Recognition of Polish Derivational Relations Based on Supervised Learning Scheme

Maciej Piasecki, Radosław Ramocki, Marek Maziarz

Institute of Informatics, Wrocław University of Technology,
maciej.piasecki@pwr.wroc.pl, marek.maziarz@pwr.wroc.pl, radoslaw.ramocki@pwr.wroc.pl

Abstract

The paper presents construction of *Derywator* – a language tool for the recognition of Polish derivational relations. It was built on the basis of machine learning in a way following the bootstrapping approach: a limited set of derivational pairs described manually by linguists in plWordNet is used to train *Derivator*. The tool is intended to be applied in semi-automated expansion of plWordNet with new instances of derivational relations. The training process is based on the construction of two transducers working in the opposite directions: one for prefixes and one for suffixes. Internal stem alternations are recognised, recorded in a form of mapping sequences and stored together with transducers. Raw results produced by *Derivator* undergo next corpus-based and morphological filtering. A set of derivational relations defined in plWordNet is presented. Results of tests for different derivational relations are discussed. A problem of the necessary corpus-based semantic filtering is analysed. The presented tool depends to a very little extent on the hand-crafted knowledge for a particular language, namely only a table of possible alternations and morphological filtering rules must be exchanged and it should not take longer than a couple of working days.

Keywords: recognition of derivational relations, Polish, wordnet, plWordNet, morphology learning

1. Introduction

Derivational relations (e.g. aspect, deminutivity or young being) are numerous in Slavic languages, encode semantic information and are important for the construction of lexical semantics resources. However, morphological analysers rarely provide extensive description of derivational relations, and for some languages, e.g. Polish, do not encompass them. Handwritten rules are typically used for the description of derivatives, but their construction is time-consuming. Instead, we aim at constructing an analyser of derivational relations for Polish based on the bootstrapping approach and supervised training on the basis of a limited set of examples. The task of the analyser is to recognise a word form as a derivative and identify its derivational base and the precise relation linking them. The starting point are derivational relations described in *plWordNet* – the largest Polish wordnet, but the algorithm should be open to any set of relations and different languages.

Works dedicated to derivational morphology learning are relatively rare. Derivational rules are mostly extracted as a part of the general morphology learning task. In this field, two groups of methods can be distinguished (Walther and Nicolas, 2011): aimed at automated construction of morphological analysers and extraction of morphological models (e.g. segmentation and rules). e.g. (Golénia et al., 2010). In the first group, most methods are based on unsupervised learning from large annotated corpora. Combinations of different methods of statistical analysis are used in order to identify affixes, stems and word form families, e.g. (Schone and Jurafsky, 2001), *Minimum Description Length* concept is often used in discovering segmentation, e.g. (Kohonen et al., 2009), cf overview in (Walther and Nicolas, 2011). Methods supported by declared linguistic knowledge were also proposed, e.g. for Polish (Sagot, 2009). Walther and Nicolas (2011) presented corpus-based extraction of derivational rules. Only derivative candidates above some minimal frequency in the corpus were considered. 62,158 derivative candidates were extracted from 37.5 million token French. 1,511 new derived French lemmas were identified after ranking candidates. Manual evaluation of a small sample of 100 lemmas resulted in: 42 lemmas and relations identified as correct, and 43 lemmas definitely incorrect (many due to foreign words and typos). Contrary to corpus based approaches we aim at using a limited but manually annotated set of derivational pairs and to build an analyser as a tool for automated expansion of the data prepared by linguists.

In a similar way to our goal, *memory based learning*, e.g. (van den Bosch and Daelemans, 1999), and *transformation-based learning* paradigms, e.g. (Oflazer et al., 2001) were applied to limited training data. The latter was even applied to Polish, but no evaluation was reported. Both approaches are relevant to our goal, however, Polish derivational rules can be reduced to: prefix and suffix, in relation to the derivative base stem combined with a limited set of *internal stem alternations* (Rabiega-Wiśniewska, 2009). Moreover, prefix and suffix occur very rarely together in the same rule. Rules of this scheme can be directly implemented in a transducer. Morphological guessers based on transducers extracted from anotated data were successfully applied to Polish, e.g. for Polish (Daciuk, 2001) and a large scale Polish guesser called *Odgadywacz* (Piasecki and Radziszewski, 2008) of high precision and recall. However, transducer guessers have problems with alternations (store exact word parts) and with prefixes (are mostly built on the basis of *a tergo* indexes). We propose solutions to both problems.

2. Derivator

The need for the analyser of derivatives originated from the work on extending plWordNet (a Polish wordnet) with derivational relations. High productivity of many Polish derivational rules suggested a bootstrapping approach: a limited set of examples added to *plWordNet* by linguists becomes a basis for automated construction of derivational analyser called *Derivator*. It is next used in expanding derivational relations in *plWordNet*. The process is started with applying Derivator to a large set of Polish lemmas, next derivatives are recognised together with their derivational bases and the relations. In each iteration rules learned from the described derivational pairs are applied to all Polish lemmas in order to filter out other derivational pairs fulfilling the trained patterns. Supervised learning algorithm is preferred in this scheme to obtain high precision derivational rules from a limited set of examples.

