Control and Cybernetics

vol. **40** (2011) No. 2

Language resources for named entity annotation in the National Corpus of Polish*

by

Agata Savary[1] **and Jakub Piskorski**[2]

[1]Université François Rabelais Tours, Laboratoire d'Informatique
Blois, France

[2]Institute of Computer Science, Polish Academy of Sciences Warsaw, Poland

Abstract: We present the named entity annotation subtask of a project aiming at creating the National Corpus of Polish. We summarize the annotation requirements defined for this corpus, and we discuss how existing lexical resources and grammars for named entity recognition for Polish have been adapted to meet those requirements. We show detailed results of the corpus annotation using the information extraction platform *SProUT*. We also analyze the errors committed by our knowledge-based method and suggest its further improvements.

Keywords: natural language processing, proper names, named entities, corpus annotation, Polish National Corpus, SProUT.

1. Introduction and motivation

The development of linguistic resources is one of the key aspects in natural language processing (NLP). Such resources include electronic lexicons and grammars widely used in knowledge-based NLP applications, as well as annotated corpora supporting both linguistic research and data-based applications. The on-going project of the National Corpus of Polish (NKJP for *Narodowy Korpus Języka Polskiego*, http://nkjp.pl/)[1] is meant to create a large annotated versatile corpus of Polish language. It is designed so as to be representative and balanced with respect to different genres (Przepiórkowski et al., 2009), and assume several levels of annotation (Bański and Przepiórkowski, 2009), one of which addresses named entities (NEs). To the best of our knowledge, this is the first attempt at a large-scale comprehensive corpus annotation of Polish NEs. Its results are expected to boost the automatic named entity recognition (NER)

*Submitted: November 2010; Accepted: May 2011.

[1]Financed by a research and development grant KBN-R17-003-03 from the Polish Ministry of Science and Higher Education

and other higher-level information extraction tasks in Polish, similarly to other more resourced languages where NER has been a hot topic for over a decade.

The final NKJP corpus will consist of a high-quality manually annotated 1-million-word subcorpus, and an automatically annotated 1-billion-word main corpus. The manual subcorpus annotation, in order for it to be effective, needs an automatic pre-annotation by knowledge-based methods. We describe how lexical resources and grammars for Polish, existing within a shallow text processing and information extraction platform *SProUT* (http://sprout.dfki.de), have been adapted and extended to meet the NKJP requirements.

The remaining part of this article is organized as follows. In Section 2 we describe the existing, both traditional and electronic, lexicons of Polish proper names, and we resume the efforts for an automatic processing of Polish NEs. In Section 3 we recall the NE annotation scope, rules and strategies defined for the National Corpus of Polish. Section 4 shows how the existing Polish lexicons and grammars have been adapted and extended within the NKJP project, and Section 5 gives evaluation results with respect to the automatic corpus pre-annotation. The results of error analysis are described in Section 6. In Section 7 a comparative analysis with other approaches is provided. Finally, we give conclusions and perspectives for further work in Section 8.

2. State of the art

2.1. Traditional lexicography of Polish proper names

Onomastic studies on the origin, history and regional particularities of Polish proper names have been performed for decades (Rzetelska-Feleszko, 2005). Many efforts have also been made in traditional lexicography concerning such names, in particular by the members of the NKJP consortium. Rymut (2002) collects family names, with their distribution over Polish provinces and counties. Hydronyms, i.e. the names of rivers, canals, lakes, and other water bodies in Poland are listed in Rymut (2008). They are accompanied by encyclopedic, geographical, etymological and historical data. Kubiak-Sokół and Łaziński (2007) gather 7,400 selected names of Polish cities, towns, villages and other settlements, together with their derived adjectives and inhabitant names, as shown in example (1). This last dictionary, converted to a gazetteer, is largely used in the present project (see Section 4.2).

(1) **Hel** -lu, -lu
przymiotnik: helski
mieszkańcy: helanin, helanka
'Hel; genitive and locative endings: -lu, -lu; adjective: helski; inhabitants names (male and female): helanin, helanka'

2.2. Internet resources

Well known problems arise in converting traditional human-readable lexicons to electronic machine-readable ones. Moreover, all lexicons mentioned in the preceding section contain only strictly Polish names, i.e. names of persons living and objects located on the Polish territory. Clearly, resources containing names of persons and objects from outside of Poland are also necessary for the NKJP annotation.

An important source of such data are the publications of the Commission for Standardization of Geographic Names outside Polish Frontiers[2], freely available at `http://www.gugik.gov.pl/komisja/`. They contain very detailed data about main geographic objects in most countries in the world, together with their original and Polish names (if such exist). We drew the list of all countries in the world from this resource. Other publications of the Commission were too rich to be exploited rapidly, but they remain for us a normative reference, and a source of data for future efforts in computational lexicography.

Wikipedia (`http://pl.wikipedia.org/`), where Polish belongs to leading languages with respect to the number of entries, is another useful open resource of proper names. Several lists were drawn from Wikipedia for the needs of our project: (i) capitals of administrative units of different countries[3], (ii) rivers[4], (iii) historical regions of Europe[5], (iv) mountain chains[6], (v) adjectives and citizen names stemming from country names[7].

Another freely available Internet source of data was the *World Gazetteer*[8], containing population figures and area size for most countries, their administrative divisions, cities and towns. We drew the list of 200 biggest Polish cities from this website.

Finally, the data stemming from a heraldic service[9] yielded a list of Polish family names accompanied by numbers of their bearers.

2.3. Automatic processing of Polish named entities

Although considerable work on named-entity recognition for significant number of languages exists, relatively few efforts towards developing NER for Polish have been reported. To our best knowledge the work reported in Piskorski (2005) describes the first systematic attempt towards creation of a fully automated rule-based NER system for Polish, built on top of *SProUT* (see Section 4), covering

[2] Komisja Standaryzacji Nazw Geograficznych poza Granicami Rzeczypospolitej Polskiej

[3] `http://pl.wikipedia.org/wiki/Stolice_jednostek_administracyjnych`

[4] `http://pl.wikipedia.org/wiki/Rzeki_Afryki`, `http://pl.wikipedia.org/wiki/Rzeki_Azji`, etc.

[5] `http://pl.wikipedia.org/wiki/Kategoria:Regiony_i_krainy_historyczne_Europy`

[6] selected from several Wikipedia categories

[7] http://pl.wiktionary.org/wiki/Indeks:Polski_-_Państwa_Świata

[8] `http://www.world-gazetteer.com`

[9] `http://www.futrega.org/etc/nazwiska.html`

the classical named-entity types, i.e., persons, locations, organizations, as well as numeral and temporal expressions. The NER resources created in this first study have been adapted and further extended by Abramowicz et al. (2006) in order to create information extraction tools used in cadastral information systems.

Marcińczuk and Piasecki (2007) report on a memory-based learning approach to automatically extract information on events in the reports of Polish Stockholders. In particular, resources for extracting locations and temporal expressions for Polish were created. In a follow-up work, Marcińczuk and Piasecki (2010), focussing on the same domain, some accuracy results of an NER algorithm based on the Hidden Markov Model are presented. Also in Lubaszewski (2007 and 2009) some general-purpose information extraction tools for Polish are addressed.

Other recent efforts led to an annotated corpus of dialogs concerning the Warsaw transportation system (Mykowiecka et al., 2008), as well as an electronic dictionary of Warsaw urban proper names (streets, bus stops, buildings, etc.) oriented towards NE recognition and synthesis of both text and speech (Savary et al., 2009; Marciniak et al., 2009).

Graliński et al. (2009b) present *NERT*, another rule-based NER system for Polish, which covers similar types of NEs as Piskorski (2005), but the underlying grammar formalism is significantly simpler. *NERT* has been mainly implemented for deployment in machine anonymisation and translation (Graliński et al., 2009a).

3. Annotation rules for named entities in the Polish National Corpus

The rules admitted for named entities in the NKJP project result from a compromise between the precision of linguistic data and the richness of naming phenomena in Polish texts.

In Savary et al. (2010) we describe the scope of the NE annotation chosen for NKJP, as well as their type hierarchy inspired by TEI P5 (Burnard and Bauman, 2008), as shown in Fig. 1. We take into account most NE types common for different NE projects, such as names of persons, locations, organizations, and numerical expressions. Note that some differences exist in our list of basic NE categories with respect to other state-of-the-art approaches such as Sekine et al. (2002). Notably, locations are distributed within two types called `placeName` and `geogName`. According to TEI P5, the former is meant for hierarchically-organized geo-political or administrative units (districts, regions etc.), while the latter refers simply to objects having geographical features such as mountains or rivers. This distinction may be useful because names of administrative units frequently appear as metonyms (designating the inhabitants of the unit), in which case they should be seen as organizations rather than locations (see Chinchor, 1997). Clearly, many units, if considered out of context, are

potentially ambiguous between the `placeName` and the `geogName` categories, e.g. *mazowiecki* can be seen as an adjective related to the place name *województwo mazowieckie* 'Masovian Voivodship' or to the historical region name `Mazowsze` 'Masovian Region'. Within the corpus most of such ambiguities could be solved so far.

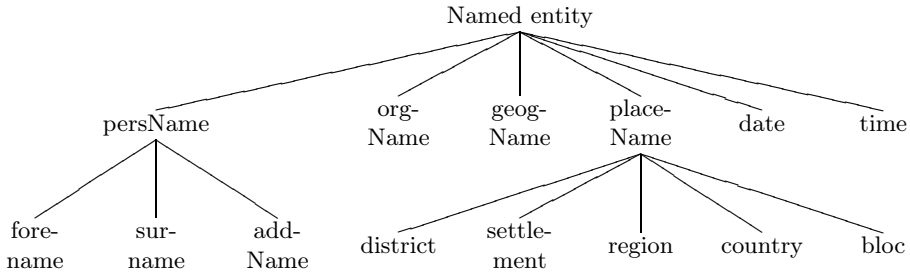

Figure 1. Type hierarchy of Polish NEs

For the time being we do not, however, annotate other NEs, such as events, quantities and measures, product and vessel names, titles of works and texts. Within numerical expressions we do not treat expressions of duration (*przez dwa dni* 'for two days'), sets (*co drugi dzień* 'every other day') and relative time (*wczoraj* 'yesterday'). We are, though, interested in some units that are less frequently covered by other projects, such as relative adjectives (denoted by attribute `@derivType` equal to `relAdj`) stemming from person, location and organization names, as well as names of inhabitants and organization members (denoted by attribute `@derivType` equal to `persDeriv`). These derived names form a typology which is vertical to the one from Fig. 1.

