Creation and Development of the Romanian Lexical Resources

Elena Boian, Constantin Ciubotaru, Svetlana Cojocaru, Alexandru Colesnicov, Ludmila Malahov, Mircea Petic
Institute of Mathematics and Computer Science,
Chişinău, Moldova
{lena,chebotar,svetlana.cojocaru,kae,mal,mirsha}@math.md

Abstract

This article describes the Romanian lexical resources containing morphological data and dictionaries: synonyms, Romanian-English, and Romanian-Russian.

The inflection process at the creation of morphological resources based on the functional grammar with scattered context is considered. An arbitrary word is inflected knowing only its part of speech, and the gender for nouns.

New words were obtained also by using prefixing and suffixing. The research in automated prefixing and suffixing permitted us to determine some word classes for which this method is applicable, and to implement the corresponding algorithms.

We describe the database structure, and the DB population programming tools.

The article describes an approach to the checking of integrity and correctness of the morphological resources presented as a database mapping Romanian words to their morphological derivatives.

1 Introduction

Creation and development of the lexical resources are important parts of Natural Language Processing (NLP).

One of such resources for the Romanian language are Reusable Resources for the Romanian Language (RRRL).[1]

This article describes how these lexical resources were created and developed, the database structure, and the corresponding programming tools. The morphological and, more specifically, the inflectional aspects are pointed out.

[1] http://www.math.md/elrr/

In Sec. 2, the implemented programs are described that populate the resources by automatic inflection (Boian and Cojocaru, 1996). The starting point for this approach was the book (Lombard and Gâdei, 1981), where most of Romanian productive classes of words (nouns, adjectives and verbs) were classified according to their inflection groups. The classification was made from the linguistic point of view, and, for example, the accents were taken into account. In this case, it is possible to operate only with the graphical representation of the word that equally simplifies and complicates the problem. Nevertheless, this classification was useful and led to the idea to formalize word-forms producing with the special grammar that is presented into Sec. 3. Other parts of speech (numerals, pronouns, articles, conjunctions, prepositions, interjections) were entered into the Database (DB) manually as being not so numerous.

It is shown also how the inflectional model for an arbitrary word can be determined (Sec. 4). Knowing this information it is possible to perform the inflection automatically (Sec. 5).

The proposed solution is not restricted by the Romanian language but could be also applied to other natural languages with inflectional mechanisms similar to these of Romanian.

Another way to populate the DB is affixing. New words were generated by prefixing and suffixing (Cojocaru et al., 2009). The research in the possibilities of automated prefixing and suffixing permitted to determine some word classes for which this method is applicable, and to implement the corresponding algorithms. This led to considerable lexicon extension (Sec. 6).

In Sec. 7, the structure of the DB is described, the relations between the main and auxiliary tables, and some techniques are discussed that were used to check the DB integrity and information correctness in maximally automated mode (Cojocaru et al., 2006).

Proceedings of Recent Advances in Natural Language Processing, pages 678–685,
Hissar, Bulgaria, 12-14 September 2011.

2 Acquisition of lexical resources

An important direction in NLP is acquisition of lexical resources. The problem of the automation of words inflexion process in Romanian was investigated in (Boian and Cojocaru, 1996). The obtained results permitted to construct an electronic lexicon RRRL containing the lemmas and their word-forms. Lexical resources acquisition is carried out by using static and dynamic methods for words inflexion.

Static methods use the morphological dictionary (Lombard and Gâdei, 1981), where the inflexion groups are indicated explicitly. The algorithm based on static method uses the formalism of the inflexion grammar for a natural language proposed in (Boian and Cojocaru, 1996).

Dynamic methods tried to find the inflexion model analyzing the word structure, and especially its affixes. These affixes were determined by examining of different lexicographic sources. Dynamic method attempts to calculate the inflexion paradigm using some classifications. The inflexion programs created on the base of these methods permit to generate approx. 90% of all Romanian inflexions. Sometimes the user intervention is requested to solve ambiguities.

