

Lemmatization of Slovak words by a tool Morphonary*

Stanislav Krajčí, Róbert Novotný
stanislav.krajci@upjs.sk, robert.novotny@upjs.sk

Institute of Computer Science, Faculty of Science,
UPJŠ Košice, Slovakia

Abstract. Using a simple observation that the declination of Slovak word depends on its ending mainly, we have implemented the exchanging algorithm, the task of which is finding a lemma of a given word form. In the beginning of this process, the presence of this word form is checked in our list of (basic forms of) Slovak words (which is almost complete due to the official Slovak dictionary, *Slovník slovenského jazyka* [5]). If this word form is not found in this list, it is searched in a list of guides – declined word forms with pointers to their basic forms, i.e. lemmas. And if this process is not done yet, our exchanging algorithm is used for finding a word with the same ending and then for deriving a probably lemma of the given word. The existence of such probably lemma is checked in the above-mentioned list of Slovak words and in the negative case it is repeated.

1 Starting points

In the full-text document information retrieval documents and words are stored in the so-called vector (or object-attribute) model, i.e. a rectangle table, fields of which storing (either binary or more complex) information about the presence of the word representing by a column in the document representing by a row of this table. In this process it is very important and useful to associate different forms of the same word to one group and to pick its suitable representant which can be the stem of word or its lemma.

In the project NAZOU (Tools for Acquisition, Organization and Presenting of Information and Knowledge) we are interesting in two languages – Slovak and English – which are totally different from this point of view. English used the declination in very simple way and hence have very few forms of its words. Rules of making word forms from the common basic form (usually the stem) are very well definable, the number of exceptions (e.g. irregular verbs) is relatively small. An usual method for this associating is a stemming implemented in Porter’s algorithm ([4]) which is based on removing of prefixes and suffixes (like “-ed”, “-ing”, “-s”, “pre-” and so on).

* Partially supported by a state project of research and development Building of information society, Tools for Acquisition, Organization and Presenting of Information and Knowledge, 1025/04.

On the other hand, Slovak (like all Slavic languages) belongs to the so-called flexion languages, where the great part of words have many (dozens of) forms which are made by hundred ways. The time investment to looking for and applying rules of the declination would be very big, probably bigger than (time-consuming, but one-off) finding of all possible forms of all possible words. It is worth to commemorate very interesting work [3] of a former Slovak mathematics linguist E. Páleš, which made an attempt in this way, but his effort was not successful (in this sense). Looking of all possible word forms was not thinkable long time because of mankind bias towards classical paper. Recall that the official Slovak dictionary, *Slovník slovenského jazyka* [5] has six rather thick volumes, moreover it contains the lemmas only. But if we consider that the number of these words is 150,000 which have together about a few million forms only, the task of their storage in some database in our times is not problematic in the point of room – they (including their explanations and pointers to similar words) would occupy less than 1 CD!

Having this idea in mind, in the beginning of 00-ies students of our Institute of Computer Science made under our supervision an electronic versions of the above-mentioned *Slovník slovenského jazyka* [5] and, moreover, of the dictionary of about 60,000 words of the foreign origin ([7]), of course with many, many errors. After our careful (but till now not complete) refining now we have almost complete lists of these words. Of course, we understand that Slovak language is changing day-by-day and we know about new official contemporary Slovak dictionary *Slovník súčasného slovenského jazyka* [6] (till now the first volume is published) but we are convinced that “our” lists are good starting point for our next research in this fields. Our ambition is to develop a tool *Morphonary* (or *Tvaroslovník* in Slovak) which will provide and administrate all forms of a given word. We have implemented only the most important part of functionality – the lemmatization, i.e. the looking for the lemma of a given word.

2 Lemmatization

Our process of the lemmatization of a given word form can be expressed by the following pseudo-code. X is an input word form, the algorithms returns the lemma (or at least its approximation). Moreover we assume that we have:

- the list of all lemmas (mentioned in the previous section),
- a list of some words associated with their lemmas (made manually by declining some words), we will call them *guides* in the following.

Now the pseudo-code:

- 1 check the presence of X in the list of lemmas;
if “yes” then return X itself and stop
- 2 check the presence of X in the list of guides;
if “yes” then return the lemma of guide X and stop
- 3 repeat the following:

- 3.1 look for a guide Y with the same ending as the ending of X (as long as it can be)
- 3.2 derive the (probably) lemma X'' of X by comparing with the (known) lemma of Y (details beneath)
- 3.3 check the presence of X'' in the list of lemmas; if “yes” then return X'' and stop

The process of the derivation of the probably lemma in the point 3.2, i.e. our exchanging algorithm can be illustrate on the following example. Let us assume that our task is to find the lemma of a word form “ponúk” and this word form is not present in the list of guides. In contrary, the word “ruka” is present there including all its forms.

