Intelligent Information Systems
ISBN 978-83-7051-580-5, pages 141–154

Lexicons and Grammars for Named Entity Annotation in the National Corpus of Polish

Agata Savary[1,2] 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 initial results in 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 Polish named entities have been adapted to meet those requirements. We show first results of the corpus annotation using the information extraction platform SProUT.

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 the 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 corpus annotation of Polish NEs. Its results are expected to boost the automatic Named Entity Recognition (NER) 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.

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

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. This last dictionary converted to a gazetteer is largely used in the present project (see Sec. 4.2).

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 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, however 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 have drawn the list of 200 biggest Polish cities from this website.

[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`

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* (cf Sec. 4), that covers 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. Also in Lubaszewski (2007) and Lubaszewski (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

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

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 (cf Chinchor (1997)). Clearly, many units, if considered out of context, are potetially 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 Voivodeship' or to the historical region name `Mazowsze` 'Masovian Region'. Within the corpus most of such ambiguities could be solved so far.

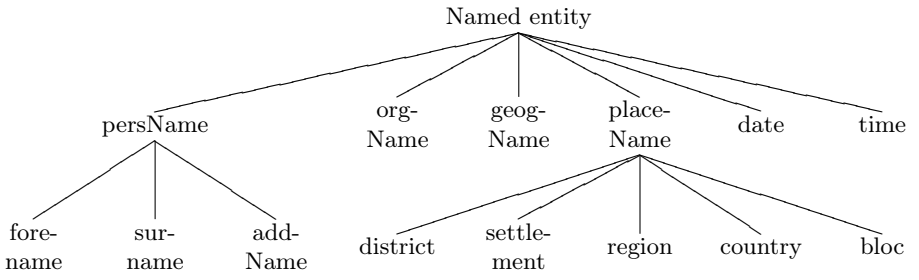

FIGURE 1: Type hierarchy of Polish NEs

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 however interested in some units that are less frequently covered by other projects, such as relative adjectives stemming from person, location and organization names, as well as names of inhabitants and organization members. These derived names form a typology which is vertical to the one in Fig. 1.

Traditional NER, MUC and CoNLL campaigns, has 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, 2009; 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:

(1) $[[\text{Maria}]_{forename}\ [\text{Skłodowska}]_{surname}\text{-}[\text{Curie}]_{surname}]_{persName}$

(2) $[\text{ulica}\ [[\text{Mikołaja}]_{forename}\ [\text{Kopernika}]_{surname}]_{persName}]_{geogName}$

(3) $[[\text{Wydział Prawa}]_{orgName}$
$[\text{Uniwersytetu}\ [\text{Warszawskiego}]_{relAdj:settlement(Warszawa)}]_{orgName}]_{orgName}$

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. The syntactic words, on 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 (e.g. *Stany Zjednoczone* for *Stanów Zjednoczonych* 'United States')
- TEI-P5-inspired normalization of date and time (e.g. *2009-10-30*, *09:45*)
- For derivations, the basic NEs they were derived from (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's why we put a special impact 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 multilingual 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 texts (Piskorski *et al.*, 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" (cf. Sec. 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 to name 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 data 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 of *SProUT* simply recognizes occurrences of gazetteer entries in text on the basis of static lexica. Thus, the 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 about 60,000 inflected forms of Polish and foreign first names, and 1,500 forms for Polish positions names (e.g. *poseł* 'depute', *generał* 'general').

These data have been completed by subsets of resources mentioned in Sec. 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 (cf. Sec. 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 Sec. 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 (4)-(5).

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 answering 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 (6).

(4) Buga | GTYPE:gaz_river | G_LEMMA:Bug | G_NUMBER:singular | G_CASE:gen | G_GENDER:masc3 | G_SOURCE:Wikipedia | G_INFL_SOURCE:Morfeusz

(5) Kowalskim | GTYPE:gaz_surname | G_LEMMA:Kowalski | G_NUMBER:singular | G_CASE:ins | G_GENDER:masc1 | G_SOURCE:Futrega | G_INFL_SOURCE:Morfeusz

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

(7) inspektorowi | GTYPE:gaz_position | G_LEMMA:Inspektor | G_CASE:dat | G_GENDER:masc1| G_NUMBER:singular

Some words which are not individual proper names themselves, can help in extracting NEs in texts. Some of such triggers words (also known as internal and external evidences, see Sec. 4.4) appeared in the gazetteers created by Piskorski (2005), and were completed in the current project, as shown in example (7). 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. 3.