Traditional NER, MUC and CoNLL campaigns have focused on identifying and classifying flat longest-match NEs. More recent research shows the importance of representing the internal structure in recursively embedded NEs (Galicia-Haro and Gelbukh, 2007; Finkel and Manning, 2009b; Kravalová and Žabokrtský, 2009) and their overlapping with nominal phrases (Finkel and Manning, 2009a; Osenova and Kolkovska, 2002) in multi-level annotation. Thus, in NKJP we annotate each NE together with other NEs possibly included in it. For instance:

(2) [[Maria]$_{forename}$ [Skłodowska]$_{surname}$-[Curie]$_{surname}$]$_{persName}$

(3) [ulica [[Mikołaja]$_{forename}$ [Kopernika]$_{surname}$]$_{persName}$]$_{geogName}$
'Mikołaj Kopernik Street'

(4) [[Wydział Prawa]$_{orgName}$
[Uniwersytetu [Warszawskiego]$_{relAdj:settlement(Warszawa)}$]$_{orgName}$]$_{orgName}$
'Law Faculty of the University of Warsaw'

We believe that such representation has two advantages: (i) it enlarges the density of annotated NEs in the corpus, (ii) it facilitates further treatment of

coreferences, as well as relations occurring between different NEs, (iii) it may help in NE type disambiguation.

As mentioned before, the NKJP admits a stand-off multilevel annotation. The level of named entities is defined over the level of so-called syntactic words, which are minimal single tokens or groups of tokens corresponding to traditional parts of speech[10]. The syntactic words, in their turn, are built over morphosyntactic annotation of elementary tokens. Thus, from the annotation of each NE we have an access to morphosyntactic annotation of its constituents. Note that (Savary et al., 2007) the morphosyntax of a compound NE is not always a straightforward function of the morphosyntax of its constituents. However, within the NKJP project we do not annotate the morphosyntax of NEs manually. We expect, instead, that it can be deduced later, largely automatically, from the underlying level of syntactic words, from the lemma of each NE, and from the annotated syntactic groups (Głowińska and Przepiórkowski, 2010).

Apart from the main type, and possibly the subtype of the NE, other annotated attributes important for the creation of resources and grammars include:

- Lemma (attribute `@base`, e.g. *Stany Zjednoczone* for *Stanów Zjednoczonych* 'United States')
- TEI-P5-inspired normalization of date and time (attribute `@when`, e.g. *2009-10-30*, *09:45:00*)
- For derivations, the basic NEs they were derived from (attribute `@derivedFrom`, e.g. *Francja* 'France' for *francuski* 'French')

Note that determining the lemma of a NE, is a non trivial task in a highly inflected language such as Polish, in particular for compound and personal names, as discussed in Piskorski et al. (2009). That is why we put a special emphasis on the creation of NE resources containing such lemmas, as well as their automatic deduction in grammar rules.

4. Adapting the SProUT extraction platform to Polish named entity annotation

SProUT (Becker et al., 2002; Drożdżyński et al., 2004) is a general purpose multi-lingual NLP platform. It is equipped with a set of reusable Unicode-capable on-line processing components for basic linguistic operations and a cascaded unification-based finite-state grammar parser and interpreter. The basic processing components include, i.a., tokenizer, sentence splitter, morphological analyzer, gazetteer look-up component, etc. They can be flexibly combined into a pipeline that generates several streams of linguistically annotated structures, which serve as an input for the cascaded grammar interpreter, applied at the next stage. *SProUT* has been adapted to processing Polish (Piskorski et al.,

[10]The syntactic words are being annotated simultaneously to the NEs, thus, temporarily the NE level builds upon morphosyntactic segments, i.e. tokens.

2004), and grammars for extracting 'classical' named-entities (e.g., names of persons, organizations, locations, etc.) from Polish texts have been developed (Piskorski, 2005). In the remaining part of this section we mention the particularities related to adapting *SProUT* to processing Polish and we describe how the existing NER grammars were utilized and extended for the purpose of the named-entity annotation task.

4.1. Morphological analysis and generation

In order to be able to perform the morphological analysis, *Morfeusz* (Woliński, 2006), a morphological analyzer for Polish, which uses a rich tagset based on both morphological and syntactic criteria (Przepiórkowski and Woliński, 2003), has been integrated. We are presently using its two different versions. The older one, called *Morfeusz SIAT*, is fully integrated with *SProUT* as the module for Polish morphological analysis. It recognizes circa 1,800,000 word forms, including few proper names. The newer version, *Morfeusz SGJP*, enlarged with a morphological generation module (Savary et al., 2009), is used as a stand-alone tool for creating "gazetteers" (see Section 4.2). It contains a large Grammatical Dictionary of Polish (Saloni et al., 2007), including about 4,000,000 word forms corresponding to 250,000 lemmas. Some 10,000 of those lemmas are single-word proper names (mainly forenames, surnames and locations).

4.2. Gazetteers

In information extraction the term *gazetteer* is often used for a domain-specific list of entries allowing to customize general data-driven NLP applications. In *SProUT* a gazetteer is a dictionary containing ontological and/or morphological information describing either a domain-specific or a general-language vocabulary. Each gazetteer entry may be associated with a list of arbitrary attribute-value pairs. The gazetteer look-up component in *SProUT* simply recognizes occurrences of gazetteer entries in text. Thus, vocabulary description in a gazetteer has to be done extensively, i.e. by explicitly listing all inflected forms of each entry, as shown below. Since Polish is a highly inflected language, the list of those forms may range from several to several dozens for each lemma. Therefore, efficient compression and look-up procedures are needed to offer real-time corpus processing (Daciuk and Piskorski, 2006; Budiscak et al., 2009).

For the NKJP project we obtained Polish gazetteers developed by Piskorski (2005). They contain a subset of circa 70,000 uninflected entries for Germanic languages (mainly first names, locations, organizations, and titles), as well as additional language-specific resources acquired from various Web sources. Notably, there are some 60,000 inflected forms of Polish and foreign first names, and 1,500 forms of Polish position names (e.g. *poseł* 'depute', *generał* 'general').

These data have been completed by subsets of resources mentioned in Section 2.2, as well as by the relational adjectives and inhabitant names stemming

from Kubiak-Sokół and Łaziński (2007). In order to obtain inflected forms of those names we created a text filter chain. The resources were grouped into files according to their categories (rivers, mountains, countries, etc.). Multi-word entries were initially eliminated from the lists. The *Morfeusz SGJP* generator (see Section 4.1) was applied to each list, and the inflected forms of the list entries known to *Morfeusz* were retained. Incorrect homonymic entries were eliminated (e.g. *Morfeusz* recognized *Apolinary* as masculine human singular forename, while the same form appeared as a plural-only settlement name). Each form was decorated by the information about its category, as well as about the source of both its lemma and its inflected forms (these data are needed for further management of license-bound resources, such as those mentioned in Section 2.1). The inhabitant names unknown to *Morfeusz* were decorated with base form inflectional values only. The resulting forms were transformed into the *SProUT* gazetteer format, as shown in examples (5)-(7).

Obviously, many NEs are multi-word units, usually corresponding to well-formed Polish nominal groups. Complex declension system, and gender-number-case agreement and government rules in Polish, call for specialized methods of generating inflected forms of such units. *Multiflex* (Savary et al., 2009) is a tool responding to this need. Some multi-word inflected forms were inherited from Piskorski (2005). We have completed them by new entries (countries, cities, and rivers) described and generated within *Multiflex*, and transformed into the *SProUT* gazetteer forms, as shown in example (8).

(5) **Buga** | GTYPE:gaz_river | G_LEMMA:Bug | G_GNUMBER:singular
| G_CASE:gen | G_GENDER:masc3 | G_SOURCE:Wikipedia
| G_INFL_SOURCE:Morfeusz

(6) **Kowalskim** |GTYPE:gaz_surname | G_LEMMA:Kowalski
| G_GNUMBER:singular | G_CASE:ins | G_GENDER:masc1 | G_SOURCE:Futrega
| G_INFL_SOURCE:Morfeusz

(7) **Helskim** | GTYPE:gaz_city_deriv | G_LEMMA:helski | G_NUMBER:singular
| G_CASE:loc | G_GENDER:masc1_masc2_masc3_neutrum1_neutrum2
| G_DERIV_TYPE:reladj | G_DERIVED_FROM:Hel
| G_SOURCE:PWN_Miejscowe | G_INFL_SOURCE:Morfeusz
| G_LETTER_CASE:first-upper

(8) **Skarżyskiem-Kamienną** | GTYPE:gaz_city
| G_LEMMA:Skarżysko-Kamienna
| G_GNUMBER:singular | G_CASE:ins | G_GENDER:neutrum2
| G_SOURCE:WorldGazetteer | G_INFL_SOURCE:Morfeusz_Multiflex

(9) **inspektorowi** | GTYPE:gaz_position | G_LEMMA:inspektor | G_CASE:dat
| G_GENDER:masc1 | G_GNUMBER:singular

Some words, which are not individual proper names themselves, can help in extracting NEs in texts. Some of such trigger words (also known as internal and external evidences, see Section 4.4) appeared in the gazetteers created by Piskorski (2005), and were completed in the current project, as shown in example (9).

They include: positions (*inspektor* 'inspector'), titles (*Prof. Dr.* 'professor doctor'), personal name infixes (*van der*), days of week (*środa* 'Wednesday'), months (*luty* 'February'), uppercase initials (*K*), integers describing years, months, days of month, hours, minutes and seconds (*1999, 12, 28, 23, 59, 5, 05*).

