Intelligent Search and Replace for Czech Phrases

Zuzana Nevěřilová, Vít Suchomel

Natural Language Processing Centre
Faculty of Informatics
Masaryk University

5 December 2014

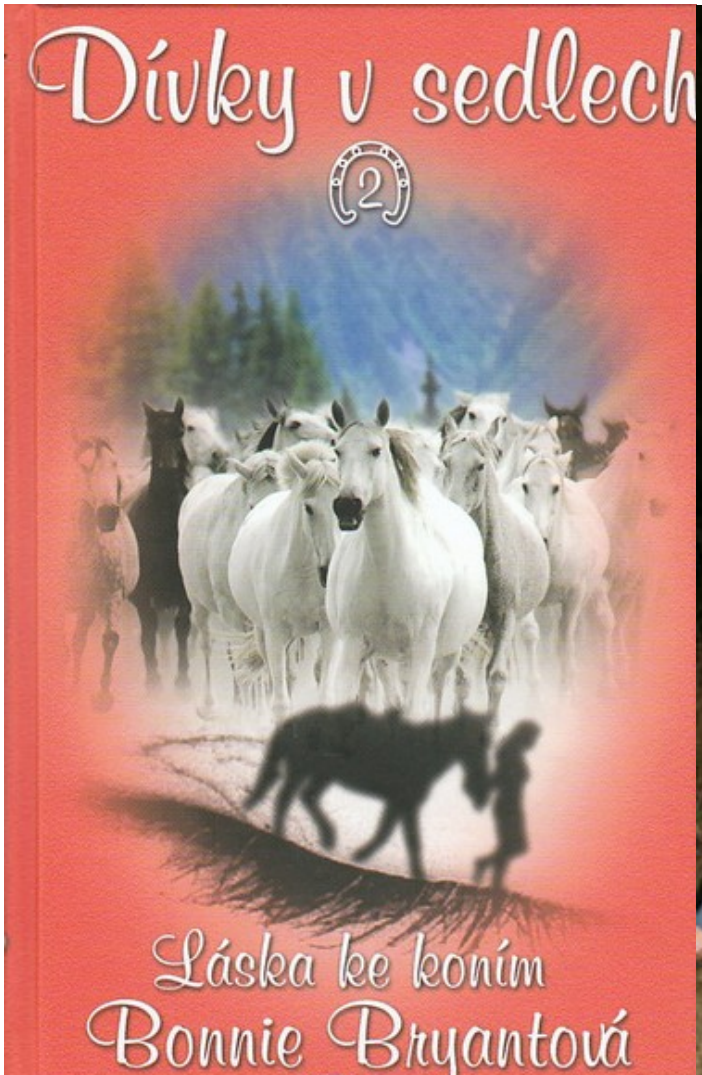
Partially supported by the Czech-Norwegian Research Programme within the HaBiT Project 7F14047.

The idea

Are you the chief editor in a translation project merging different translations of the same term?

The idea

Or have you read "Girls in saddles" by Bonnie Bryant and thought they originally drank *beer* rather than *lemonade*?



The idea

Or has your sex been suddenly changed?
Karel Novák $\Longrightarrow$ *Karla Nováková*

More intelligence for the search and replace function

The standard search and replace

- a part of all text processors,
- works with strings,
- replaces exact occurrences of the search string by the replacement string,
- no "text understanding" (no morphology, the Scunthorpe problem).

The more intelligent functionality should

- handle morphology
- deal with advanced grammar, e.g. grammatical agreement,
- accept words as well as multiword phrases,
- be reliable (ask the user rather than making a mistake).

Use cases

Intelligent search and replace can help to save editing time:

- correction of often repeated mistakes,
- unification of terms in translation (done by the chief translating editor),
- especially unification of terms in localisation (e.g. GUI descriptions: ‘dialog box’ – ‘dialogové okno’ (neuter gender), ‘dialogový box’ (masculine inanimate),
- adjusting general parts of manuals to particular products,
- changing ingredients in recipes,
- replacing person or company names in standardised documents,
- especially in legal text, e.g. common parts of contracts.

Initial conditions

1. identify noun phrases (`SET [?]`)
2. determine phrase lemma $\neq$ lemmata
   "tato dvě červená jablka" (these two red apples) $\neq$ "tento" (this), "dva" (two), "červený" (red), "jablko" (apple)
3. determine other sentence parts with grammatical agreement (verb phrase in past tense, predicative comp
   - head modifiers
   - active verb
   - predicative complement

The Algorithm: replace $p \rightarrow r$ in text T

1. parse whole T: detect phrase lemmata $p_i(lemma)$
2. look if $p(lemma) \in p_i(lemma)$ for some i
3. replace such $p_i(lemma)$ with $r(lemma)$
4. change modifiers if needed
5. change active verb participle if p_i is the subject
6. change predicative complement if p_i is the subject and the clause has copula verb

Examples: dům → budova (house → building)

- vysoký dům → *vysoká* budova
  (tall house → tall building)
- vysoký dům stál na nabřeží → *vysoká* budova *stála* na nábřeží
  (the tall house was on the waterfront → the tall building was on the waterfront)
- dům byl velmi *vysoký* → budova *byla* velmi *vysoká*
  (the house was very tall → the building was very tall)

Pitfalls

- anaphors: dům byl vysoký a neměl hromosvod (the house was tall and *it* did not have the lightning conductor)
- idioms: jít o dům dál (to continue (mainly after a failure))
- synonyms: barák (house)


Kovář, V., Horák, A., and Jakubíček, M. (2011).
Syntactic analysis using finite patterns: A new parsing system for Czech.
In *Human Language Technology. Challenges for Computer Science and Linguistics*, volume November 6-8, 2009, pages 161–171, Poznań, Poland.