# Building a Morphosyntactic Lexicon and a Pre-syntactic Processing Chain for Polish

Benoît Sagot

# Building a morphosyntactic lexicon and a pre-syntactic processing chain for Polish

Benoît Sagot[1,2]

[1] ALPAGE** — INRIA Paris-Rocquencourt & Université Paris 7
Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay cedex, France
[2] Instytut Podstaw Informatyki Polskiej Akademii Nauk (IPI PAN)
ul. J.K. Ordona 21, 01-237 Warszawa, Poland
benoit.sagot@inria.fr

**Abstract.** This paper introduces a new set of tools and resources for Polish which cover all the steps required to transform a raw unrestricted text into a reasonable input for a parser. This includes (1) a large-coverage morphological lexicon, developed thanks to the IPI PAN corpus as well as a lexical acquisition techique, and (2) multiple tools for spelling correction, segmentation, tokenization and named entity recognition. This processing chain is also able to deal with the XCES format both as input and output, hence allowing to improve XCES corpora such as the IPI PAN corpus itself. This allows us to give a brief qualitative evaluation of the lexicon and of the processing chain.

## 1 Introduction

In recent years, a considerable effort has been made towards efficient and robust surface processing of large corpora for various tasks such as information extraction and retrieval, linguistic information acquisition, grammar induction, and others. However, this effort has been mostly focused on a few major languages, notably English. Less effort has been made on most other languages.

This paper concentrates on Polish, one of the Slavonic languages for which resources and tools do exist, although much less than for, e.g., Czech. Indeed, [1] introduces a rule-based named-entity recognition system for Polish built on top of the NLP plateform SProUT [2]. As regards linguistic resources, which are needed for the construction and/or acquisition of linguistic processing chains, [3] presents the results of projects POLEX, CEGLEX and GRAMLEX, which consitute, among others, a morphological resource for Polish. But this resource is not freely available. On the contrary, the IPI PAN corpus of Polish [4], which is morphologically annotated, is publicly available.

Therefore, this corpus is a valuable starting point for developing NLP tools and resources for Polish. This paper describes the two first steps of a long-term program, namely the development of a morphlogical lexicon and of a pre-parsing processing chain. The following step, the development of a phrase-level

** The work described in this paper has been carried out when the author was a member of the SIGNES team of INRIA, during a 3-month stay at IPI PAN.

parser, is ongoing. It should be followed by a syntactic lexicon (which is to be acquired thanks to results provided by the phrase-level parser, thanks to techniques already presented in [5]) and, finally, a deep parser.

The work presented here can be considered as the application and adaptation to Polish of a set of tools that have been initially developed for French. We first discuss the construction of a baseline morphological lexicon for Polish from the IPI PAN data, then techniques to improve this lexicon, and finally the development of a robust pre-parsing processing chain, SxPipe-pl. We sketch how these results already enabled us to improve the IPI PAN corpus, which could lead in a near future to a new version of the corpus.

## 2 A baseline Polish morphological lexicon

### 2.1 Lexical framework

An NLP lexicon has to represent several kinds of information: morphological, syntactic, and possibly semantic. However, there are different ways to model such a rich information, and in particular different levels of information factorization. We call **extensional lexicon** a resource that associates with each *form* a detailed structure that represents all this information. Such a lexicon is typically used by parsers. We call **intensional lexicon** a resource that factorizes the information, by associating with each *lemma* a morphological class and deep syntactic information. In [6], the authors sketch a framework named Alexina that implements this two-level vision of lexical information, and introduce the Le*fff*, a large-coverage syntacic lexicon for French which relies on (an more recent version of) this framework.