All aforementioned lists were merged with the previously created gazetteers, and compiled into a binary gazetteer ready for lookup in *SProUT*. As a result, we dispose currently of a gazetteer, whose composition is presented in Fig. 2 (homonyms and syncretic forms are distinguished).

Data category	Lemmas	Inflected Forms
First names	17,068	39,461
Family names	17,474	85,651
Organizations	1,752	1,863
Countries and regions	373	3,472
Cities	2,952	6,569
Rivers and mountains	361	2,003
Adjectives	1,871	128,424
Inhabitants	12,292	19,387
Trigger words	564	2,409
TOTAL	54,707	289,239

NE types	Rules
Persons	29
Organizations	20
Locations	25
Temporal expr.	24
Derivations	5
Auxiliary	17
TOTAL	120

Figure 2. Composition of the Polish SProUT gazetteers and NER grammars

4.3. Type hierarchy

All information in *SProUT* is typed. Types are abbreviations for feature structures. They allow for a modular, thus easily manageable, representation of treated objects, and help avoid bugs in grammar development, since many potential semantic errors are shifted into the syntactic level.

The type hierarchy designed for Polish in Piskorski (2005) has been adapted so as to match the needs of the NKJP project, as shown in Fig. 3. Parts of the hierarchy[11] which have been left intact concern: (i) several built-in subtypes (lines 1–2), (ii) types of tokens (we reuse the *SProUT*-native tokenizer with which *Morfeusz SIAT* was harmonized, see lines 4–5), (iii) the typology of the Polish morphology[12] (lines 6–11). The type describing a gazetteer entry (lines 12-19) inherits three attributes from the `sign` type, which are crucial for our grammars, as can be seen in Section 4.4: (i) SURFACE, which indicates the exact string occurring in the text (i.e. the surface realization of a concept), (ii) CSTART

[11] The upper case identifiers name attributes, the lower case ones name types, *avm* is the most general feature structure with no attributes, the ":<" operator means 'is a subtype of', the ":=" means 'extends'.

[12] This is a simplified version of morphological types. In the actual hierarchy also combinations of different cases, genders, etc., constitute separate types useful in case of morphological syncretism.

```
1  *avm*                    := *top*.
2  string                   :< *top*.
3  index-avm                := *avm*.
4  tokentype                := index-avm.
5  all_capital_word, lowercase_word, first_capital_word, ...:< tokentype.
6  part_of_speech, infl     :< *avm*.
7  adjective                :< part_of_speech.
8  infl_adjective := infl & [CASE_ADJECTIVE case, GENDER_ADJECTIVE gender,
9                            NUMBER_ADJECTIVE number, DEGREE_ADJECTIVE degree].
10 case                     :< *avm*.
11 nom, gen, ...            :< case.
12 gtype                    := index-avm.
13 gaz_surname              :< gtype.
14 gaz_city                 :< gtype.
15 ...
16 sign := *avm* & [SURFACE string, CSTART string, CEND string].
17 gazetteer := sign & [GTYPE gtype, LEMMA string, G_NUM_BASE string,
18                      G_CASE case, G_GENDER gender, G_NUMBER number,
19                      G_SOURCE string  G_INFL_SOURCE string, ...].
20 agr-nkjp:= *avm*&[NE_NUMBER number,NE_CASE case,NE_GENDER gender].
21 ne-nkjp-type              :< *top*.
22 place_name               :< ne-nkjp-type.
23 country                  :< place_name.
24 ...
25 int_proof_geog_name      :< string.
26 "Góra", "Półwysep",...   :< int_proof_geog_name. ;;'Mountain, Peninsula'
27 ext_proof_geog_name      :< string.
28 "rzeka", "zatoka", ...   :< ext_proof_geog_name. ;;'river, bay'
29 ne-nkjp                  := sign &  [BASE string, NE_TYPE ne-nkjp-type,
30                                      TREE string, MORPH agr-nkjp].
```

Figure 3. Extract of the Polish type hierarchy

corresponding to the position of the first character of that string, counted from the beginning of the text file, (iii) CEND corresponding to the position of its ending character. CSTART and CEND are automatically instantiated for each occurrence of an elementary unit (token, gazetteer entry or morphological unit). The `gazetteer` type has been further redefined so as to fit the dictionaries described above. Note that if an attribute defined by the hierarchy does not appear in a gazetteer entry, it is instantiated to the most specific known type, e.g. G_SOURCE takes the value `string` for the entry in example (9). New attributes in this type are G_NUM_BASE, G_SOURCE, and G_INFL_SOURCE allowing to indicate respectively: the normalized version of a month name (e.g. *02* for *luty* 'February'), the source of an entry's lemma (e.g. *Wikipedia*), and the source of its inflected forms (e.g. *Morfeusz*, *Morfeusz_Multiflex*, *Manual*).

Further we define a basic morphological structure for all NEs, which are nominal phrases or adjectives (line 20), and we formalize the NE topology depicted in Fig. 1 (lines 21–24). We include in the type hierarchy about 170 lexical items (lines 25–28) which frequently appear inside or in the vicinity of named entities (35 of them appeared in the previous hierarchy), the so-called internal and external evidences. We use them as triggers in grammar rules. Finally, we define the main output structure common for all extracted NEs (lines 29–30). It inherits the SURFACE, CSTART and CEND attributes from `sign`, and includes 4 other attributes. BASE corresponds to the lemma of a (possibly multi-word) named entity form, NE_TYPE is one of the types from Fig. 1 (not necessarily a leaf type), MORPH carries its inflectional features, and TREE is crucial for representing embedded structures, as explained in Section 4.4.

4.4. Grammars

For the Named-Entity Annotation task, we have also rearranged the NE grammars described in Piskorski 2005, containing over 160 rules (39 of them dedicated to personal names, 77 to temporal expressions, 23 to numerical expressions, 12 to locations, 10 to organizations).

A grammar in *SProUT* consists of the so-called pattern/action rules, where the left-hand side (LHS) is a regular expression over typed feature structures (TFS), representing the recognition pattern, and the right-hand side (RHS) is a TFS specification of the output structure. Additionally, functional operators may be used on both sides of the rules. They provide a gateway to the outside world, and they are primarily utilized for forming the output of a rule (e.g., lemmatization of small-scale structures) and for introducing complex constraints.

For instance Fig. 4 shows a simple rule, named `surname_gaz_based`. The LHS is delimited from RHS with `->`. The symbol `&` denotes unification, and variables are strings preceded by the symbol `#`. Here the LHS allows to recognize any gazetteer entry provided that it is a surname (all attributes other than `GTYPE` are free variables that can be instantiated to any values of the proper types). The RHS triggers creation of an `ne-nkjp` structure. Seven slots are assigned values. In particular, `SURFACE`, `BASE` and `MORPH` are directly instantiated, via variables, with the corresponding attributes from the gazetteer entry. Starting and ending character numbers, `CSTART` and `CEND`, are straightforwardly instantiated by the analyzer. The `TREE` attribute is used for storing nested annotations (see below) that are transformed into trees in the annotated corpus. Its value is created by the functional operator `ConcWithBlanks`, which concatenates (with separating blanks and bars) the surface form, the lemma, the beginning and ending character number, and the priority of the interpretation. For instance, the value of this attribute for the occurrence of entry (6), Section 4.2, at position 127 in the input text would be `[ Kowalskim | Kowalski | forename | 127 | 135 | prio_1 ]`.

```
surname_gaz_based :/ gazetteer & [SURFACE #surface, G_LEMMA #lemma,
                                  GTYPE gaz_surname,G_NUMBER #number,
                                  G_CASE #case, G_GENDER #gender,
                                  CSTART #s, CEND #e]
->
ne-nkjp & [SURFACE #surface, BASE #lemma, NE_TYPE surname,
           MORPH  agr-nkjp & [NE_NUMBER #number,
                              NE_CASE #case,NE_GENDER #gender],
           TREE #tree, CSTART #s, CEND #e],
where #tree=ConcWithBlanks("[", #surface, "|", #lemma, "| forename |",
                           #s, "|", #e, "| prio_1 ]").
```

Figure 4. Grammar rule for the recognition of a forename belonging to the gazetteer

```
person_1 :> ((@seek(full_position) & #position])(token & [TYPE comma])?)?
             (@seek(title) & #title) ?
             (@seek(forename) & [SURFACE #surf1, BASE #lemma1, MORPH #morph,
                                 TREE #tree1, CSTART #s1, CEND #e1])
             (@seek(forename) & [SURFACE #surf2, BASE #lemma2, MORPH #morph,
                                 TREE #tree2, CSTART #s2, CEND #e2]]) ?
             (@seek(surname) & [SURFACE #surf3, BASE #lemma3, MORPH #morph,
                                TREE #tree3, CSTART #s3, CEND #e3] & #surname)
             (@seek(name_suffix) & #suffix)?
->
ne-nkjp & [SURFACE #surface, BASE #lemma, TYPE persName,
           TREE #tree, CSTART #s1, CEND #e3],
where #surface = ConcWithBlanks(#surf1, #surf2, #surf3),
      #lemma = ConcWithBlanks(#lemma1, #lemma2, #lemma3),
      #tree = ConcWithBlanks(#tree1,#tree2,#tree3,
               "[",#surface,"|",#lemma,"| persName |",#s1,"|",#e3," ]").
```

Figure 5. Grammar rule for the recognition of a person name, with embedded rule calls

Priorities are used in the postprocessing phase when two concurrent rules offer different interpretations of a matched sequence. For instance, two other rules allow to recognize surnames such as *Kowalskiej* ('Kowalski' in feminine genitive) on the basis of: (i) their homonymy with common nouns or adjectives (here *kowalskiej* 'related to a smith'), (ii) their simple orthographic features, such as initial uppercase letter. When these rules are applied, the generated structures contain lemmas equal to *Kowalski* and *Kowalskiej*, respectively, while the correct lemma mentioned in the gazetteer is *Kowalska*. We solve this problem by producing value `prio_1` by gazetteer-based rules, `prio_2` by morphology-based ones (lemmas they generate are sometimes correct, e.g. for masculine adjectival proper names) and `prio_3` by token-based rules[13]. Unfortunately, this mecha-

[13]The problem of concurrent analyses is handled in other approaches by a cascaded processing, combined with a proper rule prioritization, however, the particular design of the *SProUT* cascade seems inadequate for our corpus annotation task due to retaining, for the second cascade level, only the structures produced on the first level.