3 Scattered Context Grammars for Vocabulary Generation

The scattered context grammar rules have the following form:

$$[/] * [\#][N_1]a_1\overline{b_1}a_2 \ldots a_{n-1}\overline{b_{n-1}}a_n \rightarrow a'_1\overline{b_1}a'_2 \ldots a'_{n-1}\overline{b_{n-1}}a'_n N_2,$$

where a_i, a'_i are arbitrary words, and either b_i is nonempty word, or the special symbol $*$ stands instead of b_i, admits arbitrary f_i. Numbers N_j are codes of the ending sets.

The interpretation of this rule is as follows. Let w be the base word to produce word-forms. Every slash / indicates cutting the last letter from w. The word v obtained after this is considered as a root (if N_1 exists) and N_2 is its index in ending set list L. In any case the word v should have the form

$$f_0a_1f_1a_2f_2 \ldots a_{n-1}f_{n-1}a_nf_n,$$

where every f_i is an arbitrary (possible empty) word, not containing (for $i = 1, 2, \ldots, n-1$) the veto sub-word b_i. Veto for b_i is conditioned by the necessity to determine the position of the subword a_i to be substituted. If there exists more then one representation of this kind the first one (scanning v from left to the right or vice versa if the sign $\#$ is present) should be selected.

Let us take the example word w ="*frate*" (Eng. "brother") that fits this case: masculine gender, singular number, indefinite form, is inflected using the rule M46 / 5 t→ţ 3. We have two sublists of endings for this word: $T_5 = \{e, e, e, ele, elui, e\}$ and $T_3 = \{i, i, i, ii, ilor, ilor\}$. The rule is interpreted as follows. First of all the last symbol of word w is deleted. It gives the root v_1 ="*frat*" that is concatenated with the set of endings T_5. One part of inflections is formed without alternation. The list of inflected words is: *frate, frate, frate, fratele, fratelui, frate*. Then the alteration of consonants $t \rightarrow$ ţ is performed in the root v_1. The obtained root v_2 is concatenated with the set of endings T_3. The list of the rest inflected words for v_2="*fraţ*" is the following: *fraţi, fraţi, fraţi, fraţii, fraţilor, fraţilor*.

The obtained inflected words for w="*frate*" are: *frate, frate, frate, fratele, fratelui, frate, fraţi, fraţi, fraţi, fraţii, fraţilor, fraţilor*.

Using such grammar rules, the process of creating of the decomposed vocabulary was formalized. The inflexion grammar for Romanian contains 866 rules and 320 ending sets. They were used to obtain a morphological lexicon using dictionary with about 30,000 lemmas (Lombard and Gâdei, 1981).

4 Description of the Inflection Process

Romanian is a highly inflected language. As we mentioned already, open productive parts of speech for Romanian are nouns, adjectives, and verbs. These open classes contain tens of thousands elements, and are characterized by a productive process of inflection, derivation and composition. In this case the problem is complicated not only because it is impossible enumerate the elements existing at the moment, but also because a successful formalism should be able to serve future neologisms that could occur in the language. In the following we operate with the paradigms of inflection, by which we understand the systematic arrangement of all inflection forms of a word.

We work not with the whole words, but with their variable parts, including roots and inflectional morphemes added to them. Below, we mention list of inflectional morphemes as the (inflec-

tional) paradigm.

An incomplete set of rules was shown in papers (Tufiş et al., 1996; Peev et al., 1996; Hristea and Moroianu, 2003). There, concatenation of inflectional morpheme for nouns and adjectives is performed not concerning the problem of the alternations in the root. Therefore, having the aim to achieve the model of inflection, we developed a formalism, which includes two processes: alternation in the root, and concatenation of an inflectional morpheme.

5 Determining the Inflection Group

We use the word spelling only to determine its inflexion group. The grammar rules define, in fact, the inflexion model on the algorithmic level: cutting a given number of symbols at the word ending; obtaining different roots by substitutions (in order to produce vowel and consonant alternation), attaching the corresponding morphemes (endings) to the roots.