- 3.2.1 the given word form: $X = \text{“ponúk”}$
- 3.2.2 the guide (found in the point 3.1): $Y = \text{“rúk”}$
- 3.2.3 the (longest) common ending: $E = \text{“úk”}$
- 3.2.4 the beginning of the guide: $Y' = \text{“r”}$ (hence $Y = Y' + E$)
- 3.2.5 the beginning of the given word form: $X' = \text{“pon”}$ (hence $X = X' + E$)
- 3.2.6 the lemma of the guide: $Y'' = \text{“ruka”}$
- 3.2.7 the ending of the lemma of the guide: $E' = \text{“uka”}$ (hence $Y'' = Y' + E'$)
- 3.2.8 the (probably) lemma of the given word form: $X'' = X' + E' = \text{“ponuka”}$

Of course, the word “oblúk” can be a guide for “ponúk” too (if it is in the list of guides). But it leads to the probably lemma “ponúk” (because the lemma of “oblúk” is again “oblúk”) and the check in the point 3.3 fails.

Emphasize that the notion of guide is not identical with the notion classical “school” patterns of the Slovak declination like “chlap”, “hrdina”, “dub”, “stroj”, and so on. Each Slovak (declinable) word can be a guide if it was put by someone to the list. The only important thing is its ending.

Moreover, note that we do not respect the morphological structure of word (in our example. $E = \text{“úk”}$ is not suffix from the grammatical point of view). In fact, this structure is not substantial in the declination process.

3 A few notes on implementation

As times go by, Morphonary has three different implementation (all in Java):

- 1 The first one, rather naive, had the lists of words and of guides in the text files with simple structure, moreover it considers nouns only. It was very slow, but usable for a single word. This version was successfully integrated with a tool OnTeA ([1]) in our project NAZOU.
- 2 The data in the next version was stored in the database (namely MySQL). The process of lemmatization of given word is rather speed, it costs about quarter of one second per word (in average). This version worked with all word types, not only with nouns. It was successfully integrated with a tool JDBSearch ([2]) in our project NAZOU (although it needed some modifications).

3 In the actual version both lists are implemented as Java class `TreeSet` based on the special `ReverseWordComparator` (recall that the common word endings are needed). Thus these two lists can be serialized and de-serialized, hence and we can work with them directly in memory (they have together less than 10KB). It means that work with them is very quick, the finding of lemma of one word lasts about 1 millisecond. This version is actually integrated with no tool, but its ambition is to replace the both previous versions.

The successfulness of searching of lemmas is rather satisfying: the comparing with the “manual” lemmatization repeatedly shows that it is over 90% (experiments were made on ordinary journal articles with hundreds of words). This number can be getting better by replenishing guides in the guide list. But it should be said that the goal 100% is impossible because of the substantial ambiguity of natural language.

There is another principal problem with the idea of storing all forms of all Slovak words: e.g. proper names (especially people surnames), rules of which are very free (if there are at all), In such case we must be satisfied by qualified estimation – the most probably lemma found in the point 3.2 (although the check in 3.3 fails). The same is true for all neologisms (its form and declination usually respect rules valid for similar registered word) or till unknown foreign words (but they are usually indeclinable).

4 Conclusion

In this paper we try to illustrate a simple but effective method for searching the lemma of a given Slovak word form. As we note before, this is only a part (although substantial) of planned functionality. In the full version we want to know (or at least derive) all declined forms of a given word. These future features will be good starting point for the main goal – (at least semi-)automatic syntactic analysis of Slovak sentences, but this problem is far away from the Morphony functionality.

References

1. Laclavík M., Šeleng M., Babik M.: OnTeA: Semi-automatic Ontology based Text Annotation Method, proceedings of Tools for Acquisition, Organization and Presenting of Information and Knowledge, P. Návrat, P. Bartoš, M. Bieliková, L. Hluchý, P. Vojtáš eds., pp. 49–63
2. Lencses R.: Fulltext Indexing and Querying with a Support of Relational Database, proceedings of Tools for Acquisition, Organization and Presenting of Information and Knowledge, P. Návrat, P. Bartoš, M. Bieliková, L. Hluchý, P. Vojtáš eds., pp. 105–110
3. Páleš E.: Sapfo, parafrázovač slovenčiny, Veda, Bratislava, 1994. ISBN 80-224-0109-9.

4. Porter M. F.: An algorithm for suffix-stripping, *Program*, 14 (3), 1980, pp. 130–137.
5. kol.: *Slovník slovenského jazyka*, Vydavateľstvo SAV, Bratislava, 1959–1968.
6. kol.: *Slovník súčasného slovenského jazyka, A–G*, Veda, Vydavateľstvo SAV Bratislava, 2006.
7. Šaling S., Ivanová-Šalingová M., Maníková Z.: *Veľký slovník slovenského jazyka*, SAMO, 2003.