SURFACE	Janem Kowalskim
BASE	Jan Kowalski
TYPE	persName
	[Janem | Jan | forename | 121 | 125 | prio_1]
TREE	[Kowalskim | Kowalski | surname | 127 | 135 | prio_1]
	[Janem Kowalskim | Jan Kowalski | persName | 121 | 135]
CSTART	121
CEND	135

Figure 6. Structure resulting from processing the text *Prezydentem Janem Kowalskim* by the rule `person_1`

nism does not handle ambiguities between names obtaining the same priority, e.g. when *Wisła* as a town and as a river obtain `prio_1` the choice between both interpretations is arbitrary.

Grammar rules can be recursively embedded. Fig. 5 shows the `person_1` rule for recognition of person names. First, an optional ('?' denotes optionality) position and title are matched, via a call to adequate rules: `@seek(full_position)` and `@seek(title)`. Next, one or two forenames are sought: `@seek(forename)`. Finally, a surname is consumed by an embedded rule roughly equivalent to Fig. 4. In the resulting `ne-nkjp` structure the `SURFACE` slot is created via concatenation of the forenames and the surname (call to `ConcWithBlanks(#surf1, #surf2, #surf3)`, whereas the `BASE` collects base forms on the LHS. The attribute `TREE` is a list of the `TREE` values of the embedded names, followed by the description of the whole structure. For instance, matching the text fragment *Prezydentem Janem Kowalskim* would result in producing the structure depicted in Figure 6.

As mentioned in Section 4.3, about 170 lexical items are defined in the *SProUT* type hierarchy as external and internal evidences of NE occurrences. These items are used in grammar rules as triggers for particular NE types. For instance, Fig. 7 shows a grammar rule matching geographical names preceded by an external evidence, e.g. *rzeka Okawango* 'Okawango river'. Note that the first token matched by this rule, here *rzeka*, must be a morphologically known item whose lemma belongs to the list of external proofs for geographical names, as defined in the type hierarchy (see line 28 in Fig. 3). When such a token is followed by a capitalized token (with or without a hyphen) the resulting structure of type `geogName` contains the second token only. For instance, when applied to the sequence *na rzece Okawango* 'on the Okawango river', starting at character 2011, the rule produces the structure depicted in Fig. 8.

Conversely, when a rule based on an internal evidence is applied, the resulting structure contains the trigger word. For instance, Fig. 9 shows a rule matching geographical names like *Półwysep Helski* 'Hel Peninsula'. Calling the rule `capitalized_noun` allows to consume a trigger word, here *Półwysep* 'Peninsula', provided that it is a capitalized geographical internal proof (see line 26 in Fig. 3). The following item must be a relational adjective appear-

```
geogr_names_ext_proof_2 :/
   (morph & [SURFACE #surface1, STEM ext_proof_geog_name & #lemma1,
             CSTART #s1, CEND #e1])
   ( (token & [TYPE first_capital_word, SURFACE #surface2,
               CSTART #s2, CEND #e2]) |
     (token & [TYPE word_with_hyphen_first_capital, SURFACE #surface2,
               CSTART #s2, CEND #e2]) )
->
ne-nkjp & [SURFACE #surface2, BASE #surface2, NE_TYPE geog_name,
           TREE #tree, CSTART #s2, CEND #e2],
where Capitalized(#surface2),
      #tree=ConcWithBlanks("[", #surface2, "|", #surface2,
                           "| geogName |", #s2, "|", #e2, "| prio_3 ]").
```

Figure 7. Grammar rule for the recognition of geographical names preceded by an external evidence, e.g. *rzeka Okawango*

SURFACE	Okawango
BASE	Okawango
TYPE	geogName
TREE	[Okawango | Okawango | geogName | 2020 | 2027 | prio_3]
CSTART	2020
CEND	2027

Figure 8. Structure resulting from processing the text *na rzece Okawango* by the rule `geogr_names_ext_proof_2`

ing in the gazetteer, as in example (7). The resulting structure obtained for the sequence *na Półwyspie Helskim* (starting at character 11332) is depicted in Fig. 10. Its TREE attribute contains the embedded structure for the relational adjective constructed on the basis of gazetteer data by the `reladj_gaz_based` rule.

The final (slightly simplified) example illustrates a problem occurring during the lemmatization of Polish compound names. Namely, the lemma of a multi-word name can contain tokens which are not lemmas themselves. For instance in *Izba Skarbowa* 'Tax Chamber' the adjective *Skarbowa* 'tax-related' is in nominative feminine, while its lemma, as an individual token, is in nominative masculine (*skarbowy*). Therefore, when this name appears in an inflected form, e.g. in genitive *Izby Skarbowej*, its correct lemmatization cannot simply rely on lemmatizing both components. The rule `adj_with_special_stem` in Fig.11 allows to recognize an adjective such as *Skarbowej* and assign it a special stem, which is in nominative, but in the same gender as the recognized form, here in feminine. This process is possible due to the Polish-specific functional operator `CorrectSuffixPL` introduced to *SProUT* by Piskorski (2005). Further, the rule `capitalized_adj_with_special_stem` imposes that this adjective starts with a capital letter, and transforms its special stem *skarbowa* to the capitalized variant *Skarbowa*. Eventually, the rule `org_trigger_adj` matches a capitalized organization internal proof, such as *Izba* 'Chamber', followed by the capitalized

```
geogr_names_int_proof_1 :/
   @seek(capitalized_noun) &
   [SURFACE #surface1, STEM int_proof_geog_name & #lemma1,
    INFL infl_noun & [GENDER_NOUN #g, NUMBER_NOUN #n, CASE_NOUN #c],
    CSTART #s1, CEND #e1]
   @seek(reladj_gaz_based) &
   [SURFACE #surface2, BASE #lemma2,
    MORPH  agr-nkjp & [NE_NUMBER #n, NE_CASE #c, NE_GENDER #g],
    TREE #tree2, CSTART #s2, CEND #e2]
->
ne-nkjp & [SURFACE #surface, BASE #lemma, NE_TYPE geog_name, TREE #tree,
           CSTART #s1, CEND #e2],
where Capitalized(#surface2),
      #surface=ConcWithBlanks(#surface1, #lemma2),
      #lemma=ConcWithBlanks(#lemma1, #surface2),
      #tree=ConcWithBlanks(#tree2, "[", #surface, "|", #lemma,
                           "| geogName |", #s1, "|", #e2, "| prio_1 ]").
```

Figure 9. Grammar rule for the recognition of geographical names containing an internal evidence, e.g. *Półwysep Helski* 'Hel Peninsula'

SURFACE	Półwyspie Helskim
BASE	Półwysep Helski
TYPE	geogName
TREE	[Helskim | Helski | settlement | relAdj | Hel | 11345 | 11351 | prio_2]
	[Półwyspie Helskim | Półwysep Helski | geogName | 11335 | 11351 | prio_1]
CSTART	11335
CEND	11351

Figure 10. Structure resulting from processing the text *na Półwyspie Helskim* 'on the Hel Peninsula' by the rule `geogr_names_int_proof_1`

adjective with a special stem (*Skarbowa*). The resulting structure in Fig. 12 contains the fully correct lemma *Izba Skarbowa*. Note that both tokens selected by this rule have to agree in gender, number and case due to common unification variables `#g`, `#n` and `#c`.

While comparing the original grammar from Piskorski (2005) with our present NKJP grammar we note different impact on various elements of the resulting structures. The former grammar is meant for information extraction. Extracted names are linked with encyclopedic data, thus there is more granularity in the output types. For a person name we wish to return not only the information about his/her fore- and surname, but also his/her sex, title (*Mrs.*), function (e.g. *prezydent* 'president'), etc. We also extract types (governmental, academic; river, lake, etc.) and location (country) of organizations and places. In NKJP these data are not needed, thus a unique type `ne-nkjp` covers all entities. However, due to the NKJP annotation rules we make a particular effort in identifying embedded names. Both grammars share the full attention paid to determining the proper lemma for each recognized entity.

```
adj_with_special_stem :/
  (morph & [SURFACE #surface, STEM #lemma,
   INFL infl_adjective & [GENDER #g]] & #infl)
->
morph & [SURFACE #surface, STEM #new_lemma, INFL infl_adjective & #infl],
where #new_lemma=CorrectSuffixPL(#lemma,#g).

capitalized_adj_with_special_stem :/
  @seek(adj_with_special_stem) & [SURFACE #surface, STEM #lemma,
                                  INFL infl_adjective & #infl]
->
morph & [SURFACE #surface, STEM #new_lemma, INFL infl_adjective & #infl],
where Capitalized(#surface),
      #new_lemma=CapitalizeWord(#lemma).

org_trigger_adj :/
  @seek(capitalized_noun) &
      [SURFACE #surface1, STEM int_proof_org_name & #lemma1,
       INFL infl_noun & [GENDER_NOUN #g, NUMBER_NOUN #n, CASE_NOUN #c],
       CSTART #s1]
  @seek(capitalized_adj_with_special_stem) &
      [SURFACE #surface2, BASE #lemma2,
       MORPH  agr-nkjp & [NE_GENDER #g, NE_NUMBER #n, NE_CASE #c],
       TREE #tree2, CEND #e2]
->
ne-nkjp &[SURFACE #surface, BASE #lemma, NE_TYPE org_name, CSTART #s1, CEND #e2],
where #surface=ConcWithBlanks(#surface1, #surface2),
      #lemma=ConcWithBlanks(#lemma1,  #lemma2),
      #tree=ConcWithBlanks("[", #surface, "|", #lemma, "| orgName |",
                           #s1, "|", #e2, "| prio_2 ]").
```

Figure 11. Grammar rules for the recognition of organization names containing adjectives with a special stem, e.g. *Izba Skarbowa 'tax chamber'*

SURFACE	Izby Skarbowej
BASE	Izba Skarbowa
TYPE	orgName
TREE	[Izby Skarbowej | Izba Skarbowa | orgName | 710 | 723 | prio_2]
CSTART	11335
CEND	11351

Figure 12. Structure containing the correct feminine stem for *Izby Skarbowej* 'tax chamber'

.