2.1. Learning

Derivative relations are manifested in Polish and other Slavic languages by relatively regular processes of transforming derivative bases (word forms) into derivatives by adding suffixes (mostly) and prefixes (less often, only for some types). Thus Polish derivative relations are encoded in a similar way to morphological oppositions: the differences between a morphological word form and the lemma (morphological base form) is mostly expressed by a suffix and only some of them by a prefix. For a word form not included in a morphological dictionary values of its grammatical categories (e.g. case, number, gender etc.) can be 'guessed' (predicted) on the bases of extracting from the analysed word form a prefix and/or suffix.

The starting point for the learning algorithm was *Odgadywacz* – a large scale, high accuracy morphological guesser for Polish which generates morphological descriptions for unknown Polish word forms with relatively high accuracy on the basis of suffixes learned from annotated examples (Piasecki and Radziszewski, 2008). The guesser is based on a deterministic transducer which takes a reversed sequence of letters of a word and returns morphological description attached to the leaf node, in case it was reached, or a branch node which was a final node for some word form during learning.

The guesser learning is divided into two phases: transducer tree building and pruning. First, a transducer path is built for each reversed word form and each letter. Nodes that were terminal for some word form are marked. Next, all final non-branching path parts are cut off, but all terminal nodes are preserved. Information concerning lemmas (morphological base forms) is acquired in a form of reconstruction rules: the number of letters to be cut off from a word form and a suffix to be added. The rules are stored in the terminal nodes together with the tags.

On the basis of word form frequency list collected from a large corpus, morpho-syntactic tags and base form reconstruction rules and morphological specifications are added to the terminal nodes. Generalisation can be improved by pruning some terminal nodes, e.g. those with the smallest number of training examples.

In order to expand *Odgadywacz* into *Derivator* two problems had to be solved: internal stem alternations and construction of derivatives on the basis of both: suffixes and prefixes (sometimes both, too). Representation of suffixes and prefixes has been provided by training two guessers applied in parallel in reversed directions, but only one at a time for the given training example, see the algorithm below.

Support for stem alternations had to be added to the *Odgadywacz* algorithm. The idea is to find a sequence of alternations that makes the derivational base overlapping on its ending or beginning with a derivative. The identified sequences are applied to the derivational bases before passing the training data to the suffix-based, or prefix-based guesser. The sequences are stored in the guesser nodes together with the relation tag. Thus, a derivational rule is learned in a form of a suffix / prefix and associated sequence of alternations that can affect the whole word form not only the suffix/prefix. The recorded alternations are used for reconstructing derivative bases during guessing.

Learning algorithm

Input: $L = \langle$a derivative, a relation tag, a derivative base$\rangle$, T – table of alternations (a mapping: letter sequences to letter sequences (up to 4 letters))

For each $e = \langle d, r, b\rangle \in L$:

1. $t_p =$ sequence of at most k substitutions from T such that t_p makes P a longer shared beginning of d and b.
2. $t_s =$ as in the above but in the relation to a shared ending S
3. If $length(P) \geq length(S)$
   - then add $\langle d, r+t_p, t_p(b)\rangle$ to the training set of the normal guesser,
   - else add $\langle rev(t_s(d)), r + t_s, rev(b)\rangle$ to the training set of the 'reversed' guesser.

Table of alternations (in the form defined by linguists) contains N lines. In each line there are K_n letter sequences defining alternation group. Alternation rule is a mapping from one letter sequences to another. Every alternation group n is used to build $K_n^2 - K_n$ alternation rules, that is all two-element permutations. Below there are two examples of alternation group expanded into alternation rules:

- ⟨ch sz chi⟩: (ch, sz), (ch, chi), (sz, ch), (sz, chi), (chi, ch), (chi, sz)
- ⟨o ó⟩: (o, ó), (ó, o)

In steps 1–2, the longest ending and beginning shared by the derivative and its base are identified. But, we assume that alternations can occur in any position and T (127 possible alternation groups defined by a linguists, full list of alternation groups is attached at the end of this paper) is used to iteratively extend the found shared beginning and ending.

Alternation rules application algorithm

Input: $\langle d, b\rangle$ – derivative and it's base (in case of longer shared ending letters of d and b must be reversed), R – list of alternation rules, k – maximum number of alternation rules that can be applied to the derivative base (in our experiments we used 3)

Output: t – list of applied rules and modified derivative base b

```
For each $r \in R$:
  If $length(t) \geq k$:
    return $(t, b)$
  For each word $w$ constructed by application of rule $r$ to $b$:
```

```
  If sharedPrefixLength($d$, $w$) $\geq$ sharedPrefixLength($d$, $b$):
    1. add $r$ to $t$
    2. $b = w$
return $(t, b)$
```

E.g., for the derivative *świecznik* ‘candlestick’ and its base *świeca* ‘candle’ the mapping `['cz'/'c']` is added to t_p: $b =$‘świecza’ and $P =$‘świecz’. Here are another examples:

- a pair *dolatywać* ‘(about flying objects) reach$_{impf}$’ – *dolecieć* ‘reach$_{pf}$’ includes two alternations [‘la’/‘le’] and [‘t’/‘c’], $b =$ ‘dolatieć’ and $P =$ ‘dolat’,
- for a pair *pszczelarz* ‘beekeeper’ – *pszczoła* ‘bee’ three mappings [‘cze’/‘czo’], [‘le’/‘ła’] and [‘la’/‘le’], $b =$ ‘pszczela’ and $P =$ ‘pszczela’ are found,
- a deminutive form *bułeczka* ‘nice/little (bread) roll’ and its base *bułka* ‘roll’ have two alternations [‘łe’/‘ł’] and [‘cze’/‘k’], $b =$ ‘bułeczea’ and $P =$ ‘bułecz’.

In step 3 the lengths of the shared beginning and ending are compared in order to decided which guesser to use. Longer beginning means suffix-based derivation and the normal guesser. The reversed guesser is used in the case, e.g., *zrobić* ‘to do$_{perf}$’ – *robić* ‘to do$_{imperf}$’. The prefix-based guesser is simulated by reversing the letter order in both elements of the training samples: the derivative and its transformed base.

2.2. Application – Derivative Base Recognition

We assumed that the input is a word form which is possibly a derivative. The task is to identify its derivative base, if exists, and the derivational relation (including its subtype) which links them. For an input word there is no information which guesser – *Derivator* module – to use: the suffix or the prefix-based one, if any (the input does not need to be a derivative). Thus, both modules must be applied in parallel and the appropriate result selected on the basis of the available knowledge sources.

First of all, we can expect that *Derivator* produces proper Polish word forms. Thus, we applied *Morfeusz SGJP*[1] – a Polish morphological analyser of extensive coverage – to filter the output: only words recognised by it are accepted. As this procedure is limited only to the lemmas covered by *Morfeusz*, we tested also corpus based filtering, see Sec. 4.. Morphological filtering applied next is based on the observation that for most derivational relations we can formulate constraints on the acceptable morphological characteristics of the derivational pair, e.g. for *inhabitant* only nouns in the nominative case are accepted on both sides and the derivative cannot be in the non-animated gender, while the base gender is not restricted, see the rules in the appendix.

Recognition algorithm

Input: a lemma l, *Derivator* modules, R – morphosyntactic filtering rules.

1. l is delivered to both modules (to one of them in a reversed form) that return a set of triples: $\langle b, t, r\rangle$ where b is a base as reconstructed by the guesser, t a sequence of substitutions associated with the guesser node during learning, r – relation tag.
2. For each triple:
   (a) b is transformed by the reversed sequence of substitutions t.
   (b) if b was generated by the prefix-based guesser than it must be reversed.
3. Triples: $\langle l, r, b\rangle$ are filtered:
   - l and b are first morphologically analysed – non recognised pairs are discarded,
   - and next all morphological descriptions (forms can be ambiguous) are compared with the filtering rules for r – at least one pair of descriptions must match the rule.

In step 3, non-words or non-lemmas, that are often generated by the guesser modules, are filtered out from the result. Morphological filtering limits the intrinsic lexical overgeneration of the guessers. Very often non-words or non-lemmas are generated as potential derivative bases, especially for input lemmas that are not derivatives. However, the filtering based on an existing analyser blocks the possibility to go beyond the word forms covered by it. An alternative is filtering based on the lemma frequency list collected from a large corpus. Its influence is discussed in the evaluation section.

3. Derivational relations in plWordNet

Derivational pairs already described in plWordNet were a basis for the evaluation. *plWordNet*[2] is the largest wordnet of Polish and one of the largest in the world, the version from 06.09.2011 (including about 70000 synsets and 100000 lexical units) was used during evaluation.

The relation set for *plWordNet* includes derivational relations that are regular (“from several hundred to several thousand occurrences in lexicon”, see (Maziarz et al., 2011a, p. 175), (Maziarz et al., 2011b)). We give frequency data after (Grzegorczykowa and Puzynina, 1998) who makes use of the ‘lexicon’ frequencies in (Doroszewski, 1969).

Femininity (N-N) is a relation linking nouns denoting females with their bases referring to male counterparts: $X_{derivate}$–Y_{base} is ‘X is female Y’. A suffix *-ka* (*psycholożka* ‘female psychologist’ < *psycholog* ’psychologist’) forms almost fully productive type[3]. Sufixes *-ini/-yni* (*władczyni* ’female ruler’ < *władca* ’ruler’), *-ica* (*tygrysica* ‘female tiger’ < *tygrys* ‘tiger’) or -a (*markiza* f. ‘marquise’ < *markiz* m. ‘marquis’) are also very frequent (Grzegorczykowa and Puzynina, 1998, p. 422-5). All formations taken together account for 1745 instances of this relation in *plWordNet*.

