Towards the Lemmatisation of Polish Nominal Syntactic Groups Using a Shallow Grammar*

Łukasz Degórski

Institute of Computer Science
Polish Academy of Sciences
ul. Ordona 21, 01-237 Warszawa, Poland

Abstract. While morphological analysers and taggers usually assign lemmata to wordforms, those tools focus on single words. For some tasks a tool that lemmatises (and thus normalises) whole phrases would be more appropriate. The paper presents, discusses and evaluates a set of tools to lemmatise nominal groups, based on a shallow grammar for Polish. The tools reach an overall success rate of over 58%, and almost 83% on the nominal groups that are correctly recognised by the grammar. The approach should be portable to other languages, especially those morphologically rich.

Keywords: lemmatisation, partial syntactic parsing, syntactic groups, nominal groups

1 Motivation

The task of finding lemmata of word forms is particularly important for morphologically rich languages, such as Polish. This is mostly dealt with by morphological analysers and taggers, as lemmatisation is an inherent, while not trivial, subtask (or side effect) of tagging. Nonetheless, all those tools focus on single words, while for some tasks, such as indexing, computing statistical measures like TF-IDF, and machine learning, a tool that lemmatises whole phrases would be useful to generate intuitively correct normalised forms.

A lemmatising engine was also directly needed for the CMS designed to manage and publish multilingual content, currently in development in the Applied Technology for Language-Aided CMS project (ATLAS; `www.atlasproject.eu`).

Note that in a synthetic, free word order language a lemma of the whole phrase is rarely a simple concatenation of lemmata of the components. For Polish, that may happen for instance for simple groups matching the pattern `Adj+Noun` or `Noun+Adj` if the adjective is masculine (*krwiożerczego potwora 'bloodthirsty +*

* The work reported here was carried out within the Applied Technology for Language-Aided CMS project co-funded by the European Commission under the Information and Communications Technologies (ICT) Policy Support Programme (Grant Agreement No 250467).

M+Gen monster + M+Gen'), for coordinations (*prezentacje i analizy 'presentations + Nom/Acc and analyses + Nom/Acc'* and also – by chance – for some forms unrecognised by the tagger. However, the simple concatenation will never work if the group contains, for instance, any non-masculine adjective/pronoun, or a nominal subgroup in genitive case:

`celem badania` '*aim+Inst+Sg research+Gen+Sg*'

should be lemmatised to

`cel badania` '*aim+Nom+Sg research+Gen+Sg*'

while the simple concatenation renders

`cel badanie` '*aim+Nom+Sg research+Nom+Sg*'.

The ongoing work in the National Corpus of Polish (NKJP[1]; `www.nkjp.pl`; see [7]) made it possible to deal with the task using a shallow grammar. An extensive grammar has been prepared for the Corpus, designed for the identification of various kinds of syntactic groups (among those – nominal) using the Spejd shallow processing tool (`nlp.ipipan.waw.pl/Spejd/`; [2]). The grammar has been handcrafted iteratively, using samples from a 1-million-word manually annotated subcorpus of the NKJP (see [3] for details).

Combining the lemmatisation task with shallow parsing has one great advantage - shallow parsing gives us structure, used as a base to write lemmatisation rules (or rather: schemata, as these are not rules in the sense of the grammar). The schemata are written separately for each rule of the grammar, and operate on the strings and structure matched by that rule.

2 Related work

Previously a similar task was attempted for Czech (Pala et al., [5]) for the law domain. The paper does not get into details on the method used and achieved results. There were also some attempts for Mongolian [4], but it seems that Khaltar and Fujii focused more on lemmatising single words and on loanwords in the Mongolian language.

Other phrase lemmatisation-related research focuses mostly on named entities (for instance, Piskorski et al., [6]), which is a different task involving different methodology.

3 Implementation – the processing chain

In our approach, the lemmatisation process can be divided into four main steps: tagging, shallow parsing, generating additional needed wordforms and final post-processing.

Note that the input needs not to be a simple list of nominal groups – it can be any Polish text. Identifying the groups in a running text is a part of the task of the shallow grammar.

[1] In Polish: Narodowy Korpus Języka Polskiego.

3.1 Tagging

The input (text file) needs to be morphologically annotated before Spejd grammars can be applied. For this task we used a Brill-based tagger called Pantera. The tagger outputs information in TEI P5-conformant format used in NKJP. See `http://code.google.com/p/pantera/tagger` and [1] for more information.

