A Procedure for Word Derivational Processes Concerning Lexicon Extension in Highly Inflected Languages

Klára Osolsobě, Karel Pala, Radek Sedláček, Marek Veber
Faculty of Informatics, Masaryk University Brno
klara@ernest.phil.muni.cz, {pala,rsedlac,mara}@fi.muni.cz

Abstract
The aim of this paper is to describe an efficient tool (I_PAR) for a supervised and semi-automatic extension of a lexicon or morphological database and its easy updating. We will present the underlying algorithms and their implementation that are general enough to capture the main word-forming processes (both inflectional and derivational). They are designed for languages with a rich inflectional morphology, such as Slavonic languages, particularly Czech. The implementation is partly based on the ideas presented in the earlier paper by Klímová and Pala (2000)

1. Introduction
Our aim has been to develop a user-friendly computer system that, based on user’s set(s) of examples, discovers rules in the form of derivational patterns and tries to apply them to the existing static Czech stem dictionary (containing approximately 150,000 items). The main feature of this approach is that the actual stem dictionary or morphological database becomes dynamic, i.e. the word forms are not stored as lemmata, but they are obtained from their roots by applying the derivational or inflectional rules. They allow us to extend and complete the general lexicon in a regular and possibly exhaustive way.

2. Procedure for morphological database extension
The main idea is the following: a user, typically a linguist, is able to formulate a set of examples capturing a particular derivational process based on some inflectional and derivational paradigms he or she is interested in. But there can be problems with its generalisation to some extent. One of the most important reasons for these complications is a limited amount of data that can be considered at once. Thus, the first step consists in collecting the examples, the more, the better. The user is allowed to specify negative examples as well. Typically, on the morphological level, an example corresponds to one or more derivational steps and consists of the appropriate basic word and the respective derived one(s), e.g. učit (teach) – učitel (male teacher) – učitelka (female teacher) – učitelstvo (teachers as a societal group).

To prevent the processes from an inadequate overgeneralisation, the derived forms have to be checked by an expert (linguist) mostly using some automatic tools whether these word forms occur in dictionaries or corpora. Our system allows every derived word form to be marked with a note saying that the word form is correct or that it is a potential word, not occurring in a dictionary or corpus, but it can be used. Incorrectly derived word forms are also collected as negative examples for the next iteration of the learning algorithm. We have not tried to optimise this method yet, rather, our main aim now is to implement the appropriate algorithms and representations of the lexicon so as to enable straightforward data maintenance.

Implementation of this process will be described in the section 3.3.2.

3. Inflectional and derivational processes
Basically, there are four major types of word-forming processes that can be distinguished:
- inflection
- derivation
- compounding
- abbreviation

Inflection refers to the systematic modification of a stem by means of suffixes and sometimes prefixes. Inflected forms express grammatical categories like case, gender or number, but do not change meaning of POS. In contrast, the process of derivation usually brings about a change in meaning and often a change in POS as well. Compounding deals with the process of merging several word bases to form a new word, whereas abbreviation shortens the word bases leaving usually just the first letter or the first syllable.

3.1. Inflectional processes
Czech belongs to the family of inflectional languages which are characterised by the fact that one morpheme, typically an ending, carries the values of several grammatical categories together. For example, an ending of nouns typically expresses a value of grammatical category of case, number and gender. This feature requires a special treatment of Czech words in text processing systems. To this end, we have developed a universal morphological analyser which performs the morphological analysis based on dividing all words in Czech texts into their smallest relevant components that we call segments. The notion of segment roughly corresponds to the linguistic concept of morpheme, which denotes the smallest meaningful unit of a language.

The morphological analyser consists of three major parts: a formal description of morphological processes via morphological patterns, an assignment of Czech stems to their relevant patterns and a morphological analysis algorithm (Sedláček and Šmrž, 2001).
The detailed description of all variations in Czech paradigms enables us to define the application dependent generalisations of the pattern system. In its fully expanded form there are 1500 patterns. But if we do not need to take into consideration archaic word forms for a specific application, the number of paradigm patterns can be considerably reduced using automatic procedures.