An intensional entry, i.e., an entry of the intensional lexicon, is defined as a triple of the form *(lemma, morphological class, deep syntactic information)*. An extensional entry, i.e., an entry of the extensional lexicon, is a triple of the form *(inflected form, category, surface syntactic information)*, where the syntactic structure includes the lemma, morphological information, the sub-categorization frame (when relevant), and other syntactic features. However, since we do not consider syntactic information in this paper, both the intensional and the extensional lexicons are simplified: the *compilation* process which transforms an intensional lexicon into its extensional counterpart is mostly an *inflection* process. Moreover, both lexicons are simplified: an intensional entry becomes a couple *(lemma, morphological class)* and an extensional entry a triple *(form, lemma, (morphological) tag)*. We call *morphological lexicon* a set of such simplified extensional entries which only represent morphological information. The inflection process relies on a formalized morphological description of the language, i.e., a definition of all morphological classes.

In the remainder of this section, we show how we extracted directly from the IPI PAN corpus a baseline morphological description of Polish. In Section 3, we show how we extended this baseline, thanks, in particular, to an automatic lexical information acquisition technique.

### 2.2 Extracting a morphological lexicon from the IPI PAN corpus

When starting from a morphosyntactically annotated corpus, the most direct way to build a lexicon is to extract directly the triples *(form, lemma, tag)* that are attested in the corpus. This can be seen as a simplified version of an extensional lexicon. It is simplified because the syntactic information is virtually absent, and because for a given lemma, only the forms that are attested in the corpus are present in the lexicon. Although this step could seem trivial, is does raise several problems.

Our work is based on the IPI PAN corpus [4]. The IPI PAN corpus is a large (over 250 million words) morphosyntactically annotated and publicly available corpus of Polish. It has been developed for several years by the Linguistic Engineering Group at the Instytut Podstaw Informatyki (IPI) of the Polska Akademia Nauk (PAN). The morphosyntactic annotation has been obtained automatically, thanks to a morphological analyser named `Morfeusz` [7, ch. 4][3] and a disambiguator that has been trained on a manually annotated subset of the corpus [8]. It is encoded in a specific variant of the XCES format [9].

```
<tok>
<orth>Chciał</orth>
<lex disamb="1"><base>chcieć</base><ctag>praet:sg:m1:imperf</ctag></lex>
<lex><base>chcieć</base><ctag>praet:sg:m2:imperf</ctag></lex>
<lex><base>chcieć</base><ctag>praet:sg:m3:imperf</ctag></lex>
</tok>
```

**Table 1.** Example of a token in the IPI PAN corpus (XCES format).

It seems easy to extract a morphological lexicon from such a corpus, excluding of course unknown words (tokens tagged `ign`). For example, from the token of Table 1, one wordform can be infered (*chciał*) for which three (morphological) entries can be extracted: *(chciał, chcieć, praet:sg:m1:imperf)*, *(chciał, chcieć, praet:sg:m2:imperf)* and *(chciał, chcieć, praet:sg:m3:imperf)*.

However the IPI PAN corpus suffers from a light over-simplification of its annotation: all lemmas are lowercase, including proper nouns and other lemmas that should be capitalized. For example, in the corpus, the form *Warszawa* has *warszawa* as a lemma (i.e., as `base` attribute of the `lex` element). To (imperfectly) solve this problem, we developed simple heuristics to identify proper nouns at a lemma level. It is important to be able to identify those words which are both a proper noun and a common noun (`subst`) or an adjective (`adj`) (cf.

[3] Cf. `http://nlp.ipipan.waw.pl/~wolinski/morfeusz/`. It is important to state here the fact that the lexicon on which `Morfeusz` is based is *not* publicly available. If it were, the work of this section would be strongly simplified, since only capitalization and unknown word problems would remain.

Łódź, the city, vs. łódź, *boat*).[4] Results are satisfying, although specific problems remain for words that frequently occur in capitalized phrases (*Atlantycki*, *Demokratyczny*).

At this point, we have a baseline morphological Polish lexicon. It is a starting point both in terms of quality and coverage. It contains 865,673 entries representing 233,099 different wordforms (e.g., we have seen that the wordform *chciał* corresponds to 3 different entries). The aim of the next section is to go beyond this baseline.