3.2 Shallow parsing

The main part of the lemmatisation process is applying the shallow grammar. The NKJP grammar mentioned before has been augmented with lemmatisation schemata for nominal groups, as well as for some adjectival groups, so that it is able not only to extract those groups, but also to assign them proper lemmata.

To do this in the Spejd formalism, we add a fourth parameter[2] to the `group()` operator in every relevant rule, as in the simple example below. The lemma of the whole group is constructed by smart concatenation of lemmata and orthographic forms of the constituents. The constituents, in turn, may also be syntactic groups – in which case their lemmata are results of similar operations performed at the earlier stages of parsing. Lemmata of single syntactic words (such as both nouns in the example below) are provided by the original NKJP grammar, based on the results of the tagging phase.

```
Rule    "NGk: Noun i Noun (koordynacja)"

Match:  [pos~"Noun"]
        [base~"i|oraz|ani|lub|albo|bądź|czy|a także" && pos~"Conj"]
        [pos~"Noun"];
Eval:   unify(case,1,3);

### Original NKJP grammar just marks the~group
#     group(NGk,1,1);
### We added the~4th parameter for~lemmatisation
      group(NGk,1,1,1.base " " 2.orth " " 3.base);
```

This particular rule recognises syntactic groups, consisting of two nouns with a conjunction in between. Both nouns must be unifiable for case. The lemma (called `base` in the Spejd formalism) of the conjunction must match one of the forms enumerated in the list, in addition to having a proper POS tag.

The lemma of this group is a concatenation of the lemma of the first noun, orthographic (unchanged) form of the conjunction and the lemma of the second noun.

Let's have a look at something more advanced:

[2] Only a new, prototype reimplementation of Spejd, still in development and not yet publicly available, supports syntactic groups lemmatisation and accepts this parameter.

```
Rule    "NGg: Noun + n-Noun w gen"

Match:  ([pos~"Noun" && case!~"gen"] | [type="NGa|NGk" && synh=[case!~gen]])
        ([pos~"Noun" && case~"gen"]
            | [type="NGa|NGk" && synh=[case~gen]]
            | [type="NGk" && semh=[case~gen]]
        )+;
### Original NKJP grammar
#Eval:   group(NGg,1,1);
### Added 4th parameter for~lemmatisation
Eval:   group(NGg,1,1,1.base " " 2.orth);
```

This rule recognises syntactic groups that

- begin with a single non-genitive noun, or a syntactic group of type `NGa` or `NGk`, whose syntactic head's case is not genitive
- followed by one or more of the following: noun in genitive; nominal group of type `NGa` or `NGk` whose syntactic head is a genitive; nominal group of type `NGk` whose semantic head is a genitive

An example of a group matching this rule is *Instytut Podstaw Informatyki Polskiej Akademii Nauk – 'Institute (of) Computer Science (of the) Polish Academy (of) Sciences'*. The lemma of such a group is the concatenation of the lemma of the first (non-genitival) part with the orthographic form of the remaining genitival part.

More information about the NKJP syntactic grammar and examples can be found in [8].

3.3 Generating forms

The tools we use (Pantera + Spejd) assign as lemma the nominative masculine singular positive form for adjectives and the nominative singular form for nouns. Perfectly simple in case of single words, it becomes more complicated in lemmatising multi-word expressions, such as nominal groups. For example,

`zielonej żabie` '*green+F+Dat+Sg frog+Dat+Sg*'

on a word-by-word basis would be lemmatised to

`zielony żaba` '*green+M+Nom+Sg frog+Nom+Sg*'[3],

while we expect `zielona żaba` '*green+F+Nom+Sg frog+Nom+Sg*' here.

For this reason we cannot always simply use the Spejd `.base` operator as in the example above. For adjectival groups, we protect the original gender information (extracted using the `.gender` operator) by returning a temporary string like `ADJ(zielony,F)` instead of just `zielony`. In such cases, the appropriate line in the grammar may look like this:

[3] In fact, the gender system in the tagset is more complicated, with multiple flavours of the masculine tag; however, for the sake of clarity of the examples, it is simplified to M,F,N here

```
group(NGa,2,2,"ADJ(" 1.base "," 1.gender ") " 2.base);
```

and render a "half-lemmatised" string like *ADJ(zielony,f) żaba*. These strings need to be converted later to proper forms. This is done using Morfeusz morphological analyser's (`http://sgjp.pl/morfeusz/`) wordform generation mode.