### 3.2. Derivational processes

As indicated in the previous section, the morphological process of inflection is captured by means of paradigms in our system. Abbreviation and compounding do not play a crucial role in Czech morphology if compared with other languages, e.g. German (Kodydek, 2000).

The process of morphologically deriving new words, primarily with distinct POS categories, is considered to be taking place at one level higher than inflectional processes. Indeed, for example, a particular class of deverbal adjectives can be derived from the derivation paradigm of transitive verbs (see the "DEVEADJ*" relations in the following example). A hierarchical system of morphological paradigms has been implemented as a tool able to capture different levels of the Czech morphology.

Hierarchical patterns are constructed fully automatically from the binding defined on the level of basic forms always connecting one lemma with another by a specified type of a link. If a process could be described as a $n$-ary relation, it would be partitioned into $n - 1$ binary relations. This partitioning is much more flexible and allows automatic generalisations of derivation relations. To demonstrate the derivation binding on the level of lemmata, we present the following example with the verb participle:

- **DEVESUBST**
  - počítat (to count) \(\rightarrow\) počítání (counting)
- **DEVEADJPAS**
  - počítat (to count) \(\rightarrow\) počítaný (counted)
- **DEVEADJPASSHORT**
  - počítat (to count) \(\rightarrow\) počítán (is counted)
- **DEVEADJACTIMPF**
  - počítat (to count) \(\rightarrow\) počítající (is counting)

From this example it follows that each link connects one base form of a word with another one and names such relation. If the label of a base form is unambiguous and can therefore be used as a primary identifier, it is sufficient to specify only these labels in the binding process. If the label itself can be ambiguous, the pairs of lemma and the relevant inflectional pattern are connected. However, even this approach is not able to completely represent the dependency of the relation on a particular sense of a word. For example, the following relation between a noun and the derived possessive adjective

- **POSSADJ**
  - editor (editor) \(\rightarrow\) editorův (editor’s)

is valid only for the reading editor denoting a man who edits books but not for the reading that denotes a computer program, the second meaning of the noun editor. This is why we have implemented a system connecting the triplet pairs, namely sense-id, lemma and paradigm, by a named relation.

The indexing techniques and dictionary methods (Knuth, 1973) used in our implementation allow an efficient retrieval of related lemmata. It is also possible to quickly return a chosen base form for a set of related words – a feature which is much preferred in some applications, e.g. in the area of information retrieval or indexing Internet documents.
The system of base form binding is not limited to the basic derivative processes described above. The same principle, for example, depicts two types of relation on the level under the basic derivation, namely original/adapted orthography and inflectional/non-inflectional doublets in the case of the loanwords. The former can be demonstrated by the example of a link between gymnasium/gymnázium (high school) (in the actual version of our morphological analyser we use an even more elaborate assignment of these doublet types in the form of a basic type of relation and more specific subtype). The example of an inflectional/non-inflectional doublet is the link between the word abbé assigned to the paradigm abbé (non-inflectional) and Tony. It is of course possible to model such relations on the basic level of inflectional paradigms as a word-form homonymy. However, it would lead to the mixture of unrelated forms and would complicate special types of analyses, e.g. a style-checker analysis could make very interesting findings.

There are other relations connecting lemma above the level of basic derivative processes. We take advantage of the standard process and are able to uniformly describe such different relations as diminutives (and their degrees): the standard process and are able to uniformly describe level of basic derivative processes. We take advantage of and would complicate special types of analyses, e.g. a

Wordnet (Miller, 1993). The typical relations of hyperonymy/hyponymy, meronymy (part/whole) etc. are modelled on the higher level, the level based on synonyms, so that groups of synonyms (“synsets” as they are known in Wordnet) can be linked.

The last class of the links brings us directly to other relations that can be found in semantic networks like Wordnet (Miller, 1993). The typical relations of hyperonymy/hyponymy, meronymy (part/whole) etc. are modelled on the higher level, the level based on synonyms, so that groups of synonyms (“synsets” as they are known in Wordnet) can be linked.

The possibility of building complex structures of links, e.g. relations of relations, is also employed in connecting roots of loanwords to their Czech equivalents. We are therefore able, like Páleš (1994), to relate words derived from the Greek root kard with the group of Czech words derived from the Czech root srd (heart), e.g. osředčt (pericardium), kardiostimulátor (pacemaker), srdce (heart), cardiolgie (cardiology).