## 3 Improving the baseline lexicon

In order to improve the quality and coverage of this baseline lexicon, we decided to extend it thanks to an automatic acquisition technique, as sketched below. This technique relies on the availability of a morphological description of the language. Therefore, we first describe the morphological formalism and the (partial) morphological description of Polish that we used. We show how this description allows us to detect annotation errors in the corpus as well as extending the baseline lexicon.

### 3.1 Morphological formalism

A morphological description of a language should have four main goals: optimal factorization of the information, readability and maintainability, coverage and accuracy, and ability to be used by a morphological compiler to generate automatically both an inflection tool (from a lemma to its forms) and a (non-deterministic) lemmatization tool (from a form to all its possible lemmas, restricted or not to lemmas which are known in a lexicon).

As part of the lexical framework described in [6], such a formalism and the associated morphological compiler have been already developed and applied to French as well as Slovak [10]. The formalism, which shares some ideas with the DATR formalism [11], relies on the following scheme:

- A set of morphological (inflection) classes which can inherit (partly or completely) from one another,
- Each class contains a set of forms represented as suffixes that are to be added to the stem,
- Forms can be controlled by tests over the stem (a given rule can apply only if a given regular expression matches the stem and/or if another one does not match the stem, and so on),

[4] We used approximately the following heuristics: (1) Any lemma which is not a `subst` or an `adj` is not a proper noun (2) Any lemma whose corresponding raw tokens (its `orth` elements) start with a capital letter more that 50% of all cases exists as a proper noun, (3) Any lemma whose corresponding raw tokens (its `orth` elements) start with a capital letter more that 99% of all cases is only a proper noun.

– Forms can be controlled by "variants" of the classes (e.g., one or more form can be selected by one or more flag which complements the name of the class),
– "Collision patterns" allow to link the surface form to the sequence *stem_suffix*.

To illustrate this, Table 2 show examples of collision patterns in our morphological description of Polish described below (3.2). Table 3 shows an extract of the inflection class for m1 (personal-masculine) substantives.

```
<letterclass name="hard"
             letters="b p f w m n ł t d r s z ch h"/>
...
<collision source="r_'" target="rz_"/>
<collision source="[:soft:]_y" target="[:soft:]_i"/>
<collision source="[:kg:]_e" target="[:kg:]_ie" final="+"/>
...
```

**Table 2.** Morphological formalism: example of "letter" classes and of collision patterns. In a collision pattern, the underscore sign denotes the boundary between the stem and the suffix. A rule is applied from the "source" to the "target" when inflecting, and from the "target" to the "source" when lemmatizing

```
<class name="subst-m1" tag_suffix=":m1" stems="...*">
    <form suffix="" tag="sg:nom"/>
    <form like="sg:gen" tag="sg:acc"/>
    <form suffix="a" tag="sg:gen" except="(wol|bawol)"/>
    <form suffix="u" tag="sg:gen" stems="(wol|bawol)"/>
    <alt>
        <form suffix="owi" tag="sg:dat" var="Dowi"/>
        <form suffix="u" tag="sg:dat" var="Du"/>
    </alt>
    <form suffix="em" tag="sg:inst"/>
    <form suffix="'e" tag="sg:loc" stems="..*[:hard:]"
                          except="(syn|dom|pan)"/>
    <form suffix="u" tag="sg:loc" except="..*[:hard:]"/>
    <form suffix="u" tag="sg:loc"
                stems="(syn|dom|pan|bor)"/>
...
```

**Table 3.** Example of a morphological class

### 3.2 Description of Polish nouns and adjectives

In order to prepare the construction of an intensional lexicon of Polish and to expand the lexicon so as to lower the percentage of unknown words (`ign` tokens), we developed a morphological description of Polish in the formalism sketched above. Its main linguistic basis is [12]. Currently, our morphological description covers adjectives and common nouns:

- 1 class for adjectives, plus 2 other classes (comparatives and superlatives) that exactly inherit from the standard class and are here for technical reasons[5]
- 10 classes for substantives: *m1*, *m1a* for *m1* substantives in *-a*, *m2* which inherits from *m1* and redefines pl:nom, pl:voc and pl:acc, *m3* which inherits from *m2* and redefines sg:acc and sg:gen, class *n* (neutral), class *num* for neutrals in *-um* (inherits from class *n*, all singular forms in *-um*, pl:gen in *-ów*), classes *nen* and *net* respectively for types *ramię/ramiona* and *cielę/cielęta*, class *fv* for feminine substantives in *-a* or *-i* and class *fc* for feminine substantives with a zero ending for sg:nom.

### 3.3 Detecting annotation errors in the corpus

Our morphological description of Polish is currently limited to nouns and adjectives. Its precision and coverage already enables us to detect some errors in the annotated corpus. Indeed, any nominal or adjectival form which is in the morphological lexicon (i.e., which was found in the corpus) must be analysable by the ambiguous lemmatizer with the appropriate category, tag and lemma.

Indeed, we were able to discover some errors, including systematic ones, in the IPI PAN corpus. Of course, these errors are reproduced as such in the baseline lexicon, from which they had to be removed. Some of them come from the automatic annotation tool, `Morfeusz`, and/or its underlying lexical database,[6] whereas others come from tokenization and related problems, as we shall see in Section 4

### 3.4 Automatic extension of the lexicon

In [10], the author describes a technique to acquire automatically lexical information from a raw corpus and a morphological description of the language. It has been applied to French verbs and to all open categories of Slovak. The

[5] They assign tags in `-comp` or `-sup` instead of `-pos`, so as to match the IPI PAN corpus tagset.

[6] A few examples: (1) sg:acc (and sg:gen) of m1, except *wół* and *bawół*, is in *-a*; however, many m1 forms ending in `-u` are tagged as sg:acc and sg:gen in the corpus (*aptekarzu*, *energetyku*, *kierowniku*, *laiku*,...); (2) pl:acc for m1 is identical to pl:nom; however, a huge amount of m1 in *-a* (*archiwista*, *finansista*,...) have forms in *-y* that are tagged as pl:acc (*archiwisty*, *finansisty*, whereas pl:acc forms are *archiwistów*, *finansistów*); (3) Some relatively frequent isolated problems.

availability of the morphological description of Polish allowed us to use this technique to extend automatically (with manual validation) our Polish lexicon so as to minimize as much as possible the amount of unknown words in the IPI PAN corpus (`ign` tokens).

The idea underlying this automatic lexical acquisition technique is the following: First, we use the ambiguous lemmatizer generated from the morphological description: we build all hypothetical lemmas that have at least one inflected form attested in the corpus. Then, we inflect these lemmas and rank them according to their likelihood given the corpus (fix-point algorithm); Many kinds of information are taken into account (derivational morphology, prefixes, frequency of tags depending on the category,...). Afterwards, manual validation is performed on the best-ranked hypothetical lemmas, thanks to an easy-to-use web interface. Finally, the whole process is launched anew, and benefits from the manual validation step (this loop is repeated as many times as necessary). For details, see [10].

Thanks to Radoslaw Moszczynski and Adam Przepiórkowski, who performed the manual validation, a few hours proved enough to acquire 1,460 validated lemmas (only nouns, adjectives and adverbs derived from adjectives). Moreover, a quick study of unkown words in the corpus allowed to add manually 46 lemmas and 186 so-called "manual forms", mostly abbreviations of (forms of) already existing lemmas.

Let us consider all `ign` tokens of the *law* sub-corpus of the IPI PAN corpus, on which we performed this automatic lexical acquisition process (over 3 million `ign` tokens out of 75 million tokens). As we will see in the next section, an appropriate pre-processing step can eliminate, among others, several tokenization and "named-entity" problems. We apply a simplified version of this pre-processing step, *without* spelling error correction and built *before* this lexicon extension process, so as to eliminate problems that are not linked with the incompleteness of the lexicon. We also eliminate all `ign` tokens which contain a capital letter. The result includes a lot of spelling errors, hence the following result is underestimated: the 1,460 validated lemmas, acquired and validated in only a few hours, cover almost 56% of the remaining occurrences of unknown words, which is a very satisfying result.