Markedness (N-N relation) connects nouns such that a *marked* one denotes objects of almost the same type as does

[1] `http://sgjp.pl/morfeusz/`

[2] `plwordnet.pwr.wroc.pl`

[3] The type *-ka* accounts for 913 instances (Rabiega-Wiśniewska, 2003).

its counterpart (umarked base) but with additional properties. The most productive types of markedness are:

Diminutives denotes positive emotional marking or small size. The meaning could be defined thus: *'X_{deriv} is a little or pleasant Y_{base}'*. This relation subtype is very frequent in Polish. *-Ek*/*-ik*(*-yk*), *-ko* and *-ka* are the most frequent affixes: *piesek* 'little or pleasant dog' < *pies* 'dog', *kocyk* 'little or pleasant blanket' < *koc* 'blanket', *serduszko* < *serce* 'heart', *kostka* < *kość* 'bone' (Grzegorczykowa and Puzynina, 1998, 425-6). From historical perspective some formations could be considered as derived, such cases we used to treat from the synchronic point of view: e.g., word *młotek* 'hammer' was derived from *młot* 'heavy hammer' with suffix *-ek*, it will be ridiculous nowadays to say that *młotek* is 'small or pleasant *młot*' and this relation instance is not included in *plWordNet*.

Augmentatives express negative emotional marking and grand size of denotatum, and its paraphrase is: *X_{deriv} is huge or terrible Y_{base}'*. Suffixes of augmentatives are *-uch*, *-isko*(*-ysko*) or *-al*: *staruch* 'terrible old man' < *starzec* 'old man', *komarzysko* 'huge or terrible mosquito' < *komar* 'mosquito', *nochal* < *nos* 'nose' (Grzegorczykowa and Puzynina, 1998).

Young being names youth of derivative's denotatum and may be paraphrased as: *'X_{deriv} is young Y_{base}'*. Two formants are used (their distribution is regionally conditioned): *-ę* and *-ak*, e.g.: *ptaszę* 'young bird, esp. nestling' < *ptak* 'bird', *kocię* 'kitten' < *kot* 'cat', *kociak* 'kitten' < *kot* 'cat', *psiak* 'pup' < *pies* 'dog' (Grzegorczykowa and Puzynina, 1998, pp. 429-30).

Role (N-V) refers to thematic roles of predicate arguments, e.g., agent, object, instrument etc. The most frequent is an agent subtype with suffixes *-acz* (*badacz* 'researcher' < *badać* 'to research'), -ca (*władca* 'ruler' < *władać* 'rule'), *-iciel* (*pocieszyciel* 'comforter' < *pocieszyć* 'to comfort, to console'), *-ator* (*restaurator* 'restorer' < *restaurować* 'restore'). Also (less frequent) backward (paradigmatic) derivation occurs in that subtype (*szpieg* 'spy' < *szpiegować* 'spy'). Suffixal and parafigmatic formations account for above 3500 instances in (Doroszewski, 1969), cf. (Grzegorczykowa and Puzynina, 1998, pp. 398-416). In *Słowosieć* the pointer was used 4072 times and is a most favourite subtype among editors.

Role inclusion (V-N) expresses also thematic roles of predicate arguments, but those evoked by a verb structure. This derivation type is relatively frequent in *plWordNet*: 1262 instances. According to (Wróbel, 1998, pp. 577-83) *instrument* and *result* are the most frequent subtypes in Polish, e.g. (*solić* 'to salt' < *sól* 'salt', *dziurkować* 'to perforate' < *dziurka* 'hole'), next *object* (*kartkować* 'to leaf through' < *kartka* 'a sheet') and *subject* (*sędziować* 'to referee' < *sędzia* 'referee') come (the rest types are less productive). The assumptions find confirmation in *plWordNet* data statistics.

Cross-categorial synonymy is of extreme frequency. The relation is a type of *transposition* (two related words differ only in their parts of speech).

N-V subtype links deverbal nouns (gerunds) with their bases. We account for regular type on *-anie*, *-enie*, *-cie*: *granie* 'playing' < *grać* 'play$_{impf}$', *zapełnienie* 'filling' < *zapełnić* 'fill$_{pf}$', *bicie* 'hitting' < *bić* 'hit$_{impf}$' (Grzegorczykowa and Puzynina, 1998, pp. 393-8). In *plWordNet* we have 1334 instances of the relation.

N-Adj type refers to deadjectival nouns on *-ość*: *bladość* 'paleness' < *blady* 'pale', *władczość* 'imperiousness' < *władczy* 'imperious', *małość* 'smallness' < *mały* 'small'; this type is regular. In (Doroszewski, 1969) there are noted about 3500 such derivatives, but because of the productivity of the formation, there potentially could be much more such derivatives (Grzegorczykowa and Puzynina, 1998, p. 417). In *plWordNet* we have 1513 instances of the formation.