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. 2. Parts of the hierarchy[10] 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[11] (lines 6–11). The type describing a gazetteer entry (lines 12-19)

[10]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'.

[11]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.

```
*avm*                    := *top*.
string                   :< *top*.
index-avm                := *avm*.
tokentype                := index-avm.
all_capital_word, lowercase_word, first_capital_word, ... :< tokentype.
part_of_speech, infl     :< *avm*.
adjective                :< part_of_speech.
infl_adjective := infl & [CASE_ADJECTIVE case, GENDER_ADJECTIVE gender,
                    NUMBER_ADJECTIVE number, DEGREE_ADJECTIVE degree].
case                     :< *avm*.
nom, gen, ...            :< case.
gtype                    := index-avm.
gaz_surname              :< gtype.
gaz_city                 :< gtype.
...
sign := *avm* & [SURFACE string, CSTART string, CEND string].
gazetteer := sign & [GTYPE gtype, LEMMA string, G_NUM_BASE string,
                     G_CASE case, G_GENDER gender, G_NUMBER number,
                     G_SOURCE string  G_INFL_SOURCE string, ... ].
agr-nkjp := *avm* & [NE_NUMBER number,NE_CASE case,NE_GENDER gender].
ne-nkjp-type             :< *top*.
place_name               :< ne-nkjp-type.
country                  :< place_name.
...
int_proof_geog_name      :< string.
"Góra", "Jezioro",...    :< int_proof_geog_name. ;;'Mountain, Lake'
ext_proof_geog_name      :< string.
"rzeka", "zatoka", ...   :< ext_proof_geog_name. ;;'river, bay'
ne-nkjp                  := sign &  [BASE string, NE_TYPE ne-nkjp-type,
                                     TREE string, MORPH agr-nkjp].
```

FIGURE 2: Extract of the Polish type hierarchy

Data category	Lemmas	Inflected Forms
First names	17,096	39,463
Family names	17,673	88,864
Organizations	1,733	1,865
Countries and regions	305	3,472
Cities	2,954	6,572
Rivers and mountains	362	2,003
Adjectives	1,871	128,424
Inhabitants	12,292	19,387
Trigger words	476	2,422
TOTAL	54,762	292,472

NE types	Rules
Persons	29
Organizations	20
Locations	25
Temporal expr.	24
Derivations	5
Auxiliary	17
TOTAL	121

FIGURE 3: Composition of the Polish SProUT gazetteers and NER grammars

inherits three attributes from the `sign` type, which are crucial for our grammars, as can be seen in Sec. 4.4: (i) `SURFACE`, which indicates the exact string occurring in the text (i.e. the surface realization of a concept), (ii) `CSTART` 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 (7). 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*$_M$ *ultiflex*, *Manual*).