The current numbers of rules for NKJP, covering various types of named entities are summarized in Fig. 2. The fact that our rules are by more than 40 less numerous than the previous set is mainly due to a rather restricted scope of temporal expressions to be annotated in NKJP.

5. Evaluation

As discussed in Savary et al (2010) and Waszczuk et al. (2010), the annotation of named entities in the NKJP corpus relied initially on an iterative methodology: (i) we automatically parsed texts with *SProUT*, (ii) the results were corrected and completed through human annotation, (iii) we used error reports from annotators to improve the *SProUT* grammars until we could consider them relatively stable. By June 2010, about 34% of the corpus (over 387,000 tokens), henceforth called the *development corpus*, has been processed in this way. The remaining 66%, henceforth the *evaluation corpus*, was treated without any further changes operated on the *SProUT* grammar.

The quality insurance procedures in the NKJP project assume that the automatic pre-annotation of each corpus text is manually corrected and completed by two independent annotators. Furthermore, discrepancies between two annotators are resolved by an adjudicator within the so-called super-annotation process.

In order to evaluate the final version of our grammar most accurately we proceed in the following way:

1. We only use the evaluation corpus texts.
2. For each text we consider its annotation produced by the latest *SProUT* grammar.
3. We compare this *SProUT* output with the super-annotated version of the same text.

Table 1 shows how many occurrences of NEs belonging to different NE categories have been found in the evaluation corpus. *Persons*, being the most numerous NEs, correspond to all NEs of type `persName`, and possibly of any of its subtypes (see Fig. 1). *Locations* represent all NEs of types `geogName` or `placeName` (and any of its five subtypes). *Organisations* relate to type `orgName`. *Temporal expressions* designate types `date` and `time`. Finally, *derivations* embrace items having the attribute `derivType` (equal to whether `relAdj` or `persDeriv`), and relative to any of the first four types and/or their subtypes in Fig. 3. Two different levels of embedding are considered: (i) all existing NE occurrences, in particular those embedded in other NEs, (ii) only the longest-match occurrences, i.e. those NEs that are not embedded in any other larger NEs.

For grammar evaluation, these two embedding levels are combined with two levels of attribute granularity. Firstly, for each NE we consider all its attributes (relevant to its type): (i) its text range, i.e. the list of tokens from the morphosyntactic level, which are attached to this NE, (ii) its type, (iii) its subtype (if any), (iv) its lemma, (v) its standardized date form (if any), (vi) its derivation type and base (if any). Secondly, we take only the first three attributes into account.

As shown in Table 2, we obtain four sets of precision and recall results, accordingly. The first scenario (all NEs and all attributes) corresponds precisely

Table 1. Density of NEs in the evaluation part of the gold standard corpus

	Persons	Locations	Organizations	Temporal expr.	Derivations	Overall
All NEs	32,015	9,238	6,039	3,458	5,428	56,178
Longest-match NEs	13,905	8,518	5,757	3,448	4,392	36,020

Table 2. Precision and recall of the NE recognition

Category	Evaluation range			
	All NEs All attributes	All NEs Tokens, types, subtypes	Longest-match NEs All attributes	Longest-match NEs Tokens, types, subtypes
Persons	P=0.71 R=0.30	P=0.81 R=0.34	P=0.66 R=0.23	P=0.79 R=0.28
Locations	P=0.73 R=0.45	P=0.76 R=0.47	P=0.68 R=0.45	P=0.71 R=0.47
Organizations	P=0.55 R=0.21	P=0.62 R=0.23	P=0.54 R=0.21	P=0.60 R=0.24
Temp. expr.	P=0.78 R=0.55	P=0.83 R=0.59	P=0.78 R=0.55	P=0.83 R=0.59
Derivations	P=0.76 R=0.55	P=0.80 R=0.57	P=0.71 R=0.58	P=0.73 R=0.60
Overall	P=0.72 R=0.36	P=0.78 R=0.39	P=0.68 R=0.35	P=0.74 R=0.38

to the context of our project, i.e. to the automatic corpus pre-annotation prior to manual correction, where possibly embedded NEs and their whole range of attributes are to be annotated. Thus, for instance, 70% of all person names suggested by *SProUT* are fully correct, i.e. require no action from the annotator (except their validation). Conversely, 71% of all existing person names are not correctly recognized by *SProUT*. That does not mean, however, that these names are completely overseen. Many of them are spotted, however, some amount of correction and completion actions is required for them from the annotator.

The second scenario (all NEs, tokens, types and subtypes only) shows how well *SProUT* deals with the basic set of attributes, without going into morphological and semantic details.

The third scenario (longest-match NEs, all attributes) shows *SProUT* efficiency with morphologically and semantically detailed treatment of non-embedded NEs.

Finally, the fourth scenario (longest-match NEs, tokens, types and subtypes only) corresponds to the most classical NER task, in which embedding and morphology are not an issue. When considering these figures with respect to other NER approaches, notably those based on machine learning, note, however, that the comparison is not completely straightforward. Namely, we present results in which an evaluation item is a (possibly multi-word) named entity rather than a text token (classified as belonging or not to a named entity of a certain type).

Analysing the results in Table 2 we can draw some general conclusions. Obviously, the figures are higher if only tokens, types and subtypes are considered than when all attributes are taken into account. The difference between these

two cases ranges from 0.02 to 0.13 for precision, and from 0.02 to 0.05 for recall. In most scenarios the best quality is obtained for temporal expressions, outscored only by derivations in recall for longest-match occurrences. Conversely, the most problematic names to recognize in each case are organization names, with precision not higher than 0.62 and recall below or equal to 0.24. Note also that our grammar privileges precision over recall. Since good recall seems crucial within corpus pre-annotation (it is faster to manually delete an incorrect NE than to insert a new NE) we have tried to introduce relaxed grammar rules such as e.g. 'if a token appears in a gazetteer as a forename, and it hasn't been annotated before by a contextual rule, then annotate it as a forename independently of its context'. This, however, resulted in a big number of false positives, which are homonyms between frequent Polish functional words and inflected forms of surnames. For instance *Ale* in the beginning of a sentence is ambiguous between the conjunction 'but' and the plural form of the surname *Ala*. Thus, relaxed rules of this kind were not satisfactory.

These results show that automatic pre-annotation prior to human annotation speeds up the construction of the corpus. As many as 72% of all NEs suggested by *SProUT* can be retained as such by the annotator, saving several manual actions per NE. Correct lemmas and derivation bases are most important here, since they allow for avoiding error prone text editing (the values for other attibutes are selected from closed lists). We also think that the general results are satisfactory when texts of very different genres are to be processed, such as the rather noisy data of spoken dialogs.

Let us finally note that application of grammars to text is relatively slow. Namely, applying our NE grammar to a 1-million-word corpus took 45 minutes and 18 seconds. While this may not be satisfactory for real-time text processing it suffices in the context of automatic off-line corpus pre-annotation.

6. Error analysis

We have performed an analysis of errors committed by our grammar in the evaluation corpus for each NE category.

For all categories the most frequent false positives are those where an incorrect list of tokens has been matched, i.e. the NE is either fully redundant or its left and/or right frontier is incorrect. We shall analyse these cases in deeper detail before developing further the grammar. The following remarks result from the analysis of all other types of errors, i.e. those occurring in NEs, whose text range has been correctly recognized, as well some false positives.

6.1. Errors in personal names

For personal names the most frequent error concerns the incorrect base form. This originates directly from the incompleteness of our gazetteers. For instance, the genitive masculine forename and surname *Stanisława Gucwy* was recognized

by a rule which searches for a known forename (*Stanisława*) followed by an unknown capitalized token (*Gucwy*). The output structure contains a compound lemma built of the lemmatized forename and the recopied surname (**Stanisław Gucwy*) since no hints are accessible on how to correctly lemmatize this surname (*Stanisław Gucwa*). Some other masculine fore- and surnames can be ambiguous with respect to their feminine equivalents. For instance, *Mirosława Lenka* was considered by a grammar rule as a feminine name in nominative, while it was in fact a masculine name in genitive that should be lemmatized to *Mirosław Lenek* instead of **Mirosława Lenka*. Also, some spelling errors appearing in the corpus could result in spurious lemmas. For instance, the name *Włodzimierza Lenina* 'Vladimir Lenin in genitive' was misspelled as **Włodzimierza Lenia*, which resulted in the false lemma **Włodzimierz Leń*. Other interesting problems specific to lemmatization of Polish personal names have been largely discussed by Piskorski et al. (2007).

The second most frequent type of errors is due to subtype confusion which can easily occur when narrow context is given for disambiguation. For instance, in the sequence *słowa pani Zofii zagłuszył trzask* 'a crackle drowned out the words of madame Zofia', the name *Zofii* has been incorrectly tagged as a surname. That is because in the gazetteer *Zofii* appears as both a forename and a surname. A grammar in its turn says that the trigger word *pani* 'madame' can be followed by either a fore- or a surname if no other capitalized token follows. Thus, two different output structures are created and only one of them is randomly chosen during post-processing (an annotator can only be presented with one proposal for each NE). Thus, this frequent problem is not caused by a grammar error but rather by grammar ambiguity. Also, some subtype errors occur when a nickname appears in a context where we usually expect a surname, as in *Aleksander Macedoński*. Few hints allow to guess that *Macedoński* should be tagged here by subtype `addName` instead of `surname`. Such cases probably need explicit mentions in the gazetteer.