3.3. Formal description of the derivational processes

Formally we can define all the potential sets of patterns:

- $T$ – set of all possible tags
- $A$ – alphabet, $A^*$ – set of all chains on alphabet $A$
- $P$ – class containing all sets of patterns can be defined recursively:

1. $\emptyset \in P$
2. $P \in P \Rightarrow Q \cup P \in P$, where $Q = \{(p, p, s, t, L) | p, p, s, \pi \in A^* \land \pi \subseteq T \land L \subseteq P\}$
3. nothing else is a set of patterns

$S_x(l) : A^* \rightarrow A^*$ – simple (one step) substitution for the pattern $x \equiv (p, p, s, \pi, t, L) \in P$ and word form $l \equiv p \oplus r \oplus \pi$, where $r \in A^*$, is defined:

$$S_x(l) = p \oplus r \oplus \pi$$

$T(l, x) : A^* \times P \rightarrow \mathfrak{p}(A^* \times \mathfrak{p}(T))$ – function for derivation of tagged word forms from $l \in A^*$ using pattern $x \equiv (p, p, s, \pi, t, L) \in P$ can be explained as:

$$T(l, (p, p, s, \pi, t, L)) \equiv K,$$

where if $L = \emptyset$ then $K = \{(S_x(l), t)\}$
else $K = \{(k, t \cup R) | y \in L \land (k, R) \in T(S_x(l), y)\}$

Introduced formalism is partially demonstrated in Table 2. This example describes word forming process, which derive word forms from family names (part of one pattern). The family names in Czech can be divided to three classes based on:

- male name (MAL)
- female name (FEM)
- name labelling all family together (FAM)

The example shows that all three groups of word forms can be divided grammatically into substantive (SUB), possessive (POS) adjective (ADJ).

The groups of possessive adjectives can be separated by the grammatical gender of an object which is possessive into male, female or to the family. The clusters can be further refined according to grammatical number, ….

Our approach is very similar to “two level morphology” described in Koskenniemi (1983). We introduce a hierarchical model of patterns which enables us to separate the sets of patterns for inflectional and derivational processes. The patterns can be named, marked according to their function, and sorted by linguists. A finely-tuned database of the

1. not show in the example for a simplicity’s sake
patterns will be used as a formal description of word forming processes in natural languages.

Our database starts with the inflectional level only and we will explain the process of adding a derivation level. Generally we can start with patterns (Goldsmith’s “signatures”) given from the unsupervised learning of corpora data, see Goldsmith (2001).

In our application of this work, we will not use all elements of set $P$. One finite set $Q$ will be selected for the specified language at one point in time:

$$Q \in P \land \forall (p, \bar{p}, s, \bar{s}, t, L) \in Q \Rightarrow L \subset Q$$

(4)

The stem dictionary $S$ is a set of tuples $(l, v)$, where $l \ldots$ lemma, $v \ldots$ pattern:

$$S^Q = \{(l, v) | l \in A^* \land v \in Q\}$$

(5)

The whole dictionary $D_{S^Q}$ for the stem dictionary $S$ (see equation 4) and the set of patterns $Q$ (see equation 5) will be defined:

$$D_{S^Q} = \bigcup_{(l, v) \in S} T(l, v)$$

(6)

The applications which use this formal description require that the stem be determined from the lemma and from the pattern: the reverse is also possible. We will next introduce set $L$.

There is one to one correspondence between the pattern $v$ and the triple $(\bar{p}, \bar{s}, t)$, where the $\bar{p}$ . . . the prefix, $\bar{s}$ . . . the suffix, and $t$ . . . the set of tags. Let’s assign $w \equiv (p, \bar{p}, s, \bar{s}, t, \emptyset)$ and $z \equiv (p, \bar{p}, s, \bar{s}, t, \emptyset)$, then:

$$L = \{(v, w, z) | v \in Q \land p, \bar{p} \in A^* \land t \subseteq T \land \bar{Q} \subseteq \{v|\exists (v, w, z) \in L\} \land (v, w_1, z_1), (v, w_2, z_2) \in L \Rightarrow (w_1, z_1) = (w_2, z_2)\}$$

