Proceedings of the International Workshop on Intelligent Information Systems
IIS'2011, September 13-14, 2011, Chisinau, Republic of Moldova

Procedural Method in Derivative Lexicon Completion

Mircea Petic

Abstract: The article presents a study about generative approaches of Romanian derivational morphology processes. It consists of lexical processes during derivation description by highlighting several models of derivation. A special compartment deals with the algorithm of automatic lexical derivation.
Keywords: lexicon, prefix, suffix, derivational morphology, generative morphology.

1 Introduction

The studies of the lexical derivational process conclude that it is impossible to develop a universal derivational algorithm, which would allow a practical implementation of a generator of derivatives for all opportunities. If we raise the problem of obtaining a considerable coverage of derivatives lexicon, we can consider two approaches:

- *declarative* - in this case we store all derivatives, obtained a priori from certain sources including manual derivation;
- *procedural* - derivatives are obtained in a special automatic way from the roots and themes [1].

The subject of our research is the *procedural method*. From the above we conclude that lexicons completion can be achieved by automatic means taking into account the productive properties of derivational processes. Thus the basis for generating new derivatives is an existing lexicon. The lexicon should contain not only graphical representation of the words, but also its parts of speech.

2 Process of lexical derivation

Let us examine the process of lexical family generation. The problem is formulated as follows: the set of all possible cartesian products $P \times R \times S$, where P is the set of prefixes, R - set of roots, S - set of suffixes, we construct several subsets:

a) products that can be generated by concatenation *[p]r[s]*, where $p \in P$, $r \in R$, $s \in S$, and *[ ]* denotes that the particle may be missing;

b) products that can be generated by concatenation *[p]r'[s]*, where $p \in P$, $s \in S$, and $r' \in R'$ - root set of possible alternations.

c) products *[p']r[s']*, where $p' \in P' \subseteq P$, $s' \in S' \subseteq S$, concatenation will not form valid words in Romanian.

Thus, the automatic derivation process involves three important steps:

1. **Establishing if the word is susceptible for derivation** - check whether a sequence of characters represents a correct Romanian word and belongs to the part of speech which could be derived.
2. **Derivative models application** - the most important derivative models [2] are the following:
   a) *Affixes substitution* – the usage of corresponding affixes in the process of replacement, for example, for prefixes - *în*chide-*des*chide, for suffixes – corig*enţă*-corig*ent*;
   b) *Derivatives projection* – mixtion of affixes in the case of prefixation or suffixation of the roots, for example, a lucra → lucră*tor*, a lucra → a *pre*lucra, so lucră*tor* → *pre*lucră*tor*;
   c) *Formal derivation rules* - rules that depend on graphic representation of a word and its part of speech can lead to derivatives generation of a high degree of accuracy;
   d) *Derivational constraints* - some schemes with several parameters that reduce the class roots and affixes in order to form derivatives.
3. **Generated derivatives validation** - identification of the correctness of the Romanian generated words [3].

3 Algorithm of automatic lexical derivation

Taking into account all the above, as well as features and process of derivation, will be further described algorithm for automatic generation of Romanian lexical families. We use the following notations: $\mathtt{cvt}$ – word from which will be generated lexical family; D_{RRTLN} – set of words from the lexicon RRTLN[1]; D_{eDCD} – set of derivatives of the word $\mathtt{cvt}$ existing in eDCD; D_{SA} – set of derivatives formed by affix substitution (procedure applied to D_{eDCD} words); D_{PD} – derivatives formed by derivative projection (process applied to the words included in $D_{SA} \cup D_{eDCD}$); D_{CL} – set of derivatives formed by derivational constraints (process applied to the