This method can be applied only in the case when the inflexion group (inflexion model) is known. Otherwise, the problem appears of inflexion model calculation, knowing the graphical representation of the word. Is it possible to solve algorithmically this problem? The answer is negative. The first obstacle is the determination of part of speech: there are several examples of homonyms, which represent different parts of speech, e.g., *mare* (Eng. big) is an adjective, and *mare* (Eng. sea) is a noun.

Let us restrict the formulation of the problem: is it possible to establish the model of inflection (in the conditions indicated above) knowing the part of speech?

The answer is negative in this case too. For confirmation we can bring a list of examples, which show us that without invoking phonetic or etymological information we cannot determine the model of inflection. Let us illustrate this assertion by analyzing feminine noun *masă*. Following the meaning of furniture object we will form plural *mese*, using the model with vowel alternation $a \rightarrow e$. But if you follow the meaning "compact crowd of people", the plural *mase* should be produced without alternation. The origin of this phenomenon is etymological: in the first case the origin of the word is from Latin *mensa*, but in the second case from the French word *masse*. The problem might be tackled in another way: to establish some criteria which permit, after the analyzing of the word structure, to conclude about the possibility to determine the inflexion model and, if this is possible, to fix the specific model. Otherwise, we will try to formulate the criterion according to which one can affirm that the inflexion process can be performed automatically and denote the corresponding model.

Let we have a word (a lemma) in its graphical representation. We know the part of speech, and the gender in the case of noun. We divide all words into three categories:

irregular, the case being determined from a preset list of words;

absolutely regular, that admitting the automatic inflexion (a unique inflectional model can be calculated);

partially regular, those words which need some additional information except the graphical representation to be inflected, and calculation produces two or more inflectional models.

To simplify, we exclude from the examination the irregular words as their presence or absence does not affect the generality of the algorithm.

In (Cojocaru, 2006), the algorithm had been proposed, which analyses the dictionary of classification into morphological groups with entries of type (w, σ), where w is a word in natural language, and σ – number (label) of inflection group, constructs two groups of sets $A = \{A_1, A_2, \ldots, A_k\}$ and $P = \{P_1, P_2, \ldots P_s\}$, $\cap_{i=1}^{k} A_i = \emptyset$, $\cap_{i=1}^{s} P_i = \emptyset$. $A_i \cap P_j = \emptyset$.

These sets consisted of sub-words α_j of the words $w = w'\alpha_j$, where $1 \leq |\alpha_j| \leq |w|$. It is shown that for certain categories of words it is possible to construct such sets A_i, that from the fact that $\alpha_j \in A_i$ it results unequivocally that the word w belongs to the single inflection group σ, and these words being named "absolutely regular". With the help of the same algorithm there are constructed also such sets P_i, that from the fact that $\alpha_j \in P_i$ it results that $w = w'\alpha_j$ can belong to several inflection groups $\sigma_1, \ldots, \sigma_m$, and the respective words being named "partially regular".

5.1 Construction of Ending Sets

Let L be the set of all words of a language. We come from the assumption (valid for majority of natural languages) that there is a classification dictionary $D \subseteq L$, so that to any $\omega \in D$ it puts into

correspondence an inflectional model ν, where ν is a positive integer. We will present dictionary D as a union of words classified by parts of speech (and gender, for nouns), $D = \cup(C)_{i=1}^{5}$, where C_i is one of the sets of words of open classes: nouns (masculine, feminine, neuter), adjectives and verbs. For each C_i the dictionary D puts into correspondence the finite set of inflectional models $N_i = \{\nu_1, \ldots, \nu_{n_k}\}$, such that for $\forall\omega \in C_i$ there is at least a $\nu \in N_i$. We will separately operate with each of these classes.