This implied deeper changes in the original NKJP grammar, as a lot of rules catch participles (imperfect and perfect) together with adjectives, whereas at the form generation level we need to treat them differently. Thus, some rules had to be multiplicated with various combinations of `ADJ`, `PPAS` and `PACT`.

3.4 Postprocessing

The final postprocessing deals with remaining simple problems that can be corrected on pure text level:

- all output is converted to lowercase for consistency, as some capital letters disappear during lemmatisation
- for some specific words two different forms (short and long, e.g. *me* and *moje*) are generated by Morfeusz; the short forms are removed using a regular expression

Technically this phase takes place together with form generation, in one Perl script. The script calls Morfeusz generator for each "half-lemmatised" string and processes its output.

4 Evaluation

For evaluation we used a subset of all nominal syntactic groups marked manually in the 1-million-word balanced subcorpus of the NKJP. From amongst almost 70000 we randomly chose a few hundred, and those have been manually lemmatised by a linguist. The linguist was instructed to skip groups that contain foreign names (Latin plant names, for instance) and names of people, unless they constituted only a small part of a longer phrase. After that we also removed a few groups for which the proper lemmatisation seemed very unclear, as we cannot expect the program to properly guess forms that even the linguists are not sure about.

In the end, 336 annotated phrases were left. They were divided into a development set of 112 and an evaluation set of 224. The development set was used to make final amendments to the shallow grammar, lemmatisation rules and postprocessing scripts. The program was then run with the final grammar and scripts on the unseen evaluation set.

The results were checked and divided into four categories:

1. program produced exactly the same result as the linguist (126)
2. grammar correctly recognised the group, but the produced lemma was different than the linguist's, and it was obviously incorrect (14)

3. grammar correctly recognised the group, but the produced lemma was different than the linguist's, however it was not obviously incorrect (18)
4. grammar incorrectly recognised the group, not giving a chance for proper lemmatisation of this group (66)

In case of any doubts, lemmata were classified to the third category. This category was later reviewed by another linguist, who marked 5 items as correct and 13 as indeed incorrect.

Thus, the evaluation results are as shown in Table 1.

Table 1. Results of the evaluation

Correct	131	58.5%
Group correctly recognised, bad lemma	27	12.0%
Group incorrectly recognised	66	29.5%
Total	224	

As a baseline we used a trivial algorithm that assigns the concatenation of lemmata of the constituents (single wordforms) as the lemma of the whole group. On the evaluation set it assigned 60 lemmata correctly (26.8%). It is worth noting that among those correctly lemmatised there were only 3 groups longer than 3 words.

The way of counting the success rate presented above may however be considered unfair to the algorithm, as it counts as an error not only wrong lemmatisation, but also wrong extraction of the nominal group (and that is, in fact, an error of the underlying grammar, not the lemmatisation patterns). It might be interesting to see how both the algorithm and the baseline performed on the subset of the evaluation set – those groups that have been correctly recognised by the grammar. In other words: to discard the 66 incorrectly recognised groups and look at the remaining 158 only. Table 2 shows these results.

Table 2. Comparison with baseline – with and without incorrectly recognised groups

	All groups in the eval set		Groups correctly recognised by the grammar	
Number of groups	224		158	
Correctly lemmatised by our alg.	131	**58.5%**	131	**82.9%**
Correctly lemmatised by baseline	60	26.8%	43	27.2%

5 Errors and potential improvements

As can be seen in the previous section, there is plenty room for improvement, but most of it is at the level of the group recognising grammar itself, and not at the lemma generation level, as the success rate on correctly recognised groups reaches almost 83%.

5.1 Group recognition

Among those 66 incorrectly recognised items, there are two basic types of error:

- catching only a part of the phrase as a nominal group
- catching a part of, or even the whole phrase, as two separate groups

It is worth noting that in many cases what has been marked as a group (a part or two separate parts of the input) is lemmatised correctly. In other words, the program correctly assigns lemmata to proper nominal groups being subsets of the provided input.

To solve this problem, we would have to use a completely different approach that takes into account the assumption that the whole input is a group and tries to match it top-down, while Spejd works bottom-up and makes no use here of the information that it is given a list of nominal syntactic groups (as opposed to free text).

In fact, using as input the manually extracted syntactic groups instead of whole sentences can make the results worse: the phrase *zielonej żabie* by itself is not lemmatised correctly, as it is not even recognised as a nominal group. However, it will be correctly marked and lemmatised in the sentence *Opowiedział historię zielonej żabie 'He told a story to a green frog'*.