Some correctly spotted names are assigned a false type, which is sometimes due to true ambiguities in text. For instance *ul. Opolskiej* 'Opole Street in locative' has been correctly tagged as geographical name (a street name) however its embedded name *Opolskiej* is ambiguous between the genitive of the surname *Opolska* and the locative feminine of the relative adjective *opolski*. External world knowledge is needed here in order to choose the right interpretation.

One of the main reasons for false negative personal names is the lack of disambiguating context, particularly in literary texts, as in *Widmar pomyślał: "Będzie deszcz"* 'Widmar thought: "It will rain"'. If extra lexico-semantic knowledge were available with respect to predicates (verbs, predicative adjectives and names), some heuristics could help analyze such cases. For instance, if we could access, from inside the grammar rules, the information that the verb *myśleć* requires a human subject, the above example could be reliably tagged as a personal name (the proper disambiguation between a fore- and a surname would be more difficult though).

6.2. Errors in location names

Frequent attribute errors in location names are related again to the lemma, in particular in case of plural and neuter names. For instance *Gór Złotych* 'Golden Mountains' in genitive is lemmatized to **Góra Złota* 'Golden Mountain' in singular instead of *Góry Złote* in plural. That is due to the fact that the Polish-specific functional operator `CorrectSuffixPL`, discussed in Section 6.1 does not handle plural and neuter words. Thus, *Gór* can only be transformed to nominative singular but not to nominative plural. An extension of this functional operator could help handling such cases properly.

Cases of frequent subtype errors concern mainly homonyms, e.g., *Rzym* 'Rome' and *Praga* 'Prague' appear as country name (ancient Rome) and a district of Warsaw, respectively, but are tagged as cities, instead. Here again errors are due to ambiguous *SProUT* output filtered out randomly during post-processing. The results could be better in some cases if a disambiguating context and extra morphosyntactic knowledge were accounted for. Thus e.g., we could use the fact that the prepositions typically preceding a place name in locative are *na* and *w* for a district and a settlement, respectively (*na Pradze* vs. *w Pradze*).

Among the false negatives in this category we find non ambiguous names missing in the gazetteer, e.g. *Tatry* 'the Tatra Mountains'. Some unrecognized street names show that more relaxed rules could be added to the grammar, which currently matches the trigger *ulica* 'street' or *ul.* 'str.' only, if followed by a known adjective or person name. Allowing for any capitalized word after these triggers would help recognize cases like *ul. Lanciego* 'Lanci Street'.

Another very frequent type of errors concerns a systematic ambiguity of locations and human collectives, i.e. of types `placeName`/`geogName` and `orgName`. For instance, the continent name *Europa* 'Europe' is used often as a synonym of the European Union or the population of Europe. According to annotation guidelines such occurrences should be annotated as organization names, but they are hard to distinguish from genuine locatives on the basis of available context.

6.3. Errors in organization names

Symmetric cases of wrongly resolved ambiguities occur for this category. For instance *Teatr Polski* 'Polish Theater' is preferably tagged as an institution, though in some contexts it concerns a building, which should be tagged as `geogName`. As above, the available context is here of little help.

In this category many incorrect lemmas are due to the gazetteer content. Some gazetteer entries contain spelling errors in their lemmatized forms. Some others indicate lemmas that are full forms of acronyms. For instance the entry *PZPR* is assigned the lemma *Polska Zjednoczona Partia Robotnicza* 'Polish United Workers' Party'. Such entries stem from a gazetteer developed for previ-

ous NER projects. The NKJP annotation rules assume, however, that acronyms should not be not developed to their full forms but only lemmatized in case of inflection (e.g. the lemma for *PZPR-owi* should be *PZPR*).

Several cases of incorrectly lemmatized names allowed to track a bug in the grammar: the names consisting of a feminine organization internal proof and followed by a feminine adjective, as in *Politechniki Warszawskiej* 'of the Warsaw University of Technology', were lemmatized to nominative and to masculine for the second component, *Politechnika *Warszawski*. As `CorrectSuffixPL` operator handles such adjectives correctly, we could see that no rule was designed to cover this case. Moreover, we saw the need to make this operator yet more sophisticated so as to adapt not only to the original gender of the word lemmatized but also to its other inflectional categories. For instance, the name *Najwyższą Izbę Kontroli* 'Highest Inspection Chamber' in accusative was lemmatized to **Wysoka Izba Kontroli* 'High Inspection Chamber' instead of *Najwyższa Izba Kontroli*. Here, the first adjective was correctly put to nominative feminine but it should not have been turned from superlative to positive degree.

We also detected cases where additional rules, based on internal or external evidence could improve the grammar precision. For instance, there is no rule dedicated to organization names, using external evidence. Thus, sequences like *organizacja Greenpeace* 'Greenpeace organization' yield correct names non existent in the gazetteer, which obtain an incorrect type (here `persName`), although the preceding word *organizacja* clearly indicates the type `orgName`.

6.4. Errors in temporal expressions

All false positives occurring in the temporal expressions concern the `when` attribute. Their analysis allows for detecting a repetitive error in grammar rules. Namely, according to the ISO standard, admitted in the annotation schema, the normalized form of a time expression should be always in the form HH:MM:SS, even if the indication of minutes or seconds does not appear in the text. Thus, for instance *godz. 15.30* was normalized as **15:30* instead of *15:30:00*. The corresponding rules can be corrected quite straightforwardly.

Some discrepancies found in the *SProUT* annotation enabled detection of errors occurring not in the grammar rules, but in the gold standard corpus, due to a conceptual error in the early stage of corpus development. Namely, the ISO standard for dates was initially wrongly understood, and dates having no year component missed the initial dash character. For instance *23 marca* 'on the 23rd of March' was initially normalized as **-03-23* instead of *--03-23*. This bug was rapidly corrected but some early annotated texts still bear errors of this kind. They should disappear after final quality checks on the corpus.

We have also examined some false negative temporal expressions. They show that additional rules are necessary to account for dates in which the order of components is unusual, e.g., *roku 1933* 'in 1933' instead of *1933 roku*, and in which numbers are spelled out *czwartego stycznia* 'the fourth of January'. More-

over, expressions concerning centuries and decades, e.g. *XIX wiek* 'XIX century' and *lata siedemdziesiąte* 'the seventies', are not covered by any rule. Finally, the set of recognizable dates is limited by the gazetteer contents. Although dates show rather regular behavior and are of theoretically unlimited number, no mechanism in *SProUT* can presently account for this fact. Thus, a year, in order to be recognized properly, must appear explicitly in the gazetteer. This makes the gazetteer maintenance rather tedious and limits the number of possible dates. Since currently only years from *1800* to *2019* are included, dates like *1652 r.* could not be correctly recognized.

6.5. Errors in relative adjectives and inhabitant names

Most attribute errors occurring in this category concern ambiguous derivational stems (i.e. the `derivedFrom` attribute). Such stems appear in the gazetteer entries built on the basis of Kubiak-Sokół and Łaziński (2007), as explained in Section 4.2. For instance, the adjective *leszczyński* relates to settlements *Leszczyna* and *Leszno*, the adjective *opolskim* to both *Opole* and *Opole Lubelskie*. Many adjectives are also homonyms of personal names, e.g. *Słowacki* is either the relational adjective relating to the country of Slovakia or a surname, and both interpretations appear in the gazetteer. Here again, when the *SProUT* output is randomly disambiguated during post-processing, an incorrect interpretation is often retained. We need, therefore, a mechanism allowing, in case of ambiguity, for giving higher probabilities to more frequent interpretations. Such heuristics could be based, for instance, on the sizes of the corresponding settlements.

Quite a few errors of this type concern the single adjective *żydowski* 'Jewish'. There exists a Polish settlement *Żydowo* whose relational adjective is a homonym of the nation-related adjective *żydowski*. Moreover, the gazetteer does not yet contain relational adjectives for people, nations and tribes that cannot be attached to a unique country or region. These lacking elements, frequent in corpora, should undoubtedly be completed.

Some errors result from a limited number of frequent derivatives. *Włoch, Czech* and *Niemiec* 'an Italian, a Czech, a German' are homonyms of the genitive forms of the corresponding country names, Italy, Czech Republic and Germany. The capitalized relative adjectives *Polski* and *Polska* 'Polish in masculine and feminine' are homonyms of the genitive and nominative forms of Poland, respectively. Note that the value of case is most often a disambiguating feature in these examples. Thus, the immediate context could be explored in order to find verbs or prepositions that require objects in a particular case. This, however, calls for lexical resources that are non existent in *SProUT* yet.

7. Comparative analysis with other approaches

Even if the study described here is dedicated to the task of corpus annotation prior to human post-editing, our work is clearly inspired by the general state

of the art in the task of Named Entity Recognition and Classification (NERC). As mentioned in the survey by Nadeau and Sekine (2007), the latter task was identified by the Message Understanding Conference (MUC) in 1996[14] as having crucial importance in information extraction, and it was further promoted by evaluation campaigns within MUC-7, CoNLL-2002[15] and CoNLL-2003[16]. Many systems in the early stage of NERC for English were based on hand-crafted rules and gazetteers, like our own approach. Later on, the trend was towards an increasing use of data-based methods: supervised (requiring a large annotated training corpus), semi-supervised (using seeds of sample names to extract contexts for new names), and unsupervised (using clustering, co-occurrence analysis or external resources like WordNet). For other languages, e.g. Portuguese, rule-based approaches are still dominant, as shown in the recent evaluation campaign HAREM (Freitas et al., 2010). Since the NERC for Polish is not yet as developed as for other widely studied languages, and few large NE-annotated corpora in Polish exist, we attempt in this section to compare our approach mainly to the existing rule-based or hybrid NERC systems.

