Fast Morphological Analysis of Czech

Pavel Šmerk

Faculty of Informatics, Masaryk University
Botanická 68a, CZ-60200 Brno, Czech Republic
smerk@mail.muni.cz

Abstract. This paper presents a new Czech morphological analyser which takes an advantage of Jan Daciuk's algorithms for minimal deterministic acyclic finite state automata. The new analyser is six times faster than the current analyser `ajka` concerning the proper analysis, i.e. returning possible lemmata and tags for a given word form, but for some other related tasks is the difference even bigger.

Key words: morphological analysis; Czech language

1 Introduction

In the last year our current Czech morphological analyser `ajka` [4,5] have started to be used in two large projects. One of its new "users" is Seznam.cz[1], the first and biggest Czech internet portal and web search engine. The other one is Masaryk University Information System[2] which services not only needs of the second largest Czech university, but also a nation-wide registries of graduate theses[3] and of colloquial and other smaller college works[4] which allow full-text search and detection of plagiarism.

It has turned out that `ajka` is too slow to satisfy the new performance requirements. We have developed a new morphological analyser `majka` which uses the same language data as `ajka`, but the analyser itself as well as the format of the data are completely new. Both are described in Section 2 and in Section 3 some performance comparison with `ajka` is shown.

2 Data

The new morphological analyser `majka` is an implementation of the approach proposed in [6]. The data are simply a list of all combinations of a recognized input and corresponding outputs of the analyser, where pairs of two words are encoded as pairs formed by the first word and a difference between the words. For example, in the following part of data for word form → lemma + tag analysis

[1] `http://www.seznam.cz/` [2] `http://is.muni.cz/?lang=en` [3] `http://theses.cz/?lang=en`
[4] `http://odevzdej.cz`, the website is only in Czech

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2009, pp. 13–16, 2009. © Masaryk University, Brno 2009

```
klouček:A,k1gMnSc1
kloučka:Cek,k1gMnSc2
kloučka:Cek,k1gMnSc4
```

the colon is a delimiter between the possible inputs and corresponding outputs and the letters A and C as the first and the third letters of the alphabet mean "to get the lemma delete n-1 (i.e. 0 or 2, respectively) last characters from the word form and then attach the rest of the string (i.e. empty string or *ek*, respectively)". Then the word form *klouček* will be analyzed as a lemma *klouček* with a morphological tag *k1gMnSc1*[5] and a word form *kloučka* as a lemma *klouček* with morphological tags *k1gMnSc2* and *k1gMnSc4*[6].

Such a list is then represented as a minimal deterministic acyclic finite state automaton using Jan Daciuk's algorithms for incremental building of minimal DAFSAs [1]. This representation dramatically reduces the size of the data (some particular figures can be seen later in Table 2). The lookup is then very simple: if the analysed string concatenated with the delimiter is found in the automaton, then each possible remaining path to a final state of the automaton encodes one of possible analyses.

It means that there is no "real" analysis as a sophisticated algorithm above some grammar model or a system of paradigms, but whole analysis is only a simple — and therefore fast — dictionary search.

3 Performance Comparison

Results of a performance comparison of the analysers `ajka` and `majka` are presented in the Table 1. The comparison was done on the first one million words from the SYN2000 corpus [7] which is a part of the Czech National Corpus[7].

Table 1. Results of comparison of the old analyser `ajka` and the new analyser `majka`

	size of data in MB		time in seconds		
	`ajka`	`majka`	`ajka`	`majka`	ratio
morphological analysis	3.1	4.4	18.22	2.88	**6.3**×
lemmatisation		4.0	16.76	1.57	**10.7**×
all word forms		6.1	55.33	8.42	**6.6**×
restoration of diacritics		3.3	8698.80	1.61	**5403**×

The measured time is a "wall clock" time as was reported by a unix command `time`. All times are averages of three runs. Outputs of the analysers were allways redirected to `/dev/null` to measure only a CPU time and not waits for a hard disk etc.

[5] *little boy* in nominative form [6] *little boy* in genitive and accusative form

[7] `http://www.korpus.cz/english/`

It should be noted that, in the contrary of what a reader might expect, the extreme difference in the speed of the diacritics restoration is the least surprising for us, because this task is implemented very poorly in `ajka`.

