Spelling-checking for Highly Inflective Languages

Jan Hajič, Januš Drózd***

*Research Institute of Computing Machinery
Loretánské nám. 3
Praha 1
Czechoslovakia

**Computer Centre of the Czechoslovak Academy of Sciences
Pod vodárenskou věží 2
Praha 8
Czechoslovakia

Abstract

Spelling-checkers have become an integral part of most text processing software. From different reasons among which the speed of processing prevails they are usually based on dictionaries of word forms instead of words. This approach is sufficient for languages with little inflection such as English, but fails for highly inflective languages such as Czech, Russian, Slovak or other Slavonic languages. We have developed a special method for describing inflection for the purpose of building spelling-checkers for such languages. The speed of the resulting program lies somewhere in the middle of the scale of existing spelling-checkers for English and the main dictionary fits into the standard 360K floppy, whereas the number of recognized word forms exceeds 6 million (for Czech). Further, a special method has been developed for easy word classification.

1. Introduction

After some delay, personal computers are now widely available in countries speaking Slavonic languages. Of course, they are used, i.a., for text processing. After solving the obvious problems with national alphabets (most of which are unfortunately not included in the standard IBM character set), the demand for a spelling-checker followed. The problem with Slavonic languages in general and with Czech in particular is that they consist of millions of word forms, thus the space needed for storing all of them directly grows over an acceptable boundary (whereas a typical Czech noun without "direct" derivatives has 7 different forms, an adjective could have 80 forms and a verb, which typically forms a dozen of derivatives - multiplied by ten or so possible prefixes - more than 5000).

Then, two methods are available to overcome this problem:

1) to compress the forms somehow, still allowing fast access;
2) to use linguistic knowledge about the regularities of the morphological behaviour of the words.

The first method fails after some investigations, even when considering some probabilistic models (which, using the multiple bit hash tables method (Fiala, 1986) with probability of false answers below 0.0005, cannot use less than 2 bits per word form stored).

Using the knowledge collected over generations of Czech linguists (e.g, Havránek and Jedlička, 1963; Slavíčková, 1975) and especially the latest works of the Prague group led by prof. P. Sgall (Panevová et al., 1981; Weisheitelová, Králíková and Sgall, 1982; Kirschner, 1983) we adapted the second method for the purpose of a spelling checking program to meet the competing requirements on space, speed and completeness.

2. The Model of Inflection

First, we decided to exclude the phonology level which is usually part of a morphological processing, because of the time penalty it would cause during processing. This means that all the phonological changes, although some of them are really regular, have to be treated in a single processing step together with the morphotactics. The space increase caused by this decision is still acceptable (for Czech, and, as far as we know, for the other Slavonic languages too).

The basic model of inflection we use assumes that a word form is a concatenation of a stem and an ending. For this purpose, we had to define the terms *stem* and *ending* in the following "computational" way to suit our purposes: the term *stem* means for us the part of the word which does not change in the course of inflection, the term *ending* means the part of the form which, when appended to the stem, completes the stem to a meaningful form. Exactly this model is used for nouns.

For verbs, it is suitable to extend this basic model to cover negation, as the negation is formed by the prefix *ne-*. Moreover, as a spelling-checker does not need to use the meanings of the words, we extended the word form definition further to cover verb prefixes. Of course, it is not economical to consider all possible verb prefixes, because most Czech verbs can have 3 to 8 derivatives by prefixes only. We use a compromise of 15 most frequent verb prefixes. All the other, as well as their combinations, are considered to be part of the stem as defined in the previous paragraph.

Our system uses two types of adjective structure. First, proper adjectives are viewed as consisting of a stem and an ending and possibly the superlative prefix (*nej-*) and/or the negative prefix. Second, verbal adjectives can have a verbal prefix in addition to the parts mentioned above. The latter type of partitioning is the most complicated one in our system.