State/feature bearer (N-Adj) and **state/feature** (Adj-N) are both very productive in Polish. The meaning of the relation linking X_N-Y_{Adj} could be articulated in following way: *X is/has feature Y*. The most frequent suffixes (more than 100 lemmas in (Doroszewski, 1969)) are *-ec* (*głupiec* 'a fool, idiot' < *głupi* 'fool'), *-ka* (*dziczka* ' rootstock' < *dziki* 'wild'), *-ak* (*dziwak* 'freak' < *dziwny* 'strange'), *-ik* (*nędznik* 'scoundrel' < *nędzny* 'poor'), these relations are represented by about 600-800 instances (Grzegorczykowa and Puzynina, 1998, p. 420-1). Till now we have introduced 219 feature bearer relations into *plWordNet* .

Inhabitant (N-N) describes X as an 'inhabitant/dweller of Y', where Y is the base denotatum. Inhabitant names are formed on the basis of geographical proper names (for countries, regions, cities, towns, villages and parts of the world) with such suffixes as *-anin* and *-czyk* or with paradigmatic backward derivation: *Amerykanin* 'American' < *Ameryka* 'America or USA', *Panamczyk* 'Panamanian' < *Panama* 'Panama', *Bułgar* 'Bulgarian' < *Bułgaria* 'Bulgaria' (Grzegorczykowa and Puzynina, 1998, pp. 437-8) (147 instances in *plWordNet*).

Aspectuality (V-V) expresses aspectual differences and *Aktionsarten*. Two subtypes are present in *plWordNet*: 9145 instances of pure aspectuality (only aspectual differences, *wykopać* 'to dig$_{pf}$ sth up' - *wykopywać* 'to dig$_{impf}$ sth up'), 3979 instances of secondary aspectuality (aspect differences + lexical meaning shift, *zaświecić* 'to start$_{pf}$ shining' - *świecić* 'shine$_{impf}$').

Derivationality groups all derivational relations that are not included in the above subtypes.

4. Evaluation

First, a modified 10-fold cross validation was performed on the level of main relations without distinguishing subtypes. Pairs for each relation were randomly divided into 10 subsets. One subset per relation was used for testing in each iteration. The results are presented in table 1. The lower recall is caused by a small number of examples for many subtypes and also by long and detailed pseudo-suffixes generated in order to distinguish different pairs: subtype and alternation sequence presented during learning, e.g. suffix *żek* encodes deminutivity, but is associated with different alternation sequences: any, ['g'/'ż'], ['bó'/'bo','g'/'ż']

In order to estimate the support provided by *Derivator* for the wordnet expansion two large scale experiments were performed on data from the outside of *plWordNet*. We used a list of 341230 word forms (nouns, gerunds, adjectives)

Relation	Precision [%]	Recall [%]
aspectuality	99.5	87.8
derivativity	78.8	31.5
feature bearer	62.7	10.8
femininity	97.0	63.6
inhabitant	71.7	12.4
state	66.3	12.2
markedness	97.7	61.3
semantic role	83.7	37.8
c-c synonymy	93.5	81.5
role inclusion	100	45.5

Table 1: Coarse-grained cross-validation results.

from *Morfeusz SGJP*. In the first experiment, the feasibility of the bootstrapping approach in terms of the gain received from manually annotated examples was analysed. *Derivator* was trained twice: first on data acquired from *plWordNet* the version 18.08.2011 (15718 examples) – *basic training set* – and the second time on version 06.09.2011 (17971 example, 14% more) – *extended training set* (it was used in Tab. 1). Both versions were next applied to the whole lemma list. *Derivator* trained on the basic training set recognised 158363 potential derivational pairs while the one based on the extended set 166600 pairs. The difference, gain obtained from the manually added pairs to the wordnet, is 8237 pairs, while there were 2253 new derivational pairs added by linguists to the wordnet. An algorithm for the selection of new pairs to be added manually in a way increasing the gain is required.

Not all pairs recognised by *Derivator* are correct. For the evaluation of precision, we selected a subset of 26727 recognised pairs such that the generated derivative base has only one sense in *plWordNet*. Derivational relations can be valid only for specific senses of the base and in the case of the monosemous bases, evaluators do not need to consider different possible senses. Precision for the selected subset was checked manually. For each derivational subtype a sample of at most 50 pairs was randomly selected (for smaller subtypes all pairs were included). Samples were evaluated by linguists and they were asked to assign pairs to the three classes: Correct *Subtype*, Correct *Type* – a pair does not represent the subtype returned by *Derivator* but another subtype of the same type, e.g. the pair is not a *role:tool* but a *role:object* instance, *Derivatives* (a different type of derivation), *non-derivational* pair, see Tab. 2. Concerning the limited number of the training examples, the precision of the identification of derivatives is on a good level, mostly above 80%. Only for a few subtypes, the precision is unacceptable or lower. The main reasons are: too limited sets of training examples and too broad dictionary of *Morfeusz* including many proper names.