The resulting lexicon has 929,184 entries for 243,330 different wordforms. It is freely available under the Cecill-C (LGPL-compatible) license on the web site of the Alexina framework.[7]

## 4 Pre-parsing processing: a Polish SxPipe

Current parsers, both shallow and deep, are able to deal with large corpora. However, parsers often rely on lexicons and grammars designed to deal with "correct" language, which differs significantly from what can be found in real-life corpora. Hence pre-parsing processing methods are required to turn real-life

[7] http://alexina.gforge.inria.fr/

corpora into acceptable parser inputs. This pre-parsing step is not as basic as it could seem, in particular because it has to be very robust and non-deterministic. This is the goal achieved by the pre-parsing processing chain SxPipe [13, 14], developed initially for French.

We decided to develop a Polish version of SxPipe for two different reasons: first, many errors in the IPI PAN corpus do come from an imperfect pre-processing; second, a Polish SxPipe is a necessary step before developing a Polish parser, which is one of our future objectives.

### 4.1 SxPipe

In [13, 14], the authors present SxPipe, a set of tools which performs several tasks, which can be grouped into three categories:

- "named entities" (n.e.) recognition: pre-tokenization n.e. (URLs, emails, dates, addresses, numbers,...), lexicon-aware n.e. (phrases in foreign languages,...), and multi-words n.e. (numbers in full text, proper nouns...);
- tokenization and segmentation in sentences;
- (possibly) non-deterministic spelling error correction and multi-words identification (incl. re-accentuation and re-capitalization) with TEXT2DAG.

TEXT2DAG relies on an efficient spelling correction module, named SxSpell. Ontop of this module, TEXT2DAG performs sophisticated non-deterministic heuristics to segment and/or re-glue tokens into forms and to identify multi-token forms ("compound words"). Of course, both tasks strongly interact (in a quite complicated way) with the spelling correction proper.

### 4.2 A Polish version of SxPipe

Some of SxPipe modules are partly language-dependent. E.g., most "named entities" recognition tools had to be adapted and extended, because there are language-specific ways to say most things covered by named entities (addresses, dates, times...). Spelling correction rules used by SxSpell are partly encoding-specific ($s$ vs. $\acute{s}$,...) and partly language-specific ($\dot{z}$ vs. $rz$,...). However, once these adaptations are done, tokenization, spelling and multi-token identification tools just needed to be linked with the Polish (morphological) lexicon.

Moreover, SxPipe has been extended so as to deal, in input and output, with the XCES format used in the IPI PAN corpus, which includes all meta-textual information (XML content), morphological information on tokens (both ambiguous morphogical analysis and morphological disambiguation), token-boundary information (presence or not of a white space between two tokens), and others. All this information had to be preserved throughout the processing chain and restored in the output (when no correction applied), which was not possible in the previous version of SxPipe. On the other hand, some components used in the original French SxPipe have been adapted but are not used in the default configuration of SxPipe-pl, because they introduce information which has

proven irrelevant for improving the IPI PAN corpus (e.g., sequences of the form *acronym (acronym expansion)*, and others).