The outputs of the old and new analyser are not quite identical: there are still some minor differences or even bugs on either side (mainly in an analysis of compound words), which will be addressed in the near future, but none of these differences can notably affect the overall performance.

As is obvious from the previous section, the new analyser `majka` has to have separate dictionary for each task — unlike `ajka`, which has only one common data, and even smaller. It is a kind of a tax for the speedup, but this several megabytes difference is not a problem for present-day computers.

The Table 2 shows the extent of the compression of dictionaries. For other languages, Kowaltowski [3] reports 0.25 byte per one word-tag-lemma entry for Brazilian Portuguese and Daciuk [1] reports less then 0.15 byte per one word-lemma-tag entry for German and ca. 30 times better compression rate compared to gzip on that data. The presented results show similar or better compression for the Czech data as well

Table 2. Statistical information on the data files of the new analyser `majka`

type of dictionary	# of entries	size of entries	size of dict.	bytes/entry
word (diacritics restor.)	13,609,590	186,154,068	3,263,374	0.240
word → lemma	14,101,767	239,578,702	4,042,839	0.287
word → lemma, tag	80,303,929	2,477,786,062	4,353,616	0.054
word → all word forms	957,464,060	19,993,465,213	6,105,429	0.006

In both tables, there are only tasks, which can be directly handled by `ajka`. Besides, we have also dictionaries for generation all word forms from lemma (which is a "subset" of word → all word forms) and for tasks lemma (or any word form) → word forms + tags, and lemma (or any word form) + tag → word forms, but these tasks are not supported by `ajka`.

4 Conclusions and Future Work

According to Gelbukh and Sidorov [2], the designer of a morphological analyzer for an inflective language has the following choice:

- either generate all word forms and build a system with a large dictionary and a very simple 'analysis" (just searching) algorithm,
- or build a system with a much smaller dictionary of stems with information about possible endings, but with some more sophisticated algorithm (analysis through generation, in particular).

For the inflective languages they strongly suggest the second option, because it allows to use a grammar model almost directly taken over from traditional

grammars, which are oriented mainly toward generation. Our results clearly show that they are wrong. The first approach is better even for the inflective languages, because the analyser remains simple and therefore the analysis runs fast (but of course the simplicity of any program is a value in itself concerning a long term maintenance and development). Apparently, using our approach the designer of a morphological analyser is absolutely free regarding the choice of a suitable grammar model, because there are no constraints on how and from what sources one can generate the dictionary data.

The presented results are only preliminary as the new analyser `majka` is still under an intensive development. We expect that the final data files will be smaller: at least in some cases, as e.g. the generation of all word forms, we are aware of particular inefficiencies regarding the format of data. We believe that there is also a potential of some performance improvements, so that the final version will be even faster.

Acknowledgements This work has been partly supported by the Academy of Sciences of the Czech Republic under the project 407/07/0679 and by the Ministry of Education, Youth and Sports within the National Research Programme II project 2C06009 and project LC536.

References

1. Jan Daciuk. 1998. *Incremental Construction of Finite-State Automata and Transducers, and their Use in the Natural Language Processing.* Ph.D. dissertation, Technical University of Gdańsk, Poland.
2. Alexander Gelbukh and Grigori Sidorov. 2003. *Approach to Construction of Automatic Morphological Analysis Systems for Inflective Languages with Little Effort.* In: Computational Linguistics and Intelligent Text Processing. Proc. CICLing-2003. LNCS 2588, Springer-Verlag, pp. 215–220.
3. Tomasz Kowaltowski, Cláudio L. Lucchesi, and Jorge Stolfi. 1998. *Finite Automata and Efficient Lexicon Implementation.* Technical Report IC-98-02, University of Campinas, São Paulo.
4. Radek Sedláček and Pavel Smrž. 2001. *A New Czech Morphological Analyser* `ajka`. In Proceedings of the 4th International Conference TSD 2001. LNCS 2166, Springer-Verlag, pp. 100–107.
5. Radek Sedláček. 2004. *Morphematic analyser for Czech.* Ph.D. dissertation, Faculty of Informatics, Masaryk University, Brno.
6. Pavel Šmerk. 2007. *Morphemic analysis: A Dictionary Lookup Instead of Real Analysis.* Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2007. Masaryk University, Brno. pp. 77–85.
7. Czech National Corpus – SYN2000. 2000. Institute of the Czech National Corpus, Praha. Accessible at: `http://www.korpus.cz`.