A LARGE RUSSIAN MORPHOLOGICAL VOCABULARY FOR IBM COMPATIBLES AND METHODS OF ITS COMPRESSION

Igor A. BOLSHAKOV
VINITI, Academy of Sciences of USSR
Moscow 125219, Baltiyskaya ul. 14, USSR

There are only few Russian vocabularies in computerized form in the USSR now, so development of a new Russian vocabulary large enough for spell checking is still topical.

The requirements for such a vocabulary are at least as follows : 1) more than 100,000 lexemes included; 2) modern and diversified lexicon well covering the sciences, many technological fields, the humanities, and may be the everyday life; 3) mapping the most of numerous lexeme forms implied by the flectional nature of Russian, and at the same time acceptance of well-formed words only; 4) orientation to IBM-compatible PCs most commonly used in the USSR noway.

Such a vocabulary has been recently built by the author. Its parameters are as follows: 67,400 stems covering more than 104,700 Russian lexemes and their 1.425 million word-forms (i.e. 21.2 forms/stem); the minimal, the mean, and the maximal stem lengths amounting to 1, 7.8, and 32 letters accordingly; the textual form size being about 865 KB.

Our morphological classification of stems is quite original and deals not only with word formation, but also with word derivation. The scheme includes 118 classes and 1901 various flections (variable suffixal chains). Separate classes were introduced among mentioned ones for invariant words, irregular forms, and abbreviations. The first 38 classes cover more than 83% of all stems.

The split borders of stems were freely moved to the left while classifying, if morphological alternations or identical final letters in a whole stem class have been encountered. The shortest flection is an empty one, the longest flections include up to 12 letters (e.g. ИРОВАВШИМИСЯ), so the mean flection length grew up to 6 letters, which is comparable to the mean stem length.

The textual form of vocabularies is not convenient for applications and has to be transformed into binary working form. The well known archivization packages such as PKARC/PKXARC are not acceptable for this purpose because of low squeeze ratio and uselessness of the archivized form as a working one for spellers or any other application. So several other methods of compression were analyzed.

Basically the Huffman method has been selected for coding morphological class numbers, and the Cooper method has been picked up for the stems. Additionally the RADIX-50 method was applied to both of the components of a vocabulary entry.

Several other techniques are turned out to be useful for additional stem compression in large vocabularies. They are based on 1) frequent recurrences of differently classified, but literally identical stems; 2) commoness of events in nearly saturated vocabularies, when the first letter in the deflecting part of a stem is alphabetically adjacent to the letter in the same position within previous stem; 3) availability of several free positions in RADIX-50 code table (only 33 of 40 are grasped by Russian letters and a delimiter). These unoccupied values might be used for re-coding final stem letters, digrams, and trigrams most frequent in different stem classes. This technique squeezes the letter part of a vocabulary entry and make the delimiter preceding the next entry unnecessary.

All methods mentioned were investigated, separately and in combinations. The Huffman's + the Cooper's + RADIX-50 combination has given us a squeeze ratio about 3.4, whereas addition of the rest techniques has incremented the ratio up to 4.2 - 4.5. So only about 190 KB in memory is needed for this working form, which is easy allocatable as a resident part of a modern text processor. As compared to vocabularies in available English language spellers, the size achieved seems to be highly competitive in our more complex inflectional case.

The vocabulary is available both in the textual and in binary forms. Several utilities concerned with its compiling, debugging, and squeezing are ready too. The utilities were written using Turbo Pascal 5.0 and Turbo Professional packages and are wholly applicable for processing any other natural language vocabulary.