This work resulted in an XCES-compatible SxPipe available for three different languages: SxPipe for French, SxPipe-pl for Polish, and a very preliminary SxPipe-sk for Slovak. Since then, other versions of SxPipe have been developed for English, Spanish and Italian. All these tools are freely available.[8] The list of modules used in SxPipe-pl is shown in Table 4.

| (conversion from XCES to internal format) |
|---|
| e-mail addresses recognition |
| URLs recognition |
| dates recognition |
| phone numbers recognition |
| times recognition |
| postal adresses recognition |
| smileys, other special punctuation and oral transcripts marks recognition |
| numerical prefixes recognition |
| numbers and (numerical/symbolic) list markers recognition |
| (embedded n.e. removal) |
| tokenization and segmentation |
| TEXT2DAG: non-deterministic multi-word identification, tokenization correction and spelling error correction |
| recognition of numbers in full text |
| proper nouns identification |
| (conversion from internal format to XCES) |

**Table 4.** Sequence of components used by SxPipe-pl in its default configuration

### 4.3 Tokenization, spelling, and named entities problems in the corpus

As said above, the IPI PAN corpus contains a non-negligible proportion of unknown words, i.e., `ign` tokens — e.g., in the 75-million-token *law* sub-corpus, 3 million (4%) of tokens are `ign`. Some of these tokens are really words that are unknown from `Morfeusz`, and which have to be added to the lexicon, as previously described.

However, in order to identify these "real" unknown words as well as directly improve the corpus, all other sources of problems in the original corpus have to be identified. Hence the use of SxPipe-pl, whose impact on 430,924 tokens of the *law* subcorpus is summed up in Table 5.[9]

[8] `http://gforge.inria.fr/projects/lingwb/`

[9] Precision and recall measurements still need to be performed. Manual obervation of the results lead to the following conclusion: all modules but the spelling correction

The most frequent problems in the corpus that are detected and solved by SxPipe-pl are the following:

- The "special double-quote"[10] tokenization-based errors;
- "Named entities", especially numbers and proper nouns (tokens starting with a capital letter);
- Productive prefixes (e.g., wielko-, post-, agro-, anty-,...);
- Spelling errors (e.g.: *aberacji* (*aberracji*), *abmasadora* (*ambasadora*), *abowiem* (*albowiem*), *abp* (*aby*), *abrbitralności* (*arbitralności*), *absolutniej* (*absolutnej*)...).

Of course, the last important source of `ign` tokens in the original IPI PAN corpus are abbreviations (manually added in the lexicon) and "real" unknown words (e.g.: *abolicjonistycznej*, *abonamencka*, *aborcyjna*, *abortera*, *absolutoryjny*...). We have previously shown how to extend the lexicon so as to decrease the importance of this problem.[11]

## 5 Conclusions and perspectives

We have introduced the morphological lexicon for Polish we have developed, based on the IPI PAN corpus annotations and extended thanks to an automatic lexical acquisition technique. We also introduced SxPipe-pl, a full-featured pre-syntactic processing chain for Polish.

Our long-term objective is to develop a phrase-level LFG grammar and the associated parser (which will take as input the output of SxPipe-pl), so as to enable the automatic acquisition of syntactic information from the output of this parser (sub-categorization frames,...), using techniques evoked in [5]. This will lead to a full syntactic lexicon of Polish, which is a necessary step before the development of a robust large-coverage (LFG) parser for Polish.

## References

1. Piskorski, J.: Automatic named-entity recognition for Polish. In: International Workshop on Intelligent Media Technology for Communicative Intelligence, Warsaw, Poland. (2004)

module have extremely high precision and recall. The spelling correction module introduces a bit of noise because it sometimes manages to correct tokens which are unknown but correct.

[10] There are two different Unicode double-quote-like characters; only the usual one was recognized by the original tokenizer, hence many erroneous `ign` tokens such as *˝aby*.

[11] A non-trivial problem remains: the corpus is a sequence of tokens with associated morphological interpretations, whereas SxPipe's output is a graph of forms (defined as atomic units for a subsequent parser), with associated token-level anchors. But tokens and forms do not always correspond directly. Cf. for example *piątek 10.5.90 r.* — considered by SxPipe-pl as one (special) form, namely _*DATE* —, *po prostu*, *bez mała*, *niespełna*, *naprawdę*, *szliśmy*, *dwakroć*,... We solved this problem by introducing special XML elements to identify complex forms (`<sw>`, i.e., "syntactic words"). The description of the underlying mechanism and linguistic decisions is beyond the scope of this paper.