Let C be one of these classes. The idea of algorithm to build the sets of endings is the following. For each word $\omega \in C$, to which the inflectional model $\nu_m \in N$ corresponds (N is the set of integers of inflectional models for words in C), the endings were built with decreasing lengths from $|\omega|$ to 1. The pairs (γ_i, ν_m) are formed, where γ_i is a substring of length i of the word ω, $(1 \leq i \leq |\omega|)$. The pairs, constructed thus, are compared and filtered. The filtration process is carried out in the following way.

Out of each two elements (γ_i, ν_m), (η_i, ν_n), we keep only one, if $\gamma_i = \eta_i$ and $\nu_m = \nu_n$, where γ_i is a substring of length i of the word $|\omega|$, and η_i is a substring of length i of the word ψ (i. e. only non-coincident pairs are kept).

If for all the pairs in which $\gamma_i \neq \eta_i$ the equality $\nu_m = \nu_n$ takes place, then the pairs (γ_i, ν_m) and (η_i, ν_n) are elements of the set A of the endings corresponding to absolutely regular words.

If $\gamma_i = \eta_i$ and $\nu_m \neq \nu_n$, then the ending η_i indicates a substring of the word ψ partially regular from the set P, to which several inflectional models $\nu_m, \nu_n, \ldots$ correspond.

We describe the filtration process using the next example. Let $D = \{$ *(grup, 1), (grup, 2), (dulap, 1), (cuvînt, 2), (vînt, 1), (tractor, 3), (muzeu, 41)*$\}$.

Initially $A = \emptyset$, $P = \emptyset$.

We take as C all the words from D, i.e.,

$C = \{$ *grup, dulap, cuvînt, vînt, tractor, muzeu*$\}$ (in English: group, wardrobe, word, wind, tractor, museum).

$L_{max} = 7$; $N = \{1, 2, 3, 41\}$.

We construct the sets of endings of the lengths $7, 6, \ldots, 2, 1$ of words from C, to which the inflectional models N are being put into correspondence.

Sub-words were sorted descendently at their lenghts:

$D = \{$ *(tractor, 3)* $\cup$ *(cuvînt, 2), (ractor, 3)* $\cup$ *(uvînt, 2), (actor, 3), (dulap, 1), (muzeu, 41)* $\cup$ *(grup, 1), (grup, 2), (vînt, 2), (vînt, 1), (ctor, 3), (ulap, 1), (uzeu, 41)* $\cup$ *(rup, 1), (rup, 2), (înt, 2), (înt, 1), (lap, 1), (tor, 3), (zeu, 41)* $\cup$ *(up, 1), (up, 2), (nt, 2), (nt, 1), (ap, 3), (or, 3),(eu, 41)* $\cup$ *(p, 1), (p, 2), (t, 2), (t, 1), (p, 3), (r, 3), (u, 41)* $\}$.

Then we obtain the sets A and P using above mentioned rules with the following components:

$A = \{$*(dulap, 1), (ulap, 1), (lap, 1), (ap, 1), (cuvînt, 2), (uvînt, 2), (tractor, 3), (ractor, 3), (actor, 3), (ctor, 3), (tor, 3), (or, 3), (r, 3), (muzeu, 41), (uzeu, 41), (zeu, 41), (eu, 41), (u, 41)*$\}$.

$P = \{$*(grup,1, 2), (rup,1, 2), (up, 1, 2), (vînt,1, 2), (înt,1, 2), (nt, 1, 2), (p, 1, 2), (t, 1, 2)*$\}$.

5.2 Determination of the Inflection Group

We determine the inflexion group for the word $\psi \in C$.

The algorithm for the inflexion group determination is the following.

The substrings ξ_i $(1 \leq i \leq |\psi|)$ of the endings with decreasing length from $|\psi|$ to 1 of the word ψ are constructed. Initially we look for a completely regular model, comparing the ending ξ_i $(|\xi_i| = i)$ with the elements $(\gamma, \nu_m) \in A$ $(|\gamma_i| = i)$. If $\exists\gamma_i = \xi_i$, then ν_m is the inflection model number.

