Computer Science Journal of Moldova, vol.**5**, no.3(15), 1997

A method of Romanian Lexicon compression

C. Bajireanu

Abstract

Using the special uniform character to represent the alternations it is possible to compress the computational lexicon. The proposed method is illustrated on the example of Romanian lexicon.

This work deals with the computer representation of natural languages lexicons. In particular here is presented a method of compression for Romanian language lexicon. The proposed method is based on substitution of most used alternations by special uniform characters.

In order to make clear the presentation hereby, we would like to introduce some definitions.

Definition 1 *The lexicon is a lexical component, which lists lexical items (indivisible words and morphemes) in their underlying forms, and encodes morphotactic constraints [1].*

Definition 2 *The word is a sequence of letters, independent in their existence in the real language [2].*

In elaborating an effective structure of the natural language lexicon we defined several points to be achieved.

A Flexible structure.

The possibility to add in new information or to change the structure of the database without affecting functions of database maintaining.

This feature will make possible to easily customize the database and as a result a base element for different linguistic applications (translators, speech recognition systems, spell checkers etc.) will be obtained. Also, such kind of database is easy to use in other fields: training programs, encyclopedia, language learning programs etc.

B Multi indexing.

The possibility for searching information by different keys, not only the keyword but also by morphological, lexical or other information.

It is intended to store different kinds of information in the database, not only words, but additional morphological and other information. All this information should be organized in the form which will allow powerful possibilities of searching by different keys. As a result, the search function will be also able to return sets and categories of words, the fact which will make possible the statistical analysis of the language. Example: Search all masculine nouns which contain the diphthong oa and starts with a consonant.

C Minimum entries, compact representation and high access speed.

Avoid multiple entries for the same root word-forms. Obtain high level of compression based on morpho-lexical nature of the database.

1. Store the alternations as a special character. In Romanian, as well as in other languages, there are a lot of alternation letters (for example, *d-z, a-ă, e-i,* etc.). This fact creates an inconvenience in storing such words. In the proposed structure, following [3], some special characters are introduced in addition to the alphabet letters to replace the alternations. These characters have the meaning of both letters of the replaced alternation. For example: in case of alternation *d-z* (*ed - ezi*) the alternation character $ch1$ is introduced ($ch1 = d, ch1 = z$), so a new word is obtained $e[ch1], e[ch1]ului - e[ch1]ilor$. By using alternation characters, it is possible to reduce the number of entries in

databases and make the contents of the database uniform.

2. Base-word structure.
   In Romanian there are several ways of word creation. By these ways it is possible to create a group of words which contain particular parts of speech.
   Example:
   verb *a citi* (to read)
   noun *citire* (the process - reading)
   noun *cititor* (the person who performs the process - reader masculine)
   noun *cititoare* (same as above - feminine).
   In this case it is reasonable to store only the first word (base-word) and the rest are to be created dynamically according to corresponding rules. This will reduce the number of entries in the database and database size and will increase the speed of searching.

In this work a proposal related to the point C.1 is presented.

The word database from the Romanian Spelling Checker (ROMSP) can serve as an example of practical realization of computer lexicon. This program was developed at the Institute of Mathematics [4].

The structure of the word database used in ROMSP is based on the fact that the word is divided into three parts:

1. The first two characters are saved separately.

2. The rest of the root. This part was stored in a coding form. The 5 bits coding algorithm was used. The length of this part was 5 bytes, that makes possible to store 8 letters.

3. The index of the valid termination set for this root.

From the technical point of view the database of words consists from base elements which correspond to each root. Each element is composed from three parts: the length of the original word (1 byte) used for correct decoding, an array to store the packed root (5 bytes) and the pointer to corresponding termination set (2 bytes). It is necessary to mention that in the ROMSP word database the 5 bits coding algorithm was used to compress the letters. As a result a high level of compression was obtained.

In RomSP's word database all words with alternations are devided into roots which contain each variant of the alternation. For instance, the word “brad” with alternation ”d/z” which appears in the plural – “brazi” is stored as a couple of roots “brad” and “braz”, and the complete set of terminations (paradigm) is devided into two semi-paradigms. In other words, in case of alternation there will be 2 entries in the database for corresponding word. Respectively, when the word contains 2 or 3 alternations it will generate 3 and 4 entries in the database.

This causes some inconvenient factors that influence on the size of the database and the speed of search:

1. There will be more entries in the database then real words.

2. The use of semi-paradigms increases significantly the number of flexion rules and terminations sets.

3. There will be some different entries between alternation roots (brad ... brav ... brazi), so the search process will perform a lot of useless comparisons until the right word is found.

It is proposed to introduce special uniform characters for alternations. This means that some additional codes in the alphabet will appear. So, 5 bits will not be enough for representation of the alphabet. It is proposed to use the 6 bits coding. It will make possible to store all 31 letters of the Romanian alphabet and additionally 33 most frequent alternations.

There are approximately 53 alternations in Romanian language [5]. Some of them have pronunciation origin (c[k] - c [ts]) and have no difference in graphic representation. There are also complex types of alternations that can be reduced to one simple alternation (s/ş, st/şt, str/ştr → s/ş). Some of the alternations can be excluded because of their archaic origin (z/j) or very rare usage in limited number of words (ss/ş).

For statistical analysis a medium dictionary [5] was used. It contains 31638 words. From this amount 8911 words contain alternations (including 809 with 2 alternations and 20 – 3 alternations).

All words are classified into 808 groups (M – 95 groups, 8808 words; F - 164 groups, 6782 words; N - 88 groups, 7763 words; A - 122 groups, 3624 words; V - 247 groups, 4468 words; P -92 groups, 193 words).