A possible solution for the second problem is filtering based on the lemma frequency in a large corpus. Lemma frequencies were collected for the corpus of 1.5 billion words. The filtering of lemmas occurring less than 10 times applied to the samples evaluated earlier by linguists eliminated 25% of non-derivational pairs recognised by *Derivator*. While the obtained results were perceived as better by linguists, the precision was insignificantly lower. Rare lemmas were eliminated from both proper (e.g. *królobójczyni* ‘king’s assassin$_{female}$’ –femininity– *królobójca* ‘king’s assassin’) and improper pairs, e.g. *bisiorka* ‘a kind of duck’ –femininity– *bisior* ‘byssus (a kind of cloth)’. However, many errors were caused by proper names that are enough frequent and were not filtered out, e.g. *dębica* ‘town name’ –femininity– *dąb* ‘oak’.

The comparison of the general high precision with these of the subtype level shows problems with precise differentiation among different kinds of semantic associations.

5. Further Research

The presented approach is a relatively simple and it is based on automated extraction of transducers extended with internal stem alternations from annotated word pairs. However, the learned derivational rules are productive and express good overall accuracy. The list of possible internal stem alternations is the only language dependent element. Further increase of the precision would be difficult without semantic filtering of the pairs – a lemma pair matching the derivational rule does not need to be semantically associated, e.g. *fryzyjczyk* ‘Frisian’ – *fryz* ‘frieze’, *bednarka* ‘cooperage’ is not related by femininity to *bednarz* ‘cooper’ (even that the suffix *-ka* is typical for that relation), or it represents a different relation than the one of the rule, e.g. *przepychaczka* ‘declogger’ is not a feminine from *przepychacz* ‘plunger’. In many cases erroneous pairs can be eliminated only on the basis of their semantic properties. The filtering can be based on the wordnet structure or a kind of semantic, corpus-based filtering and sense identification. Appropriate selection of the new manually described examples in a way maximising the expected gain achieved in training *Derivator* on extended data is need in order to optimise human workload and guarantee exploration of different derivational rules.

Acknowledgments

Work financed by the Polish Ministry of Education and Science, Project N N516 068637.

6. References

J. Daciuk. 2001. Computer-assisted enlargement of morphological dictionaries. In *Proc. of Finite State Methods in Natural Language Processing Workshop, 13th ESSLLI, Helsinki August 2001.*

W. Doroszewski, editor. 1969. *Słownik języka polskiego*, volume I-X. PWN, Warszawa.

B. Golénia, S. Spiegler, and P. Flach. 2010. Unsupervised morpheme discovery with ungrade. In Carol Peters, Giorgio Di Nunzio, Mikko Kurimo, Thomas Mandl, Djamel Mostefa, Anselmo Peas, and Giovanna Roda, editors, *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, volume 6241 of *LNCS*, pages 633–640. Springer.

R. Grzegorczykowa and J. Puzynina, 1998. *Gramatyka współczesnego języka polskiego. Morfologia*, volume 2nd, chapter IV: Słowotwórstwo. Rzeczownik, pages 389–468. PWN.

Relation	Subtype	Precision [%]		
		Subtype	Type	Derivatives
derivativity	-	60.0	60.0	64.0
feature bearer	-	34.0	34.0	88.0
femininity	-	74.0	74.0	92.0
inhabitant	-	8.11	8.11	37.8
state	-	46.0	46.0	88.0
markedness	deminutivity	72.0	74.0	84.0
markedness	augmentativity	32.0	36.0	44.0
markedness	young being	10.5	78.9	81.6
semantic role	agent od hidden predicate	40.0	42.0	70.0
semantic role	agent	45.2	61.9	73.8
semantic role	time	20.0	20.0	100.0
semantic role	location	43.2	61.4	79.5
semantic role	location of hidden predicate	40.0	40.0	86.0
semantic role	instrument	42.0	58.0	90.0
semantic role	patient	44.0	64.0	90.0
semantic role	other	10.0	10.0	92.0
semantic role	product of hidden predicate	23.3	23.3	70.0
semantic role	product	12.0	12.0	96.0
cross-categorial synonymy	N-ADJ	81.6	81.6	87.8
role inclusion	other	45.8	60.4	79.2
role inclusion	agent inclusion	22.0	60.0	72.0
role inclusion	time inclusion	0.0	66.7	66.7
role inclusion	location inclusion	0.0	84.2	84.2
role inclusion	instrument inclusion	20.0	84.0	86.0
role inclusion	patient inclusion	8.0	76.0	76.0
role inclusion	product inclusion	40.0	66.0	76.0

Table 2: Manual evaluation of pairs with new derivative and monosemous base.

O. Kohonen, S. Virpioja, and M. Klami. 2009. Allomorfessor: towards unsupervised morpheme analysis. In *Evaluating systems for multilingual and multimodal information access, 9th Workshop of the CLEF*, pages 975–982, Berlin, Heidelberg. Springer.

M. Maziarz, M. Piasecki, J. Rabiega-Wisniewska, and S. Szpakowicz. 2011a. Semantic relations among nouns in polish WordNet grounded in lexicographic and semantic tradition. *Cognitive Studies Études Cognitives*, 11:161–181.

M. Maziarz, M. Piasecki, S. Szpakowicz, J. Rabiega-Wiśniewska, and B. Hojka. 2011b. Semantic relations between verbs in Polish Wordnet 2.0. *Cognitive Studies*, 11:183–200.