In case if we did not find an appropriate model in A, we look for it in P. If $\exists\gamma_i = \xi_i$ $(\gamma_i, \nu_{n_1}, \nu_{n_2}, \ldots, \nu_{n_k} \in P)$, the word ψ is partially regular and it has to inflect in correspondence with the inflexion models $\nu_{n_1}, \nu_{n_2}, \ldots, \nu_{n_k}$.

In the case when $\xi_i \neq \gamma_i$ for $\forall\gamma_i$ from A and P the inflection model can not be determined automatically and the intervention of user (the expert in linguistics) is needed.

Reviewing the example of construction of ending sets A and P from the previous section, we can determine the inflectional group for the word *motor* (in English: engine).

We obtain that the word $w =$*"motor"* is inflected using the inflectional group 3. The substrings ξ_i $(1 \leq i \leq 5)$ of the endings with decreasing length from 5 to 1 of the word w are constructed: *motor, otor, tor, or, r*. Initially we look for a completely regular model, comparing the ending ξ_i $(|\xi_i| = i)$ with the elements $(\gamma, \nu_m) \in A$ $(|\gamma_i| = i)$ and *tor* as substring of word w and *tor* from $(tor, 3) \in A$ coincide, then 3 is the inflection model for $w =$*"motor"*.

Characteristics	Number
derivatives	15300
roots/stems	6800
prefixes	42
suffixes	433

Table 1: The tables characteristics

6 Prefixing and Suffixing

Existent electronic linguistic resources represent one of the important moment in the process of derivatives generator elaboration. In the case of the lexicons they are not simple repositories only of words, but they need to contain the prefixes and/or suffixes with their descriptions (Carota, 2006; Petic, 2010).

To work with affixing, we take the correspodent information from the electronic variant of derivatives dictionary (S.Constantinescu, 2008) (Tab. 1) and added four tables to the DB: *prefixes, suffixes, roots-stems-derivatives*, and the table which mapes affixes to roots/stems in order to form the derivatives. The last table consists of 3 fields destinated to prefixes and 4 for suffixes, because the electronic variant of derivatives dictionary has derivatives with maximum 2 prefixes, for example, dez/ră/suci (Eng. untwist), pre/în/noi (Eng. restore), or 3 suffixes, for example, loc/al/iza/re (Eng. localization).

With this structure attached to RRRL, it was possible to elaborate some queries that allow:

- derivative extraction by a prefix or suffix;
- lexical family extraction for a root or stem;
- the part of speech establishing of the derivatives and/or roots-stems;
- determining the alternations that are present in the process of derivation.

The lexicon completion can be implemented with the help of automatic tools (Cojocaru et al., 2009). Starting with the derivation rules, an algorithm which forms a set of words corresponding to the derivation constraints is going to be elaborated. This algorithm of derivation is applied to these words and the result is a set of derivatives. Therefore not all the derivatives correspond to the norms of human language. After applying the method of validation, we obtain correct words on the basis

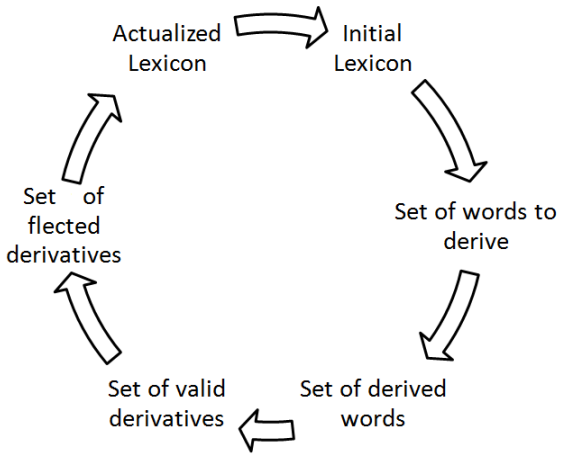

Figure 1: Cycle of the lexicon completion

of language. These words are inflected by means of programs for inflection (Boian and Cojocaru, 1996) that result in a set of inflected words. This veryfied set can complete the initial lexicon, making it actual (Fig. 1).