Some errors, especially of the first type, can be corrected by adding specific rules to the grammar, such as a rule for dates, for ages (*40–letni '40-year-old'*, for groups with particular words (*zwłaszcza 'especially', tylko 'only'*) etc. This is clearly a room for easy, however laborious, improvement.

5.2 Lemmatisation

Taking a more detailed look into the 27 correctly assigned groups for which the generated lemma is incorrect, we recognise the following sources of errors:

1. correct lemmatisation would require semantic information and/or knowledge that the shallow grammar does not have, as in *życia osobistego proroka 'personal life of the prophet'* that has been lemmatised as if it was *'life of a personal prophet'*, an intepretation that is formally correct, but very unlikely
2. plurale tantum nouns are treated as normal nouns (arguably a special case of the above): *prawa człowieka 'human rights'*, *uczucia religijne 'religious feelings'*, while retaining the plural form in the lemma is expected

3. participles caught by general (not participle-specific) rules are, according to the conventions used in Pantera, lemmatised to infinitives; in most cases this is wrong
4. part of the lemma is lost due to incorrect dealing with lower-level groups in the grammar (cascading)

The last one is a technical issue that can be corrected, albeit with a lot of work. The third can also be corrected in the grammar, but it involves deep changes in it. The first two are rather impossible to deal with in shallow grammar approach, unless we list every particular phrase like *prawa człowieka* separately. However, we should mention here the disagreements between the annotators, mostly regarding the way parts of the nominal group should (or not) be brought to singular form: should *polipy nosa i zatok 'nasal and sinus polyps'* be lemmatised to *'polip nosa i zatoki polyp (of the) nose and sinus)'* or rather *polip nosa i zatok 'polyp (of the) nose and sinuses'*?

6 Conclusions and future work

The first attempts to apply a shallow grammar to the task of lemmatising nominal syntactic groups give promising results, especially taking into account the fact that the task itself is not easy to define – even in the relatively small evaluation set we used, there is no undisputable "golden" lemma for many of the syntactic groups.

Although the grammar is obviously language-dependent, the whole approach is not. We know about a Spejd grammar being prepared now to extract syntactic groups from Modern Greek texts. That grammar can be augmented with lemmatisation patterns in the future.

An important conclusion, concerning at least Polish and the NKJP grammar, is that for the best results some parts of it should be rewritten with lemmatisation in mind (the grammar described in this paper was just quickly adapted to the task). Rewriting should be consulted with the authors of the original grammar.

Doing so, we may avoid some compromises, especially in dealing with adjectival phrases of various types (adjectives and participles) that form parts of the noun phrases.

More specific patterns (such as dates) should be added to deal with special cases.

Finally, the case of the lemmatised strings should be retained. This will be done by replicating the case pattern (regarding initial characters of words) to the lemmatised string.

References

1. Acedański, S.: a Morphosyntactic Brill Tagger for Inflectional Languages. Advances in Natural Language Processing, 2010, pp. 3-14 (2010)
2. Buczyński, A., Przepiórkowski, A.: Spejd: a Shallow Processing and Morphological Disambiguation Tool. In: Vetulani, Z., Uszkoreit, H. (eds.) Human Language Technology: Challenges of the Information Society. LNAI, vol. 5603, pp. 131-141. Springer-Verlag, Berlin (2009)
3. Głowińska, K., Przepiórkowski, A.: 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
4. Khaltar, B.-O., Fujii, A.: a lemmatization method for Mongolian and its application to indexing for information retrieval. In: Information Processing and Management: an International Journal, Volume 45 Issue 4, 438–451 (2009)
5. Pala, K., Rychlý, P., Šmerk, P.: Automatic Identification of Legal Terms in Czech Law Texts. In: Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.) Semantic Processing of Legal Texts. LNCS, vol. 6036, pp. 83-94. Springer Berlin / Heidelberg (2010)
6. Piskorski, J., Sydow, M., Kupść, A.: Lemmatization of Polish person names. In: ACL '07 Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies. Association for Computational Linguistics Stroudsburg, PA (2007)
7. Przepiórkowski, A., Górski, R.L, Łaziński, M., Pęzik, P.: Recent Developments in the National Corpus of Polish. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh conference on International Language Resources and Evaluation (2010)
8. Waszczuk, J., Głowińska, K., Savary, A., Przepiórkowski, A.: Tools and Methodologies for Annotating Syntax and Named Entities in the National Corpus of Polish. In: Proceedings of Computational Linguistics - Applications (CLA 2010), Workshop at IMCSIT 2010, Wisła, Poland, October 18-20