Further we define a basic morphological structure for all NEs, which usually are nominal phrases (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 31–32). 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 Sec. 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.

```
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

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 (5) at position 127 in the input text would be `[ Kowalskim | Kowalski | forename | 127 | 135 | prio_1 ]`.

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

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*

`prio_3` by token-based rules[12]. Unfortunately, this mechanism does not handle ambiguities between names obtaining the same priority, e.g. when *Wisła* as a town and as a river obtain $\texttt{prio}_1$ 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.

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. Only the longest-match names are extracted. They 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 locations. 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 particular attention payed to determining the proper lemma for each recognized entity.

The current numbers of rules for NKJP covering various types of named entities are summarized in Fig. 3. The fact that our rules are by over 40 less numerous than the previous set is mainly due to a rather restricted scope of temporal expressions to be annotated in NKJP.

[12]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.

Persons		Locations		Organizations		Temporal expr.		Derivations		Overall	
0.85	0.69	0.76	0.52	0.71	0.14	0.75	0.67	0.9	0.63	0.88	0.61

FIGURE 7: Precision and recall of the NE recognition

5 First Evaluation

We have performed the first evaluation of our grammars on a small corpus of 1855 words (14 KBytes) containing 224 NEs. We considered that a NE is correctly recognized if all its properties have been correctly determined: (i) its left and right frontiers, (ii) its type, (iii) its lemma, (iv) its derivation type and base if any. In terms of corpus pre-annotation prior to manual correction, this corresponds to a case when no action is required from the human annotator. Of course partially correctly recognized NEs also help limit her intervention. The most important gain is obtained when the correct lemma has been found since correcting a lemma requires text editing instead of selecting a fixed value from a list.

The results in terms of precision and recall are given in Fig. 7. Unsurprisingly, the names of organizations are the most difficult to extract due to their often unpredictable naming conventions. If the figures for precision are reasonable, those for recall are sometimes unsatisfactory. Since a good recall seems crucial within corpus pre-annotation, we have already introduced some relaxed rules (e.g. annotating fore- and surnames when no context hints are given) that speed up the annotators work, even if the precision decreases. We plan a further examination of possible relaxed rules.

6 Conclusions and Perspectives

We have presented the named entity annotation task, which is a part of the ongoing project aiming at the 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 first results show that the human annotation can be substantially supported by the automatic pre-annotation.

As the human annotation progresses we hope to get an important feedback, and perform a more thorough analysis of errors that will guide us in the correction and creation of new gazetteer resources and grammar rules. We also wish to make a wider use of the three *SProUT* functionalities: (i) list types that might allow for a more efficient representation of embedded structures, (ii) functional operators for lemmatization of Polish names developed by Piskorski *et al.* (2007), (iii) variant recognition that would cover occurrences of names whose variants have been detected elsewhere in the text. We hope for a prompt integration of the *Morfeusz SGJP* analyzer (cf. Sec. 4.1), which would increase the lexicon coverage. Finally, having manually annotated the 1-million word NKJP subcorpus we plan to develop hybrid annotation based on rule-based and machine learning methods. This final tool will require evaluation with respect to other existing NER tools for Polish.

References

Witold Abramowicz, Agata Filipowska, Jakub Piskorski, Krzysztof Wecel, and Karol Wieloch (2006), Linguistic Suite for Polish Cadastral System., in *Proceedings of the LREC'06*, pp. 53–58, Genoa, Italy.

Piotr Bański and Adam Przepiórkowski (2009), Stand-off TEI Annotation: the Case of the National Corpus of Polish, in *Proceedings of the Third Linguistic Annotation Workshop (LAW III) at ACL-IJCNLP 2009*, pp. 64–67, Singapore.

Markus Becker, Witold Drożdżyński, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu (2002), SProUT - Shallow Processing with Typed Feature Structures and Unification, in *Proceedings of ICON 2002, Mumbay, India*.

Ivan Budiscak, Jakub Piskorski, and Strahil Ristov (2009), Compressing Gazetteers Revisited, in *Proceedings of the FSMNLP'09, Pretoria, South Africa*.