Nevertheless after a cycle of bringing the lexicon up to date it is possible to apply another similar cycle (Cojocaru et al., 2009). So, after a finite number of cycles it is likely to finish the process of completion, in the end obtaining a "filled" (saturated) lexicon which will be complete from the point of view of derivation.

7 Correctness and Integrity of the DB

Before to make lexical resources widely available we checked their correctness and integrity. We did it in maximally automated mode: using some programs to select suspicious information for ulterior correction by the operator or the expert in philology to make the final decision.

7.1 Structure of the RRRL DB

The Resources DB for RRRL has two main tables and a lot of auxiliary ones. Auxiliary tables contain different codes used in the main tables, e. g., codes of morphological characteristics or languages.

The *words* table contains the part code (part of speech) and field code (domain of the word usage) fields. Numerically encoded word in the flexies table marks the base word of flexies.

The *word_flexies* table contains the flexy word field keeping derivatives of Romanian words. Each derivative is associated with its base word in the *words* table through the integer prim word

code field. The integer *morpho_code* field substantiates morphological information (tense, number, case, etc.).

As for auxiliary tables, the *morpho_code* field is substantiated using not one single table but ten auxiliary tables in correspondence to ten Romanian parts of speech: noun, adjective, verb, numeral, adverb, pronoun, preposition, conjunction, article, interjection. The fields in these tables contain codes of Romanian morphologic categories corresponding to the part of speech.

The DB was populated from textual information files.

The DB population program produces a file that shows if words were inserted, word codes, and the result for each operation. Errors are marked and can be easily found. We also see how many words were entered and which words were not entered because they double the existing ones in the DB.

Textual information files were got by a semi-automatic program that generates all word-forms for a given Romanian word (Boian et al., 2005). The program is wizard-like and the input should be done by an expert linguist.

For the *word_flexies* table, each group contains one word-lemma with all its derivatives (word-forms). Encoded morphological information is included with each word-form.

7.2 DB Integrity

The building of a lexical resource is a difficult process. We tried to automate it maximally using specially developed programs. To deal with errors, a set of techniques was developed that are described below.

First of all, it is possible apply formal methods to check validity of the DB content. These methods can be formulated using the semantics and interdependencies of the DB fields and tables. In this purpose, all DB fields are divided in four categories:

1. fields containing textual representation of words;

2. fields containing references that connect different tables, e. g., codes of Romanian word-lemmas that replace words themselves in the *word_flexies* table;

3. fields containing morphological attributes;

4. fields containing textual representation (deciphering) of attributes; these fields only exist in the auxiliary tables.

Depending of the used DB engine, some formal relationships can be supported automatically,

Non-formal checking may be executed by variety of techniques depending on the field category. Foe example, the fields of the category 1 can be checked by usual spell checkers. For Romanian, there exists a spell checker RomSP (Malahova and Colesnicov, 1996). The corresponding list of Romanian words was carefully tested and updated both by developers and users of the product, and can be taken as being quite reliable. Romanian spell checker from MS Office was also used. For Romanian, words that were rejected by both spell checkers were marked as highly suspicious. The analysis show that most of them were erroneous. Other word lists can also be used, e.g., those coming with free spell checkers like ISpell.[2]

A different method of word checking supposes the selection of n-*grams* (word fragments of n letters, $n > 2$) from the given set of words, and calculation of their frequencies. Less frequent n-grams are considered to be suspicious. Words that contain such n-grams should be checked by experts.

7.3 DB Correctness

The next check is search for duplicates. The unique field of the *words* table is prim_ word_ code. The corresponding information consists of the Romanian word in its textual form, its part of speech, and its field of usage. These data are checked for uniqueness during DB population.

We saw that category 3 fields can be formally checked as containing in one of additional tables as the record number. The correspondence between fields of categories 3 and 4 can be checked informally using interval of values for different attributes but this is partial checking only. In any case, additional tables are short and can be checked visually. We can also search for unused codes in them. The correspondence of codes in the morpho. categories table and tables for each part of speech was checked by issuing requests that show in parallel decoded values of each code.