For example, the form *nejnevykupovávanější* (lit. 'not the (item which is) mostly bought for speculative purposes iteratively') consists, from the point of view of our model, of five parts: the superlative prefix *nej-*, the negative prefix *ne-*, the "speculative" prefix *vy-*, the stem (of "to buy") *kup* and the ending *ovávanější*, which combines the functions of iterativeness, passive, comparison, and nominative singular.

Thus, we had to employ 240 sets of endings. Of course, there are also hundreds of exceptions. For them, as well as for indeclinable word classes, there is a special set consisting of a zero ending and the whole form is stored, i.e., in our terms, the whole form is considered to be the "stem".

3. User Interface

As the Czech users (not differing from their foreign colleagues in this respect) do not like learning a new text processors, we decided to follow the ideas behind Turbo Lightning. This way, using a memory resident program which is user-configurable to different text processors, we obtained a unified interface for virtually all users.

The basic functions of interactive single word/page check and/or correction are accompanied also by batch functions, which are preferred by some users for longer texts and some types of text processors. The types of texts supported by the batch mode range from simple ASCII files to files produced by WordPerfect 5.0, including the source texts for the TEX typesetting system.

The system also facilitates the process of adding word forms to the user's own dictionary. Due to the reasons discussed above, this causes problems, as the other forms of that word cannot be included fully automatically. An algorithm exists (see below) how to accomplish this task with the user's assisstance. The idea is similar to Finkler and Neumann (1988), though simplified for our purposes; Carter (1989) in his VEX system also uses the method of giving simple questions to the user (supposedly non-linguist) to learn about word's behaviour, but it is for English and primarily intended for assigning syntax properties rather than morphological. The implementation of the algorithm together with its user interface will be included as an off-line utility (in the first version, available in autumn '89, there was no such utility; it should be included in the second version).

4. The Semi-automatic Word Classification

Equipping the lexical entries with morphological information is an unpleasant task; very boring for linguists, and error-inducing for anybody. And if the dictionary is to be updated primarily by non-linguists, the need for (at least some) automation is obvious.

Fortunately, some inflectional languages (including Czech, as well as the other Slavonic languages) tend to indicate their morphological properties by (some of) the forms of the word itself, at least statistically.

As our purpose is to facilitate morphological classification of new words which are added to a dictionary, and as newly coined words or technical terms not included in the main dictionary are mostly regular, we can suppose that the irregular words are already in the dictionary.

When classifying a given word from the user dictionary (added to it during the on-line checking/correcting process), the user should first change the ending of the form moved here from the text to create the dictionary form of the word, i.e., nominative singular for nouns, nominative singular masculine for adjectives, and infinitive for verbs. In some cases, the system can provide the dictionary form automatically, but mostly the only help it can offer is to position the cursor under the last character of the word form.

Then the user should select the basic class to which the word belongs: indeclinable, verb, adjective or noun. There are no other questions for indeclinables, of course. For adjectives, the only further decision concerns the possibility of creating its comparative and/or negative forms. For verbs, the user should do two things: first, select all possible prefixes from the 15 prefixes handled by the system, and then, assign perfective/imperfective/both flag to the word and to its prefixed forms (for all the prefixed forms, this flag has the same value). For nouns, where the situation is very complicated, there is a hierarchy of questions and selections, which, for some masculine inanimates, reaches the level of five questions/selections. Fortunately, thanks to lots of investigations performed by mathematical and statistical linguists in the past, we can arrange things so that in most cases the first selection displayed is the right one.

For an experienced user, there is the possibility of writing directly the name of the appropriate class. We used this mode of operation when entering all regular Czech nouns into the dictionary.

Then the system constructs the stem and assigns the set of endings and prompts the user to confirm the resulting set of forms.