1 http://imi201.math.md/elrr/

multitude of words $D_{PD} \cup D_{SA} \cup D_{eDCD}$); D_{RD} – set of derivatives formed by formal derivation rules (process applied to the multitude of words $D_{CL} \cup D_{PD} \cup D_{SA} \cup D_{eDCD}$); $D_{gen} = D_{CL} \cup D_{PD} \cup D_{SA} \cup D_{RD}$ – set of words obtained by the automatic generation; D_{NEVAL} – set of words that are not found in Internet documents are considered invalid; D_{SEMVAL} – final set of words that require manual validation derivatives; D_{VAL} - final set of words that have a sufficient frequency in Internet documents to be considered valid and represents the lexical family of the word `cvt`. Given these notations we can write the corresponding algorithm in a conventional language:

Input: `cvt`

Output: D_{VAL}

1. if ($\{cvt\} \cap D_{RRTLN} \neq \varnothing$) then goto 3
   else goto 2;
2. if (cvt in Internet)
   then read_part_of_speach(cvt);goto 3;
   else D_{VAL} :={}; goto 7;
3. Generating sets of words
   3.1. cvt $\Rightarrow$ D_{eDCD};
   3.2. $D_{eDCD} \Rightarrow D_{SA}$;
   3.3. $D_{SA} \Rightarrow D_{PD}$;
   3.4. $D_{PD} \Rightarrow D_{CL}$;
   3.5. $D_{CL} \Rightarrow D_{RD}$;
4. $D_{gen} := D_{CL} \cup D_{PD} \cup D_{SA} \cup D_{RD}$
5. Automatic validation
   5.1. D_{NEVAL} := nonval(Dgen);
   5.2. D_{SEMVAL} := semval(Dgen);
   5.3. D_{VAL} :=val(Dgen);
6. D_{VAL} := D_{VAL} + manualval(D_{SEMVAL})
7. Write(D_{VAL});
8. endalgorithm

In step 1, we check the presence of word `cvt` in RRTLN to ensure the existence of such a word in Romanian language and to be able to automatically extract its part of speech and other morphological categories for word `cvt`. If this word is not found in RRTLN the existence of the word `cvt` is checked by the Internet resources. If found, then states his part of speech by asking the user and passed to the next step. Otherwise, the algorithm goes to step 7.

Step 3 contains successive sets generation. Thus, derivatives of the word `cvt` are extracted first from eDCD. Next will be generated starting with the set D_{eDCD} other derivatives following the derivative models such as affix substitution, derivative projection, derivative constraints and formal derivational rules. As a result, we will get a set of automatic generated derivatives $D_{gen}=D_{CL}\cup D_{PD}\cup D_{SA}\cup D_{RD}$ (step 4).

In step 5, the set D_{gen} will be split automatically in three distinct sets: D_{NEVAL} (set of invalid derivatives) D_{SEMVAL} (set of "half" valid derivatives) and D_{VAL} (set of valid derivatives).

Step 6 consists of manual validation of the set words from D_{SEMVAL} and the addition of validated derivatives to D_{VAL}.

The extraction of the set of derivatives D_{VAL} is done in step 7.

4 Conclusion

Studies on derivation process allow us to conclude that we cannot propose an effective algorithm for automatic derivation in general, but we can highlight some models of derivation, for which construction of such algorithms is possible.

It is important not only to establish the cases when automatic derivation is possible, but also to identify those cases where a particular combination `<prefix><root/stem><suffix>` is impossible. In this respect were made a number of constraints that allow filtering a priori invalid words.

References

[1] E.Boian, A.Danilchenco, L.Topal. *The automation of speech part inflexion process*, Computer Science Journal of Moldova, vol.1,no.2(2),1993,pp.14-47.

[2] M. Petic. *Unele aspectele ale morfologiei derivaţionale în limba română*. Noua Revistă Filologică, vol. 3 (2011), pp. ___ - ___ – to appear.

[3] S.Cojocaru, E. Boian, M. Petic, *Stages in automatic derivational morphology processing*, in Knowledge Engineering, Principles and Techniques, KEPT2009, Selected Papers, Cluj-Napoca, July 2 – 4, 2009, pp. 97-104.

Mircea Petic

Institute of Mathematics and Computer Science, A.S.M., Chisinau, R. M.
E-mail: mirsha@math.md