K. Oflazer, S. Nirenburg, and M. McShane. 2001. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computational Linguistics*, 27:59–85.

M. Piasecki and A. Radziszewski. 2008. Morphological prediction for polish by a statistical *A Tergo* index. *Systems Science*, 34(4):7–17.

J. Rabiega-Wiśniewska. 2003. A new classification of polish derivational affixes. In *Investigations into Formal Slavic Linguistics. Contributions of FDSL V*, Leipzig University. Peter Lang GmbH.

J. Rabiega-Wiśniewska. 2009. On the root-based lexicon for polish. In M. Marciniak and A. Mykowiecka, editors, *Aspects of Natural Language Processing*, volume 5070 of *LNCS*, pages 61–82. Springer Berlin / Heidelberg.

B. Sagot. 2009. Building a morphosyntactic lexicon and a pre-syntactic processing chain for polish. In H. Vetulani, Z.t; Uszkoreit, editor, *Human Language Technology. Challenges of the Information Society*. Springer.

P. Schone and D. Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *NAACL*. ACL.

A. van den Bosch and W. Daelemans. 1999. Memory-based morphological analysis. In *ACL*.

G. Walther and L. Nicolas. 2011. Enriching morphological lexica through unsupervised derivational rule acquisition. In *Proc. of the Inter. Workshop on Lexical Resources (WoLeR) at ESSLLI. Ljubljana*.

H. Wróbel, 1998. *Gramatyka współczesnego języka polskiego. Morfologia*, volume 2nd, chapter IV: Słowotwórstwo. Czasownik, pages 389–468. PWN.

A Appendix

Alternation groups

⟨p pi⟩, ⟨b bi⟩, ⟨m mi⟩, ⟨w wi⟩, ⟨f fi⟩, ⟨t ci c ć⟩, ⟨d dzi dz dź⟩, ⟨s si ś sz⟩, ⟨z zi ź ż⟩, ⟨n ni ń⟩, ⟨r rz⟩, ⟨l ł⟩, ⟨k cz c ki⟩, ⟨g ż dz gi c⟩, ⟨ch sz chi⟩, ⟨h ż⟩, ⟨o ó⟩, ⟨pe p⟩, ⟨pie p⟩, ⟨be b⟩, ⟨bie b⟩, ⟨mie m⟩, ⟨me m⟩, ⟨wie w⟩, ⟨we w⟩, ⟨cie t⟩, ⟨te t⟩, ⟨dzie d⟩, ⟨de d⟩, ⟨sie s⟩, ⟨se s⟩, ⟨zie z⟩, ⟨ze z⟩, ⟨nie n⟩, ⟨ne n⟩, ⟨re r⟩, ⟨rze r⟩, ⟨łe ł⟩, ⟨le l⟩, ⟨kie k⟩, ⟨cze k⟩, ⟨gie g⟩, ⟨że g⟩, ⟨sze ch⟩, ⟨che ch⟩, ⟨he h⟩, ⟨że h⟩, ⟨i ij j⟩, ⟨pie po pó⟩, ⟨bie bo bó⟩, ⟨mie mo mó⟩, ⟨wie wo wó⟩, ⟨cie to tó⟩, ⟨dzie do dó⟩, ⟨sie so só⟩, ⟨zie zo zó⟩, ⟨nie no nó⟩, ⟨rze ro ró⟩, ⟨le ło łó⟩, ⟨cze ko kó⟩, ⟨że go gó⟩, ⟨sze cho chó⟩, ⟨o ó⟩, ⟨po pa⟩, ⟨bo ba⟩, ⟨fo fa⟩, ⟨wo wa⟩, ⟨to ta⟩, ⟨do da⟩, ⟨so sa⟩, ⟨zo za⟩, ⟨no na⟩, ⟨ro ra⟩, ⟨ło ła⟩, ⟨ko ka⟩, ⟨go ga⟩, ⟨cho cha⟩, ⟨pie pa⟩, ⟨bie ba⟩, ⟨fie fa⟩, ⟨wie wa⟩, ⟨cie ta⟩, ⟨dzie da⟩, ⟨sie sa⟩, ⟨zie za⟩, ⟨nie na⟩, ⟨rze ra⟩, ⟨le ła⟩, ⟨cze ka⟩, ⟨że ga⟩, ⟨sze cha⟩, ⟨ę ą⟩, ⟨d t -ść⟩, ⟨je jo jó⟩, ⟨pie pio pió⟩, ⟨bie bio bió⟩, ⟨mie mio mió⟩, ⟨wie wio wió⟩, ⟨cie cio ció⟩, ⟨dzie dzio dzió⟩, ⟨sie sio sió⟩, ⟨zie zio zió⟩, ⟨nie nio nió⟩, ⟨rze rzo rzó⟩, ⟨le lo ló⟩, ⟨cze czo czó⟩, ⟨że żo żó⟩, ⟨sze szo szó⟩, ⟨je ja⟩, ⟨pie pia⟩, ⟨bie bia⟩, ⟨mie mia⟩, ⟨wie wia⟩, ⟨cie cia⟩, ⟨dzie dzia⟩, ⟨sie sia⟩, ⟨zie zia⟩, ⟨nie nia⟩, ⟨rze rza⟩, ⟨le la⟩, ⟨cze cza⟩, ⟨że ża⟩, ⟨sze sza⟩, ⟨ar erz er⟩, ⟨łu ło oł uł eł łu il ół łó⟩, ⟨zg żdż⟩, ⟨st szcz⟩