Gazetteers used in different knowledge-based approaches to NERC may have very variable sizes and numbers of features associated to an entry. For instance, in Farmakiotou et al. (2000) the Greek gazetteer contains about 3,000 entries representing only lemmas, although the Greek language has a relatively rich inflection. The reason is that the items in the source text are stemmed before being subject to rule matching. In Gaizauskas et al. (1995) a flat gazetteer of about 6,000 English names and trigger words is used. In Wolinski et al. (1995) 8,000 French names are represented in a knowledge base, which relates them to attributes such as type, domain and location, and to their aliases (orthographic variants, acronyms, etc.). In Wacholder et al. (1997) the gazetteer contains 3,000 English trigger words and 20,000 first names. In Mikheev et al. (1999) the English gazetteer consists of 45,000 entries, which is comparable to our Polish gazetteer containing about 55,000 lemmas. Our gazetteer entries, however, benefit from relatively large lists of grammatical and semantic features. In Schäfer (2006) – another NERC study based on *SProUT* – such gazetteers with rich sets of features (here including 20,000 English entries) are automatically extracted from OWL/RDF-encoded ontologies. Note, though, that the quality of NERC does not necessarily improve with the growing size of gazetteers. In Mikheev et al. (1999) arguments are given for using application-tuned gazetteers of limited size rather than far-fetched examples of little known places and organizations. Our observations on the manual post-editing of the National Corpus of Polish, which followed the *SProUT* application, confirm this proposal.

The size of different named-entity grammars is not always indicated in the reference papers. Our approach with 120 rules may be comparable, e.g., to Gaizauskas et al. (1995), who use about 200 rules (almost 50% thereof address

[14] http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_toc.html
[15] http://www.clips.ua.ac.be/conll2002/
[16] http://www.clips.ua.ac.be/conll2003/

organization names). Note, though, that the number of rules does not always give a good idea of the coverage and precision of the grammar, as it is related to the particular grammar formalism. For instance Appelt et al. (1995) use only 12 so called macro rules and 15 domain-dependent rules. However, the authors claim that due to compile-time transformations these rules cover approximately as many phenomena as would be described by a hundred explicit patterns.

Finite-state methods are frequently used in NERC rule-based engines. They are often supported by cascading mechanisms, where the transformed text output from lower-level rules becomes the input for higher-level rules, giving more expressive power to the resulting systems. This mechanism is used, e.g., by Hobbs et al. (1997) for English and by Friburger and Maurel (2004) for French. In our approach, although cascading is provided in *SProUT*, we do not benefit from this mechanism, due to particularities of its implementation discussed in Section 4.4.

As far as the list of types and subtypes covered by NERC systems are concerned, our approach is more fine-grained than the systems evaluated within MUC and CoNLL campaigns, such as those by Appelt et al. (1995), Gaizauskas et al. (1995), and Mikheev et al. (1999). They use three main types: ENAMEX (proper names), TIMEX (temporal expressions), and NUMEX (expressions of quantities and measures), completed by artefacts. The first type is subdivided into three categories: names of persons, locations and organizations. The French evaluation campaign ESTER-2 (Galliano et al., 2009) for NERC in spoken corpora uses a much richer typology with 7 main types and 78 subtypes. All competing systems – e.g. Nouvel et al. (2010) which is the rule-based transducer-cascade system having evolved from Friburger and Maurel (2004) – take this hierarchy into account. In HAREM, the NERC evaluation for Portuguese (Freitas et al., 2010), the admitted typology is even larger: it consists of 10 main types and 47 subtypes. An interesting methodological innovation in this last framework is accounting for possible vagueness of NE interpretation by allowing more than one tag per annotated NE. According to our experience with manual corpus post-editing, this feature proves useful in some cases of actual ambiguity of types and/or attributes. Since the beginning of the 21st century international community proposes an elaborated standard *TimeML*[17] for annotation and normalization of temporal expressions. With respect to this standard our annotation is rather elementary: we account for absolute dates and time expressions but not e.g. for relative times, periods, etc. Let us finally note that in none of the cited systems we found mentions of annotating relative adjectives and personal derivations stemming from named entities. In this aspect our approach might be seen as novel.

The NE type attached to a recognized entry may be accompanied by a series of application-dependent attributes, like in our approach, this, though, being rarely discussed in the literature. In Wolinski et al. (1995) such attributes

[17] http://www.timeml.org

include city, sector of activity, market, financial index, etc. Note that our approach pays special attention to correct lemmatization of NEs. We found no explicit references to this problem except in the original *SProUT* grammar by Piskorski (2005) and in the dedicated study by Piskorski et al. (2007).

8. Conclusions and perspectives

We have presented the named entity annotation task, which is a part of the ongoing project aiming at creation of the multi-level annotated National Corpus of Polish. We have shown how existing resources and grammars for information extraction have been adapted to meet the requirements of corpus annotation. The results show that human annotation can be substantially supported by automatic pre-annotation. Nevertheless, further improvement of both gazetteers and grammar rules is possible according to hints gained from the analysis of false positive and false negative NE candidates. Notably, most entries in our gazetteers are strictly Polish names. Adding more foreign names is necessary, but two important problems arise in this case: (i) foreign names may or may not have Polish translations or transliterations, (ii) many of such names may be inflected according to Polish declension system. Moreover, some heuristics should be found for homonyms, which constitute a high percentage of committed errors. More elaborate functional operators for *SProUT* are needed to cover complex lemmatization rules in Polish compound names.

Further error analysis should concentrate on the NE candidates which only partly overlap with correct NE occurrences, as well as on other false negative NEs, whose rather high number results in low recall. We hope to design additional rules allowing to reliably match new correct candidates. In particular, we intend to benefit from elliptical variant recognition offered by *SProUT*. It would allow to cover occurrences of single names (e.g. *Sadowski*) whose more explicit variants (e.g. *Andrzej Sadowski*) have been detected elsewhere in the same text. We hope for an integration of the *Morfeusz SGJP* analyzer (see Section 4.1), to increase the lexicon coverage. Furthermore, we plan to exploit the NER rules developed for the *NERT* formalism, Graliński et al. (2009a) to improve the coverage. Finally, having manually annotated the over 1-million-word NKJP gold standard corpus, we implemented machine learning CRF-based tools trained on the gold standard, to be applied to the 1-billion-word main corpus. In the next step we wish to develop hybrid annotation based on rule-based and machine learning methods. Namely, the CRF-based tool significantly outperforms *SProUT*, but deals only with identifying (longest-match and embedded) NEs and categorizing them. It does not cope with attributes such as lemmas, normalized dates and derivation bases, which *SProUT* can annotate reasonably well. Thus, we think that merging both tools is at least possible within a framework where each tool is dedicated to separate set of tasks. This final tool will require evaluation with respect to other existing NER tools for Polish. Note also that, while a morphological analyzer of Polish is integrated in *SProUT*, an

improvement of the grammar results could be obtained by integrating efficient tagger such as those by Piasecki and Wardyński (2006) or Acedański (2010).

Currently, another information extraction-oriented platform is being adapted to the processing of Polish texts, namely, *ExPRESS*, Piskorski (2008), which can be seen as a lightweight version of *SProUT*. Grammar rules in *ExPRESS* are regular expressions over flat feature structures, i.e., non-recursive feature structures, whose features are string valued. Furthermore, the formalism supports neither unification nor structure sharing, although these features can be simulated via functional operators. The somewhat weaker rule formalism of *ExPRESS* is compensated by significantly better run-time performance. As a matter of fact, the major design goal of *ExPRESS* was the ability to efficiently process vast amount of textual data. In particular, the linguistic resources for NE Annotation task described in this article will be converted into *ExPRESS* format in order to develop a time-efficient NER component for Polish.

References

Abramowicz, W., Filipowska, A., Piskorski, J., Wecel, K. and Wieloch, K. (2006) Linguistic Suite for Polish Cadastral System. In: *Proc. of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy. ELRA, 2518–2523.

Acedański, Sz. (2010) A Morphosyntactic Brill Tagger for Inflectional Languages. In: *Advances in Natural Language Processing. Proc. of the 7th International Conference on Advances in Natural Language Processing.* **LNAI 6233**, Springer-Verlag, 3–14.

Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., Kameyama, M., Kehler, A., Martin, D., Myers, K. and Tyson, M. (1995) SRI International FASTUS system: MUC-6 test results and analysis. In: *Proc. of the Sixth Message Understanding Conference (MUC-6).* NIST, Morgan-Kaufmann Publishers, 237–248.

Bański, P. and Przepiórkowski, A. (2009) Stand-off TEI Annotation: the Case of the National Corpus of Polish. In: *Proc. of the Third Linguistic Annotation Workshop (ACL-IJCNLP 2009), Suntec, Singapore.* Association for Computational Linguistics, 64–67.

Becker, M., Drożdżyński, W., Krieger, H.U., Piskorski, J., Schäfer, U. and Xu, F. (2002) SProUT - Shallow Processing with Typed Feature Structures and Unification. In: *Proceedings of the International Conference on NLP (ICON 2002), Mumbay, India.*

Budiscak, J., Piskorski, J. and Ristov, S. (2009) Compressing Gazetteers Revisited. In: *Pre-proceedings of the FSMNLP'09, Pretoria, South Africa.* University of Pretoria.

Burnard, L. and Bauman, S., eds. (2008) *TEI P5: Guidelines for Electronic Text Encoding and Interchange.* TEI Consortium, Oxford.

Chinchor, N. (1997) MUC-7 Named Entity Task Definition. In: *Proc. of the Message Understanding Conference (MUC-7).* Linguistic Data Consortium.

Daciuk, J. and Piskorski, J. (2006) Gazetteer Compression Technique Based on Substructure Recognition. *Advances in Soft Computing*, **35**, 87–96.

Drożdżyński, W., Krieger, H.U., Piskorski, J., Schäfer, U., and Xu, F. (2004) Shallow Processing with Unification and Typed Feature Structures – Foundations and Applications. *Künstliche Intelligenz*, **1**, 17–23.

Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C.D. and Stamatopoulos, P. (2000) Rule-Based Named Entity Recognition For Greek Financial Texts. In: *Proc. of the Workshop on Computational Lexicography and Multimedia Dictionaries (COMLEX 2000*, Patras, Greece. 75–78.