For example, when classifying the form *radionuklidy* (radionuclides), first the user deletes the ending *-y* (which is one of the plural endings). Then he/she selects "noun" as the basic class; then "masculine inanimate" is the right choice. Then, he/she should select *radionuklidu* as the right form which can follow the preposition *bez* (without), and state that *radionuklida* is not correct in this case. The last selection concerns the preposition *o* (about), after which *radionuklidu* is the only possibility (as opposed to the form *radionuklidě*, which cannot be used after the preposition *o*). Using this information, the system is able to decide that the stem is *radionuklid* (i.e., it equals to the nominative singular form) and the set of endings has the identification **hd1**. The user then confirms that *radionuklid,-lidu,-lidem,-lidy,-lidů,-lidům,-lidech* are the all and only correct forms of *radionuklid*.

5. Implementation

As mentioned above, we selected the memory resident version as the primary way of operation. The program, together with the cca 7,000 most frequent Czech words, takes approximately 110K of memory. It is able to check one screenful of a 60 column standard text (approx. 200 words) within 3 seconds on a 10 MHz PC AT with a 28msec hard disc. When the program runs as an ordinary program (in the mark- only batch mode), it is possible to have almost all the dictionary entries in main memory, and then it runs more than five times faster (100K of text in less than one minute).

The size of the main dictionary was in the first version, covering 80.000 - 100.000 Czech "dictionary" words, approximately 290K (not counting the 7000 most frequent ones, which reside in the memory anyway). This means that it can be used even on the oldest floppy based systems, e.g., in high schools. Since October 1989, the system is available for anybody wishing to avoid misprints when writing in Czech.

6. Conclusions

In the project described in this paper, the main topics were:

1) the design of the inflectional model, which will allow for a very fast parsing;
2) the design of techniques for storing the dictionary of stems together with the inflection classes in a compressed form, still allowing fast access;
3) the design of methods for allowing the user to add words with complete inflectional information to the dictionary.

We do not claim that there are no better solutions, but the resulting system has been accepted by its users both from the space as well as time point of view. However, the users (after some time of an excitement from their new toy) demand very soon the system marks false agreement (very common error in Czech), the word "farm" when used instead of "form" (the Czech words almost equal to these two English ones), etc. ... Could anybody think of a simple yet clear explanation to be given to them why they should still wait a little?

References

Carter, D. M. (1989). Lexical Acquisition in the Core Language Engine. In: *Proceedings of the 4th European Chapter ACL Conference,* ACL. Manchester. Great Britain. April 1989. pp. 137-144.

Fiala, P. (1986). Počítač v roli češtináře. (The computer as a teacher of Czech). In: *Proceedings of SOFSEM'86, Vol. II.* UVT UJEP Brno. JCMF. Liptovský Ján. Nízké Tatry. 1986. pp. 187-190. In Czech.

Finkler, W. and G. Neumann (1988). *MORPHIX - A Fast Realization of a Classification-Based Approach to Morphology.* Bericht Nr. 40. XTRA. KI-Labor am Lehrstuhl für Informatik IV. Universität des Saarlandes. Saarbrücken. 1988. 11 pp.

Havránek, B. and A. Jedlička (1963). *Česká mluvnice.* (The Czech grammar). SPN Praha. Prague. 1963. 2nd ed. 561 pp.

Kirschner, Z. (1983). MOSAIC - A Method of Automatic Extraction of Significant Terms from Texts. In: *Explizite Beschreibung der Sprache und automatische Textbearbeitung X.* Internal publications MFF UK Praha. Prague. 1983. 124 pp.

Panevová, J. et al. (1981). Lexical Input Data for Experiments with Czech. In: *Explizite Beschreibung der Sprache und automatische Textbearbeitung VI.* Internal publications MFF UK Praha. Prague. 1981. 160 pp.

Slavíčková, E. (1975). *Retrográdní morfematický slovník češtiny.* (Retrograde morphemic dictionary of Czech language). Academia Praha. Prague. 1975. 648 pp.

Weisheitelová, J., Králíková, K. and P. Sgall (1982). Morphemic Analysis of Czech. In: *Explizite Beschreibung der Sprache und automatische Textbearbeitung VII.* Internal publications MFF UK Praha. Prague. 1982. 120 pp.