(7)

We define the lemmatisation function $lm$, which transforms the stem (or root) $r$ assigned to the pattern $v$ to the tagged base form:

$$lm(r, v) = (S_z(k), t); \exists (v, w, z) \in L, where \ z \equiv (p, \bar{p}, s, \bar{s}, t, \emptyset)$$

(8)

It is possible to define function $st$ “inversive” to function $lm$, transforming base form $l$ assigned to pattern $v$ to relevant stem.

$$st(l, v) = S_w(l); \exists (v, w, z) \in L$$

(9)

3.3.1. Word-clusters

A set of word forms derived using a pattern from one lemma (stem, root) will be called a cluster. Word forms can be sorted hierarchically within a cluster. The hierarchical structure is based on breaking it down into equivalence classes. The components of a such decomposition (clusters — equivalence classes) will be called sub-clusters. Those sub-clusters can be divided into other equivalence classes. The hierarchy of word clusters is based on a hierarchical model of patterns introduced at the top of section 3.3. Sub-clusters are refined according to the structure of the associated pattern $(p, \bar{p}, s, \bar{s}, t, L)$ and the set of “sub-patterns” $L$.

As can be seen in Table 2, such a system enables the amount of duplicated information in our morphological database to be reduced. In this example we demonstrate that we can define the same dictionary (see equation 6) from different stem dictionaries (see equation 5):

$$S^Q_1 = \{(Aleš, Aleš_e),\}$$

$$S^Q_2 = \{(Aleš, Aleš_P), (Alšová, Alšová_P), (Alšovi, Alšoví_P)\}$$

$$S^Q_3 = \{(Aleš, Aleš), (Alešů, otcův), (Alšová, Alšová), (Alšově, Alšové), (Alšovi, Novákovi), Alšových, Novákův\})$$

$$D_{S^Q_1} = D_{S^Q_2} = D_{S^Q_3}$$

(10)

where $\equiv$ is equivalence only for the word forms, ignoring adding of some tags in some derivational levels. Assign $w \equiv (p, \bar{p}, s, \bar{s}, t, L)$, when $t = \emptyset$ for any used $w$, than it is real equivalence because of:

$$T(l, w) = \bigcup_{y \in L} T(S_w(l), y)$$

A potential application of our approach is the construction of a stem(root)-dictionary.

3.3.2. Dictionary extension

The user view of this process is described in Section 2. The underlying mechanism works like this: as was discussed in Section 3.3.1., the same dictionary can be based on different stem dictionaries. We are searching for some low level morphological patterns which can be connected by some higher level derivational pattern. For this search, the examples gathered by the user (see section 2.) can be used. Examples are $n$-tuples of members of the stem dictionary (see equation 5)

$$e = (m_1, . . . , m_n) \land m_1, . . . , m_n \in S$$

Appropriate patterns that connect low level ones by transforming their base forms are suggested to the user. The words assigned to those low level sub-patterns are derived from the lemmas $hl$ by suggested higher level patterns $hp$.

$$hl = p \oplus r \oplus \bar{s}$$

$$hp = (e, p, e, s, \bar{s}, \emptyset, L)$$

$$L = \{\{(p_i, e, s_i, \emptyset, \{v_i\})|p_i \oplus r \oplus s_i, v_i = m_i\}$$

The stem (root) $r$ is computed for any example of $e$ as the longest common substring of the strings $p_i \oplus r \oplus s_i$, where $(p_i \oplus r \oplus s_i, v_i = m_i$. The values of $p_i$ and $s_i$ are counted in the next step. The lemma $hl$ can be selected as one of the members $p_i \oplus r \oplus s_i$, then we assign $\bar{s} = p_i$ and $\bar{s} = s_i$.