Finkel, J.R. and Manning, Ch.D. (2009a) Joint Parsing and Named Entity Recognition. In: *Proc. of NAACL-2009*, Boulder, Colorado, USA. Association for Computational Linguistics, 326–334.

Finkel, J.R. and Manning, Ch.D. (2009b) Nested Named Entity Recognition. In: *Proc. EMNLP-2009*, Singapore. Association for Computational Linguistics, 141–150.

Freitas, C., Mota, C., Santos, D., Oliveira, H.G. and Carvalho, P. (2010) Second HAREM: Advancing the State of the Art of Named Entity Recognition in Portuguese. In: *Proc. of the Seventh conference on International Language Resources and Evaluation (LREC'10).* ELRA, 3630–3637.

Friburger, N. and Maurel, D. (2004) Finite-state transducer cascade to extract named entities in texts. *Theoretical Computer Science*, **313**, 94–104.

Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H. and Wilks, Y. (1995) University of Sheffield: Description of the LaSIE System as Used for MUC-6. In: *Proc. of the Sixth Message Understanding Conference (MUC-6).* NIST, Morgan-Kaufmann Publishers, 207–220.

Galicia-Haro, S.N. and Gelbukh, A. (2007) Complex named entities in Spanish texts. In: S. Sekine and E. Ranchhod, eds., *Named Entities. Recognition, classification and use.* John Benjamins, 71–96.

Galliano, S., Gravier, G. and Chaubard, L. (2009) The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts. In: *Proc. of Interspeech 2009.* International Speech Communication Association (ISCA), 2583–2586.

Głowińska, K. and Przepiórkowski, A. (2010) The Design of Syntactic Annotation Levels in the National Corpus of Polish. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC 2010.* ELRA, 1816–1821.

Graliński, F., Jassem, K. and Marcińczuk, M. (2009a) An Environment for Named Entity Recognition and Translation. In: *Proc. of the 13th Annual Conference of the EAMT*. European Association for Machine Translation, 88–96.

Graliński, F., Jassem, K., Marcińczuk, M. and Wawrzyniak, P. (2009b) Named Entity Recognition in Machine Anonymization. In: *Recent Advances in Intelligent Information Systems*. Exit, Warsaw, 247–260.

Hobbs, J.R., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M. and Tyson, M. (1997) FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In: E. Roche and Y. Schabes, eds., *Finite-state language processing*. MIT Press, 383–406.

Kravalová, J. and Žabokrtský, Z. (2009) Czech Named Entity Corpus and SVM-based Recognizer. In: *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, Suntec, Singapore. Association for Computational Linguistics, 194–201.

Kubiak-Sokół, A. and Łaziński, M., eds. (2007) *Słownik nazw miejscowości i mieszkańców* (in Polish). Wydawnictwo Naukowe PWN, Warszawa.

Lubaszewski, W. (2007) Information extraction tools for Polish text. In: *Proc. LTC'07, Poznań*. Wydawnictwo Poznanskie, Poznań, 567.

Lubaszewski, W. (2009) *Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu* (in Polish). AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne, Kraków.

Marcińczuk, M. and Piasecki, M. (2007) Pattern Extraction for Event Recognition in the Reports of Polish Stockholders. In: *Proc. IMCSIT–AAIA'07, Wisła, Poland*. Polskie Towarzystwo Informatyczne, 275–284.

Marcińczuk, M. and Piasecki, M. (2010) Named Entity Recognition in the Domain of Polish Stock Exchange Reports. In: *Proc. Intelligent Information Systems 2010, Siedlce, Poland*. Publishing House of University of Podlasie, 127–140.

Marciniak, M., Rabiega-Wiśniewska, J., Savary, A., Woliński, M. and Heliasz, C. (2009) Constructing an Electronic Dictionary of Polish Urban Proper Names. In: *Recent Advances in Intelligent Information Systems. Proceedings of the Balto-Slavonic Natural Language Processing Workshop, Kraków*. Exit, Warszawa, 743-749.

Mikheev, A., Moens, M. and Grover, C. (1999) Named Entity Recognition without Gazetteers. In: *Proc. of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*. Association for Computational Linguistics, Stroudsburg, USA, 1–8.

Mykowiecka, A., Marasek, K., Marciniak, M. and Rabiega-Wiśniewska, J. (2009) Annotated Corpus of Polish Spoken Dialogues. In: Z. Vetulani and H. Uszkoreit, eds., *Human Language Technology. Challenges of the Information Society, Third Language and Technology Conference, LTC 2007, Poznan, Poland*. **LNCS 5603**, 50–62.

Nadeau, D. and Sekine, S. (2007) A survey of named entity recognition and classification. *Linguisticae Investigationes*, **30**(1), 3–26.

Nouvel, D., Antoine, J.Y., Friburger, N. and Maurel, D. (2010) An Analysis of the Performances of the CasEN Named Entities Recognition System in the Ester2 Evaluation Campaign. In: N. Calzolari et al., eds., *Proc. of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. ELRA, 523–529.

Osenova, P. and Kolkovska, S. (2002) Combining the named-entity recognition task and NP chunking strategy for robust pre-processing. In: *Proceedings of the Workshop on Treebanks and Linguistic Theories, (TLT02)*, Sozopol, Bulgaria. Bulgarian Academy of Sciences, 167–182.

Piasecki, M. and Wardyński, A. (2006) Multiclassifier Approach to Tagging of Polish. In: *Proc. of the International Multiconference on Computer Science and Information Technology (IIS'06)*. Polskie Towarzystwo Informatyczne, 169–178.

Piskorski, J. (2005) Named-Entity Recognition for Polish with SProUT. *Proceedings of IMTCI 2004, Warsaw, Poland*. **LNCS 3490**, Springer, 122-133.

Piskorski, J. (2008) ExPRESS : extraction pattern recognition engine and specification suite. In: *Proc. of the International Workshop a Finite-State Methods and Natural Language Processing 2007 (FSMNLP'2007). Potsdam, Germany*, Universitaet Potsdam, 166–183.

Piskorski, J., Homola, P., Marciniak, M., Mykowiecka, A., Przepiórkowski, A. and Woliński, M. (2004) Information Extraction for Polish Using the SProUT Platform. In: S. Kłopotek, T. Wierzchoń and K. Trojanowski, eds., *Intelligent Information Processing and Web Mining*. Springer-Verlag, Berlin, 227–236.

Piskorski, J., Sydow, M. and Kupść, A. (2007) Lemmatization of Polish Person Names. In: *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies (ACL '07)*. Association for Computational Linguistics, Stroudsburg, USA, 27–34.

Piskorski, J., Wieloch, K. and Sydow, M. (2009) On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages. *Information Retrieval*, **12**(3), 275–299.

Przepiórkowski, A., Górski, R.L., Łaziński, M. and Pęzik, P. (2009) Recent Developments in the National Corpus of Polish. In: J. Levická and R. Garabík, eds., *Proc. of Slovko'09, Smolenice, Slovakia*. Tribun, Brno, 302–309.

Przepiórkowski, A. and Woliński, M. (2003) A Flexemic Tagset for Polish. In: *Proceedings of the 2003 EACL Workshop on Morphological Processing of Slavic Languages (MorphSlav '03)*. Association for Computational Linguistics, 33–40.

Rymut, K. (2002) *Dictionary of Surnames in Current Use in Poland at the Beginning of the 21st Century.* Polish Academy of Sciences, Polish Language Institute and Polish Genealogical Society of America, Kraków-Chicago.

Rymut, K., ed. (2008) *Nazwy wodne Polski.* Research project nr 1H01D01029 (electronic database), Polska Akademia Nauk, Instytut Języka Polskiego, Kraków.

Rzetelska-Feleszko, E., ed. (2005) *Polskie nazwy własne* (in Polish). Instytut Języka Polskiego Polskiej Akademii Nauk, Kraków.

Saloni, Z., Gruszczyński, W., Woliński, M. and Wołosz, R. (2007) *Słownik gramatyczny języka polskiego* (in Polish). Wiedza Powszechna, Warszawa.

Savary, A., Krstev, C. and Vitas, D. (2007) Inflectional Non Compositionality and Variation of Compounds in French, Polish and Serbian, and Their Automatic Processing. *BULAG*, **32**, 73–93.

Savary, A., Rabiega-Wiśniewska, J. and Woliński, M. (2009) Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex. **LNAI 5070**, Springer, 111–142.

Savary, A., Waszczuk, J. and Przepiórkowski, A. (2010) Towards the Annotation of Named Entities in the Polish National Corpus. In: N. Calzolari et al., eds., *Proc. of the Seventh conference on International Language Resources and Evaluation (LREC'10).* ELRA, 3622–3629.

Schäfer, U. (2006) OntoNERdIE—Mapping and Linking Ontologies to Named Entity Recognition and Information Extraction Resources. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006).* ELRA, 1756–1761.

Sekine, S., Sudo, K. and Nobata, Ch. (2002) Extended Named Entity Hierarchy. In: *Proc. of the 3rd International Conference on Language Resources and Evaluation, Canary Island, Spain.* ELRA, 3622–3629.

Wacholder, N., Ravin, Y. and Choi, M. (1997) Disambiguation of proper names in text. In: *Proc. of the Fifth Conference on Applied Natural Language Processing (ANLC '97).* Association for Computational Linguistics, Stroudsburg, USA, 202–208.

Waszczuk, J., Głowińska, K., Savary, A. and Przepiórkowski, A. (2010) Tools and Methodologies for Annotating Syntax and Named Entities in the National Corpus of Polish. In: *Proc. of IMCSIT-CLA'10 Workshop*, Wisła, Poland. Polskie Towarzystwo Informatyczne, 531–539.

Wolinski, F., Vichot, F. and Dillet, B. (1995) Automatic Processing of Proper Names in Texts. In: *EACL '95: Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics.* Morgan Kaufmann Publishers Inc., 23–30.

Woliński, M. (2006) Morfeusz – a Practical Tool for the Morphological Analysis of Polish. In: *Proc. of IIS:IIPWM'06.* Springer, 503–512.