After statistical analysis of the given dictionary the rating alternations by frequency was obtained (Tab.1).

Alternation	Frequency	Alternation	Frequency
t/ţ	1709	c/s	21
o/oa	1562	g/gh	20
e/ea	988	e/a/ă	19
s/ş	951	t/ţ/s	13
e/a	691	e/i	11
d/z	463	c/t	11
a/ă	396	(zero)/s	8
l/(zero)	300	g/s/t	5
sc/şt	200	p/ps/pt	5
e/ă	95	o/u	4
c/ch	53	c/ps/pt	4
n/(zero)	47	d/z/g/s	3
i/î	46	g/ps/pt	3
g/s	42	h/s	3
d/z/s	35	î/îi	3
o/oa/u	27	r/(zero)	3
x/cş	27		

All pronunciation alternations were excluded. Also were excluded very rare and archaic alternations. Following this procedure, the first 33 alternations were selected for codification by uniform characters in the 6 bits code table. As calculations show, this sets of alternations will cover 99.7% of all cases.

Let suppose that G_i is a group of words placed on page i and contains k words:

$$G_i = \{w_{i1}, ...w_{ik}\}.$$

Each word is characterized by its length $|w_{ij}| = m_j$; where $w_{i1}, ..., w_{ir1}$ are one root words (without alternation); $w_{ir1+1}, ..., w_{ik}$ are two roots words (with alternation).

Lets suppose that all roots have the first two letters unchangeable, all words of the group Gi are on the same page, alternations don't change the length of the root.

In this case, for 5 bits codification the volume of memory necessary for G_i will be:

$$\begin{aligned} L_5 &= \sum_{j=1}^{r_1}(m_j - 2) * 5 + 24r_1 + 2\sum_{j=r_1+1}^{k}(m_j - 2) * 5 + 2 * 24(k - r_1) \\ &= 5\sum_{j=1}^{k} m_j + 5\sum_{j=r_1+1}^{k} m_j + 28k - 24r_1; \end{aligned}$$

In case of 6 bits codification this amount of memory will be:

$$L_6 = \sum_{j=1}^{k}(m_j - 2) * 6 + 24k = 6\sum_{j=1}^{k} m_j + 12k;$$

So, the difference is:

$$L_5 - L_6 = 4\sum_{j=r_1+1}^{k} m_j - \sum_{j=1}^{r_1} m_j + 16k - 24r_1;$$

It is necessary to mention, that the estimation was made by supposing that the alternation doesn't change the length of the word. In the case when this condition is not respected, L_5 should be increased with $(k-r1)*5$ bits, but in case of 6 bits codification L_6 will remain invariant for this modification. The same results will be obtained in the case when more than one alternation are present in the word, respectively more than two roots. L_5 should be increased by the value needed the extra roots to be stored, at the same time L_6 will remain unchanged.

Using the uniform characters for alternation substitution, the number of entries of ROMSP word database can be reduced by 25%. This will influence considerably upon the size of this database. The numbers below show the advantage.

In case when there is no alternations substitution and 5 bits coding there will be 25196 entries without alternations, 12884 entries will be caused by the presence of one alternation, 1596 – 2 alternations and 60 – 3 alternations. The total number of entries will be 39736.

In case of using alternations substitution characters and 6 bits, only 52 entries caused by the presence of 26 rare alternations will remain. The total number of entries will be 31664.

As was mentioned above, 5 bytes are used to store the part of root without the first two letters. In case of 6 bites coding, 6 bytes will be needed. So the amount of memory needed to store roots of all entries:

for 5 bits coding –	198680 bytes
for 6 bits coding –	189984 bytes

But, by excluding unnecessary entries the whole records will be excluded too. That means, that each excluded entry will reduce the size of database by 3 bytes (according to the record element structure presented above).

Finally, we will obtain that for the given dictionary containing 31638 words an amount of 317888 bytes in case of the first method is necessary and 284986 bytes using the second method. This gives approximately 11.6% size reduction.

It is necessary to mention that the used dictionary was composed specially to show examples of all alternations. It includes most of words with rare type of alternation. It is supposed, that on transposition of this method on the complete Romanian language dictionary the number of alternations that are not substituted will not change significantly. On the other hand, there will be more words with substitutable alternations. It is estimated that the compression index will be around 15%.

Also, the number of termination sets will be significantly reduced. And most of terminations sets will correspond to whole paradigms. By this, a database with a more natural structure will be obtained.

The proposed solution will exclude the separation of alternation roots that will increase the speed of searching and generating words.

References

[1] Evan L.Antworth. Introduction to Two-Level Phonology. Summer Institute of Linguistics.

[2] S.Cojocaru, M.Evstunin, V.Ufnarovski. Detecting and correcting spelling errors for the Romanian language. Computer Science Journal of Moldova, vol.1, no.1, 1993.

[3] Gr.Moisil. Probleme de traducere automată. Conjugarea verbelor în limba romàna. SCL XI, 1960, nr.1.

[4] S. Cojocaru, "Romanian Lexicon: Tools, Implementation, Usage", in Dan Tufis&Poul Andersen (eds) "Recent Advances in Romanian Language Technology", Editura Academiei, 1997,ISBN973-27-0626-0, pp107-114

[5] A.Lombard. C.Gâdei. Dictionaire morpholique de la langue roumaine. Bucureşti, Editura Academiei, 1981.

C. Bajireanu
Institute of Mathematics,
Academy of Sciences of Moldova,
5 Academiei str., Kishinev,
MD 2028, Moldova
phone: (373–2) 738073

Received October 14, 1997