Now we can reduce the stem dictionary size for all given examples, replacing all members of $e$ by $(hl, hp)$:

$$S^Q_{D(l,h)} \setminus \{m|m \in e\} \cup \{(hl, hp)\}$$

But this is the reduction over the stem dictionary for the manually assigned example only. When we want to
$T = \{ \text{MAL, FEM, FAM, SUB, ADJ, POS} \}$

$Q = \{ \text{Aleš}_F, \text{Aleš}_P, \text{Alešová}_P, \text{Novákovi}_P, \ldots \} \in \mathcal{P}$

\begin{align*}
\text{Aleš}_F & \equiv (e, e, e, e, \emptyset, \{(e, e, e, \emptyset, \{\text{Aleš}_P\})), (\emptyset, e, e, e, \emptyset, \{\text{Novákovi}_P\})) \\
\text{Aleš}_P & \equiv (e, e, e, \text{MAL}, \{(e, e, e, \text{SUB}, \{\text{Aleš}\})), (\emptyset, e, e, e, \{\text{POS, ADJ}, \{\text{otcův}\}) \\
\text{Alešová}_P & \equiv (e, e, e, \text{MAL}, \{(e, e, e, \text{SUB}, \{\text{Alešová}\})), (e, e, e, \{\text{POS, ADJ}, \{\text{Alšívé}\}) \\
\text{Novákovi}_P & \equiv (e, e, e, \{\text{FAM}, \{(e, e, e, \{\text{Novákovi}\}), (\emptyset, e, e, e, \{\text{POS, ADJ}, \{\text{Novákých}\})})
\end{align*}

$T(\text{Aleš, Aleš}_F)$

\[
\begin{array}{c|c}
T(\text{Aleš, Aleš}_P) & \text{male} \\
\hline
T(\text{Aleš, Aleš}) & T(\text{Alšův, otcův}) \\
\hline
T(\text{Alešová, Alšívé}) & \text{female} \\
T(\text{Alešová, Alšívé}) & T(\text{Alšův, Alšívé}) \\
\hline
T(\text{Alšův, Novákovi}_P) & \text{whole family} \\
T(\text{Alšův, Novákovi}) & T(\text{Alšůvých, Novákých}) \\
\end{array}
\]

Table 2: Example of word cluster hierarchy of the family name Aleš

To extend our dictionary we can try to check if the words (from set $E(v_i)$) are assigned to any pattern $v_i$, where:

$E(v_i) = \{l | (l, v_i) \in \mathcal{S}^Q\}$

can be reduced by a higher level pattern.

The sets of potential “high level” lemmas are as follows:

$hl_i = \{ p \oplus r \oplus \overline{s} | x \in E(v_i) \land x = p_i \oplus r \oplus s_i \}$

we know that:

$T(p_i \oplus r \oplus s_i, v_i) \subset T(p \oplus r \oplus \overline{s}, hp)$ for any $r \in A^+$ (11)

The derived word forms can be checked if they are assigned to the other appropriate connected patterns. There are two cases:

1. $hl \in \bigcap_{v_i} hl_i$ — can be simply reduced (resulted from equation 11)
2. $hl \in \bigcup_{v_i} hl_i \setminus \bigcap_{v_i} hl_i$ — some words, automatically computed from $hl_i$, can be added, thereby extending the dictionary, and then it can be reduced.

4. Implementation

The word derivation process in Czech consists of several stages. Each step can be realized as a module understood as a procedure starting with a basic word form on the input and producing a tuple of derived word forms in the output. More precisely, in our system the module can be defined by using $n+1$ strings and implemented as a simple non-deterministic finite-state translation automaton without loops. The automaton substitutes the string $s_0$ (the suffix of an input lemma) with the strings $s_1 \ldots s_n$ for each output tuple members $(t_1, \ldots, t_n)$. The same can be done with prefixes $(p_0 \rightarrow p_1 \ldots p_n)$ and some tags can be assigned to the $t_i$ members. This automaton can be constructed in accordance with the introduced formal description using equations 5, 4, 3.
other statistical data, such as the number of segments and PentiumIII processor with a frequency of 800MHz. Some ajka second feature is the speed of analysis. In the brief mode, and generate 5,678,122 correct Czech word forms. The