Morpho-syntactic filtering rules

[aspektowość:aspektowość czysta DK-NDK] inf:imperf -> inf:perf

[aspektowość:aspektowość czysta NDK-DK] inf:imperf -> inf:perf

[aspektowość:aspektowość wtórna DK-NDK] inf:imperf -> inf:perf

[aspektowość:aspektowość wtórna NDK-DK] inf:imperf -> inf:perf

[synonimia_międzyparadygmatyczna:synonimia międzyparadygmatyczna N-ADJ] subst:*:nom:* -> adj:*:nom:*

[synonimia_międzyparadygmatyczna:synonimia międzyparadygmatyczna N-V] subst:*:nom:* ger:*:nom:* -> inf:*

[synonimia_międzyparadygmatyczna:synonimia międzyparadygmatyczna Pact-V] pact:*:nom:* -> inf:*

[nacechowanie:istota młoda] subst:sg:nom:* -> subst:sg:nom:*

[nacechowanie:deminutywność] subst:sg:nom:* depr:sg:nom:* -> subst:sg:nom:* depr:sg:nom:* subst:pl:nom:* depr:pl:nom:* -> subst:pl:nom:* depr:pl:nom:*

[nacechowanie:ekspresywność | augmentatywność] subst:sg:nom:* depr:sg:nom:* -> subst:sg:nom:* depr:sg:nom:* subst:pl:nom:* depr:pl:nom:* -> subst:pl:nom:* depr:pl:nom:*

[żeńskość] subst:sg:nom:f:* -> subst:sg:nom:m1:* subst:sg:nom:m2:* subst:sg:nom:m3:* subst:sg:nom:n:*

[rola:agens|subiekt] subst:*:nom:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:* ger:*:nom:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:pacjens|obiekt] subst:*:nom:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:* ger:*:nom:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:narzędzie] subst:*:nom:f:* subst:*:nom:m1:* subst:*:nom:m3:* subst:*:nom:n:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:f:* subst:*:nom:m1:* subst:*:nom:m3:* subst:*:nom:n:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:miejsce] subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:wytwór|rezultat] subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* ger:*:nom:f:* ger:*:nom:m3:* ger:*:nom:n:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:czas] subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:podtyp nieokreślony] subst:*:nom:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:* ger:*:nom:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:agens przy niewyrażonym predykacie] subst:*:nom:f:* subst:*:nom:m1:* subst:*:nom:n:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:f:* subst:*:nom:m1:* subst:*:nom:n:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:miejsce przy niewyrażonym predykacie] subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[rola:wytwór | rezultat przy niewyrażonym predykacie] subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> subst:*:nom:* ger:*:nom:* subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:* -> pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:*

[zawieranie_roli:zawieranie agensa|subiektu] pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:* -> subst:*:nom:* ger:*:nom:*

[zawieranie_roli:zawieranie pacjensa|obiektu] pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:* -> subst:*:nom:* ger:*:nom:*

[zawieranie_roli:zawieranie czasu] pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:* -> subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:*

[zawieranie_roli:zawieranie miejsca] pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:* -> subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:*

[zawieranie_roli:zawieranie narzędzia] pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:* -> subst:*:nom:f:* subst:*:nom:m1:* subst:*:nom:m3:* subst:*:nom:n:*

[zawieranie_roli:zawieranie wytworu | rezultatu] pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:* -> subst:*:nom:f:* subst:*:nom:m3:* subst:*:nom:n:*

[zawieranie_roli:podtyp nieokreślony] pcon:* pant:* pact:*:nom:* ppas:*:nom:* inf:* -> subst:*:nom:* ger:*:nom:*

[stan|cecha] pact:*:nom:* ppas:*:nom:* adj:*:nom:* -> subst:*:nom:*

[nosieciel stanu/cechy] subst:*:nom:f:* subst:*:nom:m1:* subst:*:nom:m2:* subst:*:nom:n:* -> pact:*:nom:* ppas:*:nom:* adj:*:nom:*

[mieszkaniec] subst:*:nom:f:* subst:*:nom:m1:* subst:*:nom:m2:* subst:*:nom:n:* -> subst:sg:nom:*

[derywacyjność] pcon:* pant:* inf:* -> pcon:* pant:* inf:* adj:*:nom:* subst:*:nom:* depr:*:nom:* ger:*:nom:* pact:*:nom:* ppas:*:nom:* -> adj:*:nom:* subst:*:nom:* depr:*:nom:* ger:*:nom:* pact:*:nom:* ppas:*:nom:*