The next category of checks is search for duplicates. Our DB population programs query for existence of the information before its insertion into

[2] http://www.gnu.org/software/ispell/ispell.html

any of tables, therefore, absence of duplicates can be supposed. Meanwhile, search for duplicates can expose some errors in the prepared data for population of the DB, or in the DB population programs themselves.

In the *words* table, the unique field is *prim_word_code*. The corresponding information consists of the Romanian word in its textual form, its part of speech and field of usage. These data are checked for uniqueness during DB population. Non-unique combination found means something wrong with these programs, and we can check their codes visually for this combination.

Moreover, we checked the words table for uniqueness of word's textual form ignoring even its part of speech. In Romanian, adjective can coincide with adverb and noun can coincide with adjective, but such cases are relatively rare. This check permitted to detect several errors also.

Uniqueness of records in the *word_flexies* table is also checked during DB population. The corresponding check can be performed after population to test the DB population programs.

We performed also the following informal semantic checks.

Normally, Romanian words have some standard number of inflective derivatives depending of the part of speech: 12 for nouns, 20 for adjectives, and 35, 39, or 40 for verbs. We queried for the actual number of derivatives for words from the *words* table. For example, the result of the first such test for one of verbs was 160 derivatives. The impossible number of derivatives for some words permitted us to correct some errors. For example, it was found analyzing the case of verbs with more derivatives than necessary that some details of Romanian grammar were misunderstood during the design stage.

Parallel dictionaries are very useful and widely used in computer linguistics. Our DB contains translations of many Romanian words into English and Russian. We could not get sufficient results from the English translations. The Russian translations permitted us to formulate several useful criteria because Russian is a highly inflective language like Romanian. We used endings of Russian translations, that are more or less standard depending of part of speech, for:

- Check for words that are not verbs but Russian translations have "verbal" endings -ти -тись -ть -ться -чь -чься. We found 4119 of them, being mostly OK, but several errors were found.

- Check for words that are not adjectives but Russian translations have "adjectival" endings -ая -ев -ий -ин -ов -ые -ый -ье -ья. No such words were found.

- Check for words that are not adverbs but Russian translations have the corresponding endings -е -о -у -ем -ём -мя -ой -ом -ски. This check was not so successful (18974 words) but we shortened the result by deleting all verbs, adjectives, and nouns, and found several errors more.

As errors were found, they were corrected in the source data files. At a small quantity of corrections, erroneous records were deleted taking into account all interdependencies, and the corresponding part of the data file was entered anew. Having a lot of corrections, we populated anew the whole DB that takes quite acceptable time.

We do not enter specific field of usage for a word where we enter its morphological derivatives. In this case, the corresponding field is always set to 1 ("general"). Therefore, we can check for uniqueness of the combination of a word's textual form and part of speech and analyze the corresponding fields of usage and tables where are used "non-general" words. We created the list of uninflected words that coincide with some inflected pairs of text and part of speech, and the list of "truly" uninflected words.

Conclusions and Results

A computational lexicon for Romanian containing about 1 mil. words (obtained by inflexion of 100,000 lemmas) was constructed. The lexicon was used for different linguistic applications: the spelling checker for Romanian, the data base of linguistic resources, the search algorithm for web pages.

Certain criteria were established for a word that allow to determine which is its inflexion model, analyzing the word structure.

The derivation rules formalization for some Romanian affixes offer the possibility to elaborate algorithms for the lexical resources completion. The process of new derivatives validation is one that raises many questions and it seems that there are

solutions though there are some difficulties in this process. Thus, it is impossible to neglect the aspect of source credibility in the process of word validation. In this context the word validation using the existent corpora seems to be the best solution.

The automatic completion cycle model for lexical resources by the derivation and inflectional mechanisms allows the consciousness of the steps in the process of lexicon enrichment.

DB was selected as linguistic information stock because of possibility of quick parallel and distant access, flexibility of possible queries, wide use and availability of the corresponding programming techniques. Other forms of information presentation like, e. g., word lists, can be easily obtained from the DB. Applications can be developed using this DB directly or indirectly.