Table 3: Statistical data

| suffix     | freq. | suffix | freq. | suffix | freq. |
|------------|-------|--------|-------|--------|-------|
| -tel       | 911   | -dlo   | 483   | -nî    | 30454 |
| -ce        | 232   | -tko   | 379   | -tí    | 3622  |
| -ař        | 456   | -an    | 254   | -ost   | 7871  |
| -aře       | 916   | -anka  | 218   | -ota   | 211   |
| -ista      | 904   | -ištê  | 577   | -stvi  | 2081  |
| -iček      | 733   | -árnă  | 360   | -ctvi  | 820   |
| -ička      | 1545  | -ovna  | 208   | -stvo  | 270   |
| -iččko     | 391   | -yně   | 318   | -ctvo  | 141   |

Total: 54155

6. References

2001. Balkanet project. http://dlablab.upatras.gr.

J. Daciuk, R. E. Watson, and B. W. Watson. 1998. Incremental Construction of Acyclic Finite-State Automata and Transducers. In Finite State Methods in NLP, Bilkent University, Ankara, Turkey, June – July.

J. Goldsmith. 2001. Unsupervised Learning of the Morphology of a Natural Language. Department of Linguistics University of Chicago.

J. Hajic. 2000. Disambiguation of Rich Inflection. Charles University Press, 1st edition.

J. Klimova and K. Pala. 2000. Application of WordNet ILR in Czech Word-formation. In M. Gavrilidou, editor, Proceedings of LREC 2000, pages 987–992.

D. E. Knuth. 1973. The Art of Computer Programming: Sorting and Searching, volume 3. Addison Wesley, 2nd edition.

G. Kodydek. 2000. A Word Analysis System for German Hyphenation, Full Text Search, and Spell Checking . . . . In P. Sojka, I. Kopeček, and K. Pala, editors, Proceedings of TSD 2000, pages 39–44, Brno, Czech Republic, Sep. Springer-Verlag.

K. Koskenniemi. 1983. Two-level morphology: A general computational model for word-form recognition and production. Technical report, University of Helsinki, Department of General Linguistics.

G. A. Miller. 1993. Five papers on Wordnet. Technical report, Princeton.

K. Osolsoñé. 1996. Algorithmic Description of Czech Formal Morphology and Czech Machine Dictionary. Ph.D. thesis, Faculty of Arts, Masaryk University Brno. In Czech.

E. Páleñ. 1994. Sapfo - Paraphraser of Slovak. Veda, Bratislava.

R. Sedláñecz and P. Smrž. 2001. A New Czech Morphological Analyser a žka. In Proceedings of TSD 2001, pages 100–107, Berlin. Springer-Verlag.

M. Veber. 2001. Tools for Text Corpora and Morphological Databases. Ph.D. thesis, Faculty of Informatics, MU, Brno. Dissertation Thesis, manuscript, in Czech.

Table 4: Frequency of selected suffixes in noun lemmata

| suffix | freq. | suffix | freq. | suffix | freq. |
|--------|-------|--------|-------|--------|-------|
| -ce    | 232   | -tko   | 379   | -tí    | 3622  |
| -ař    | 456   | -an    | 254   | -ost   | 7871  |
| -aře   | 916   | -anka  | 218   | -ota   | 211   |
| -ista  | 904   | -ištê  | 577   | -stvi  | 2081  |
| -iček  | 733   | -árnă  | 360   | -ctvi  | 820   |
| -ička  | 1545  | -ovna  | 208   | -stvo  | 270   |
| -iččko | 391   | -yně   | 318   | -ctvo  | 141   |

Total: 54155

5. Conclusions

This approach has been implemented as the morphological tool, LPAR, (Veber, 2001) (which will be demonstrated). The results of the experiments with LPAR will be presented in the paper. They capture 5 agitative suffixes (tel, ce, ař, ař, ista), 3 diminutive suffixes (iček, icka, iččko), 2 instrument suffixes (dlo, tko), 2 inhabitant suffixes (an, anka), 3 location suffixes (ištê, árnă, ovna), 1 opposite-sex suffix (yně), 2 action suffixes (nî, tî), 4 property suffixes (ost, OTA, stvi, ctvi) and 2 collective suffixes (stvo, ctvo); see Table 4. The procedure and its results is going to be used to mark the selected ILR in the Czech WordNet, which will be done within the EU Project Balkanet (Bal, 2001) and possibly extended to Bulgarian and Serbo-Croatian.