort of the author and the persons mentioned above.



Active components

☑ Morfeusz
☑ PolishGazetteer
☑ PolishTokenizer

Input text

```
sprawa jest politycznie śliska i komentowanie jej
wymagać będzie od analityków dłuższego zastanowienia
nad tym, co można powiedzieć - dodał.


Artykuły przygotowali: Aleksandra Biały, Anita
Błaszczak, Paweł Jabłoński, Szymon Karpiński, Rafał
Kasprów, Marcin Piskorski
```

Output text

```
Belka telewizyjnej "Panoramie".

Krzysztof Luft, rzecznik prasowy rządu

- W sprawie działań prowadzących do przejęcia
kontroli nad BIG Bankiem Gdańskim przez Deutsche Bank
grupa PZU działała wbrew interesom skarbu państwa -
poinformował w niedzielę w nocy rzecznik rządu
Krzysztof Luft. - Tym samym zarząd PZU utracił
zaufanie Ministerstwa Skarbu Państwa. Ministerstwo w
rozmowach z prezesami Władysławem Jamrożym i
```

Text info

```
Marcin Piskorsl
1) sprout_rule
|2|="Marcin" ]
string[ ] NAME
Marcin " TITLE
[ ] ]
```

Selected text: Ministerstwa Skarbu Państwa

OUT structures

```
ne-organization                                ne-organization                                ne-organization
SURFACE      string                            SURFACE      string                            SURFACE      string
PREPOSITIONS *list*                            PREPOSITIONS *list*                            PREPOSITIONS *list*
DESCRIPTOR   string                            DESCRIPTOR   string                            DESCRIPTOR   string
DESIGNATOR   "ministerstwo"                    DESIGNATOR   "ministerstwo"                    DESIGNATOR   "minist
ORGTYPE      company                           ORGTYPE      company                           ORGTYPE      company
ORGNAME      "ministerstwo Skarbu Państwa "    ORGNAME      "ministerstwo Skarbu Państwa "    ORGNAME      "minist
```

Figure 3. The SProUT grammar development environment.

References

D. Appelt and D. Israel. 1999. *An introduction to information extraction technology*. A Tutorial prepared for IJCAI-99 Conference.

M. Becker, W. Drożdżyński, H-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu. 2002. *SProUT - Shallow Processing with Typed Feature Structures and Unification*. In Proceedings of ICON 2002, Mumbai, India.

K. Bontcheva, D. Maynard, V. Tablan, H. Cunningham. 2003. *GATE: A Unicode-based Infrastructure Supporting Multilingual Information Extraction*. In Proceedings of the IESL 2003, Borovets, Bulgaria.

N. Chinchor and P. Robinson. 1998. *MUC-7 Named Entity Task Definition (version 3.5)*. In Proceedings of the MUC-7, Fairfax, Virginia, USA.

H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. 2002. *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. In Proceedings of the ACL'02, Philadelphia.

Ł. Dębowski. 2003. *A reconfigurable stochastic tagger for languages with complex tag structure*. In Proceedings of Morphological Processing of Slavic Languages, EACL 2003, Budapest.

W. Drożdżyński, H-U. Krieger, J. Piskorski, U. Schäfer, F. Xu. 2004. *Shallow Processing with Unification and Typed Feature Structures – Foundations and Applications*. In German AI Journal *KI-Zeitschrift*, Vol. 01/04, Gesellschaft für Informatik e.V.

T. Erjavec and S. Džeroski. 2003. *Lemmatising Unknown Words in Highly Inflective Languages*. In Proceedings of the IESL 2003, Borovets, Bulgaria.

J. Grzenia. 1998**.** *Słownik nazw własnych - ortografia, wymowa, słowotwórstwo i odmiana*. Publisher: PWN, Seria: Słowniki Języka Polskiego, ISBN: 83-01-12500-4.

V. Khoroshevsky. 2003. *Shallow Ontology-Driven Information Extraction from Russian Texts with GATE*. In Proceedings of the IESL 2003, Borovets, Bulgaria.

H-U. Krieger, J. Piskorski. 2004. *Speed-up methods for complex annotated finite-state grammars*. DFKI Report.

E. Paskaleva, G. Angelova, M. Yankova, K. Bontcheva, H. Cunningham, and Y. Wilks. 2002. *Slavonic Named entities in GATE*. Technical Report CS-02-01, University of Scheffield.

A. Przepiórkowski, and M. Woliński. *A flexemic tagset for Polish*. In Proceedings of Morphological Processing of Slavic Languages, EACL-2003, Budapest, Hungary.

M. Świdziński and Z. Saloni. 1998. *Składnia współczesnego języka polskiego*. Publisher: PWN, ISBN: 83-01-12712-0.