| Token kind | #tok | wrt all | wrt `ign` |
|---|---|---|---|
| `ign` tokens | 17,913 | **4.2%** | *100%* |

(a) Original corpus

| | | | |
|---|---|---|---|
| automatically acquired words | 453 | 0.1% | 2.5% |
| manually added words | 293 | 0.1% | 1.6% |
| multi-word units | 447 | 0.1% | 2.5% |
| dates and times | 850 | 0.2% | 4.7% |
| numbers and numeric prefixes | 4,474 | 1.0% | 25.0% |
| list markers | 256 | 0.1% | 1.4% |
| "special double-quote" | 1,404 | 0.3% | 7.8% |
| productive prefixes | 35 | 0.0% | 0.2% |
| spelling errors | 812 | 0.2% | 4.5% |
| unknown proper nouns (capitalized words) | 4,144 | 1.0% | 23.1% |
| other (remaining) unknown tokens | 413 | **0.1%** | 2.3% |

(b) Corpus processed by SxPipe-pl

**Table 5.** SxPipe-pl on 430,924 tokens of the *law* subcorpus: a few figures (note that multi-word units and named entities involve several tokens, hence the discrepancy between the number of `ign` token in the original corpus and the sum of all situations found in the processed corpus)


2. Drożdżyński, W., Krieger, H.U., Piskorski, J., Schäfer, U., Xu, F.: Shallow processing with unification and typed feature structures — foundations and applications. Künstliche Intelligenz **1** (2004) 17–23
3. Vetulani, Z.: Electronic Language Resources for Polish: POLEX, CEGLEX and GRAMLEX. In: Second International Conference on Linguistic Ressources and Evaluation (LREC'2000), Athens (2000)
4. Przepiórkowski, A.: Korpus IPI PAN: Wersja wstępna / The IPI PAN Corpus: Preliminary version. IPI/PAN, Warsaw, Poland (2004)
5. Fast, J., Przepiórkowski, A.: Automatic extraction of Polish verb subcategorization: An evaluation of common statistics. In: Proceedings of LTC'05, Poznań, Poland (2005) 191–195
6. Sagot, B., Clément, L., de La Clergerie, E., Boullier, P.: The Le*fff* 2 syntactic lexicon for French: architecture, acquisition, use. In: Proceedings of LREC'06, Genova, Italy (2006)
7. Woliński, M.: Komputerowa weryfikacja gramatyki Świdzińskiego. PhD thesis, Instytut Podstaw Informatyki Polskiej Akademii Nauk (IPI PAN) (2004)
8. Piasecki, M., Godlewski, G.: Reductionistic, tree and rule based tagger for Polish. In aw A. Kł opotek, M., awomir T. Wierzchoń, S., Trojanowski, K., eds.: Intelligent Information Processing and Web Mining. Advances in Soft Computing. Springer-Verlag, Berlin (2006)
9. Ide, N., Bonhomme, P., Romary, L.: XCES: An XML-based encoding standard for linguistic corpora. In: Proceedings of LREC'00, Paris (2000)
10. Sagot, B.: Automatic acquisition of a Slovak lexicon from a raw corpus. In: Lecture Notes in Artificial Intelligence 3658 (© Springer-Verlag), Proceedings of TSD'05, Karlovy Vary, Czech Republic (2005) 156–163

11. Evans, R., Gazdar, G.: The DATR Papers: February 1990. Technical Report CSRP 139, University of Sussex, Brighton (1990)
12. Grappin, H.: Grammaire de la langue polonaise. Institut d'études slaves, Paris (1985)
13. Sagot, B., Boullier, P.: From raw corpus to word lattices: robust pre-parsing processing. Archives of Control Sciences **15** (2005) 653–662
14. Sagot, B., Boullier, P.: SxPipe 2: architecture pour le traitement pré-syntaxique de corpus bruts. Traitement Automatique des Langues **49** (2008) (to appear).