The information containing in the DB should be thoroughly checked using different techniques. A set of methods was proposed that were found useful in the case. The discussed techniques can be applied at checking of lexical information in other cases.

References

E. Boian and S. Cojocaru. 1996. The inflexion regularities for the Romanian language. *Computer Science Journal of Moldova*, 4(1):40–58.

E. Boian, A. Danilchenco, and L. Topal. 1993. The automation of speech parts inflexion process. *Computer Science Journal of Moldova*, 1(2):14–26.

E. Boian, C. Ciubotaru, S. Cojocaru, A. Colesnicov, V. Demidova, and L. Malahova. 2005. Lexical resources for Romanian. In *Scientific Memories of the Romanian Academy*, volume 26 of *IV*, pages 267–278. Bucharest, România.

S. Cojocaru and E. Boian. 2010. Determination of inflexional group using P systems. *Computer Science Journal of Moldova*, 18(1(52)):70–78.

S. Cojocaru, M. Evstiunin, and V. Ufnarovski. 1993. Detecting and correcting spelling errors for Romanian language. *Computer Science Journal of Moldova*, 1(1):3–22.

S. Cojocaru, A. Colesnicov, and L. Malahova. 2006. Integrity and correctness checking of a lexical database. *Computer Science Journal of Moldova*, 14(1(40)):138–151.

S. Cojocaru, E. Boian, and M. Petic. 2009. Stages in automatic derivational morphology processing. In *Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques KEPT2009*, pages 49–53, Cluj-Napoca (Romania), July 2–4.

F. Carota. 2006. Derivational morphology of Italian: Principles of formalization. *Literary and Linguistic Computing*, 21(Suppl. Issue).

M. Petic. 2010. Developing a derivatives generator. *Computer Science Journal of Moldova*, 18(1(52)):82–96.

D. Tufiş, L. Diaconu, C. Diaconu, and A.M. Barbu. 1996. Morfologia limbii române, o resursă lingvistică reversibilă şi reutilizabilă (Morphology of Romanian, a reversible and reusable linguistic resource). In *Limbaj şi Tehnologie*, pages 59–65. Editura Academiei Române, Bucureşti. (in Romanian).

S. Cojocaru. 2006. The ascertainment of the inflexion models for Romanian. *Computer Science Journal of Moldova*, 14(1(40)):103–112.

1998. *Dicţionarul explicativ al limbii române (The Explanatory Dictionary of the Romanian Language)*. Academia Română, Institutul de Lingvistică "Iorgu Iordan", Editura Univers Enciclopedic. (in Romanian).

T. Hristea and C. Moroianu. 2003. Generarea formelor flexionare substantivale şi adjectivale în limba română (Generation of flexional forms for nouns and adjective in the Romanian language). In F. Hristea and M. Popescu, editors, *Building Awareness in Language Technology*, pages 443–460. Editura Universităţii din Bucureşti, Bucureşti. (in Romanian).

S. Constantinescu. *Dicţionarul de cuvinte derivate*. Editura HERRA, Bucureşti, 2008.

D. Irimia. 2004. *Gramatica limbii române (The Grammar of the Romanian language)*. Polirom, Bucureşti, 2 edition. (in Romanian).

A. Lombard and C. Gâdei. 1981. *Dictionnaire morphologique de la langue roumaine*. Editura Academiei, Bucureşti.

L. Malahova and A. Colesnicov. 1996. Implementation of the Romanian Spelling Pack for Windows. In *The International Conference on Technical Informatics CONTI'96. Proceedings. Computer Science and Engineering*, volume 1, pages 23–28, Timişoara, România.

L. Peev, L. Bibolar, and E. Jodal. 1996. Un model de formalizare a morfologiei limbii române (A formalization model of Romanian morphology). In *Limbaj şi Tehnologie*, pages 67–72. Editura Academiei Române, Bucureşti. (in Romanian).