Lou Burnard and Syd Bauman, editors (2008), *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, Oxford.

Nancy Chinchor (1997), MUC-7 Named Entity Task Definition, in *Proc. of MUC-7*.

Jan Daciuk and Jakub Piskorski (2006), Gazetteer Compression Technique Based on Substructure Recognition, *Advances in Soft Computing*, 35/2006.

Witold Drożdżyński, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu (2004), Shallow Processing with Unification and Typed Feature Structures – Foundations and Applications, *Künstliche Intelligenz*, 1/04.

Jenny Rose Finkel and Christopher D. Manning (2009a), Joint Parsing and Named Entity Recognition, in *Proceedings of NAACL-2009*, Boulder, Colorado, USA.

Jenny Rose Finkel and Christopher D. Manning (2009b), Nested Named Entity Recognition, in *Proceedings of EMNLP-2009*, Singapore.

Sofía N. Galicia-Haro and Alexander Gelbukh (2009), Complex named entities in Spanish texts, *Named Entities*, pp. 71–96.

Katarzyna Głowińska and Adam Przepiórkowski (2010), The Design of Syntactic Annotation Levels in the National Corpus of Polish, in *Proc. of LREC'10, to appear*.

Filip Graliński, Krzysztof Jassem, and Michał Marcińczuk (2009a), An Environment for Named Entity Recognition and Translation, in *Proceedings of the 13th Annual Conference of the EAMT*, pp. 88–96, Barcelona.

Filip Graliński, Krzysztof Jassem, Michał Marcińczuk, and Paweł Wawrzyniak (2009b), Named Entity Recognition in Machine Anonymization, in *Recent Advances in Intelligent Information Systems*, pp. 247–260, Exit, Warsaw.

Jana Kravalová and Zdeněk Žabokrtský (2009), Czech Named Entity Corpus and SVM-based Recognizer, in *Proceedings of ACL-NEWS'09 Workshop*, Singapore.

Aleksandra Kubiak-Sokół and Marek Łaziński, editors (2007), *Słownik nazw miejscowości i mieszkańców*, Wydawnictwo Naukowe PWN, Warszawa.

Wiesław Lubaszewski (2007), Information extraction tools for Polish text, in *Proc. of LTC'07, Poznań, Poland*, Wydawnictwo Poznanskie, Poznań.

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

Michal Marcińczuk and Maciej Piasecki (2007), Pattern Extraction for Event Recognition in the Reports of Polish Stockholders, in *Proceedings of IMCSIT–AAIA'07, Wisła, Poland*, pp. 275–284.

Małgorzata Marciniak, Joanna Rabiega-Wiśniewska, Agata Savary, Marcin Woliński, and Celina Heliasz (2009), Constructing an Electronic Dictionary of Polish Urban Proper Names, in *Recent Advances in Intelligent Information Systems*, Exit.

A. Mykowiecka, M. Marciniak, and J. Rabiega-Wiśniewska (2008), Proper Names in Polish Dialogs, in *Proceedings of the IIS 2008 Workshop on Spoken Language Understanding and Dialogue Systems*, Springer Verlag, Zakopane, Poland.

Petya Osenova and Sia Kolkovska (2002), Combining the named-entity recognition task and NP chunking strategy for robust pre-processing, in *Proceedings of the Workshop on Linguistic Theories and Treebanks*, Sozopol, Bulgaria.

J. Piskorski, M. Sydow, and A. Kupść (2007), Lemmatization of Polish Person Names, in *ACL 2007. Proc. of the Workshop on Balto-Slavonic NLP 2007.*

Jakub Piskorski (2005), Named-Entity Recognition for Polish with SProUT, in *LNCS Vol 3490: Proceedings of IMTCI 2004, Warsaw, Poland.*

Jakub Piskorski, Petr Homola, Małgorzata Marciniak, Agnieszka Mykowiecka, Adam Przepiórkowski, and Marcin Woliński (2004), Information Extraction for Polish Using the SProUT Platform, in *Proceedings of IIS'04, Zakopane, Poland.*

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

Adam Przepiórkowski, Rafał L. Górski, Marek Łaziński, and Piotr Pęzik (2009), Recent Developments in the National Corpus of Polish, in Jana Levická and Radovan Garabík, editors, *Proceedings of Slovko'09, Smolenice, Slovakia*, Tribun, Brno.

Adam Przepiórkowski and Marcin Woliński (2003), A Flexemic Tagset for Polish, in *Proc. of the Workshop on Morphological Processing of Slavic Languages, EACL 2003.*

Kazimierz Rymut (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.

Kazimierz Rymut, editor (2008), *Nazwy wodne polski*, Research project nr 1H01D01029 (electronic database), Polska Akademia Nauk, Instytut Języka Polskiego, Kraków.

Ewa Rzetelska-Feleszko, editor (2005), *Polskie nazwy własne*, Instytut Języka Polskiego Polskiej Akademii Nauk, Kraków.

Zygmunt Saloni, Włodzimierz Gruszczyński, Marcin Woliński, and Robert Wołosz (2007), *Słownik gramatyczny języka polskiego*, Wiedza Powszechna, Warszawa.

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

Agata Savary, Joanna Rabiega-Wiśniewska, and Marcin Woliński (2009), Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex, *LNAI*, 5070.

Agata Savary, Jakub Waszczuk, and Adam Przepiórkowski (2010), Towards the Annotation of Named Entities in the Polish National Corpus, in *Proc. of LREC'10, to appear.*

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata (2002), Extended Named Entity Hierarchy, in *Proceedings of LREC'02*, Canary Island, Spain.

Marcin Woliński (2006), Morfeusz – a Practical Tool for the Morphological Analysis of Polish, in *Proceedings of IIS:IIPWM'06*, pp. 503–512, Springer.