LibreOffice
The Document Foundation

LibreOffice Language Technology

László Németh

nemeth@numbertext.org

FSF.hu Foundation

TIRANA | 27 September 2018
[extended slides 11/14/18]

LIBOCON
TIRANA 2018

LIBREOFFICE LANGUAGE TECHNOLOGY

News & Best practices

LIBOCON
TIRANA 2018

Agenda

- Numbertext
- NatNum12
- “Grammar By” spell checking
- Hunspell 1.7
- Best practices

libnumbertext

C++ port of Soros lang. and Numbertext

☞http://numbertext.org

NUMBERTEXT.ORG

Home of Soros Programming Language

New features & new languages: **Albanian**, Bulgarian, Croatian, Galician, Icelandic, Malaysian, Norwegian, Swiss Standard German & Ukrainian

Numbertext integration

Outline numbering with

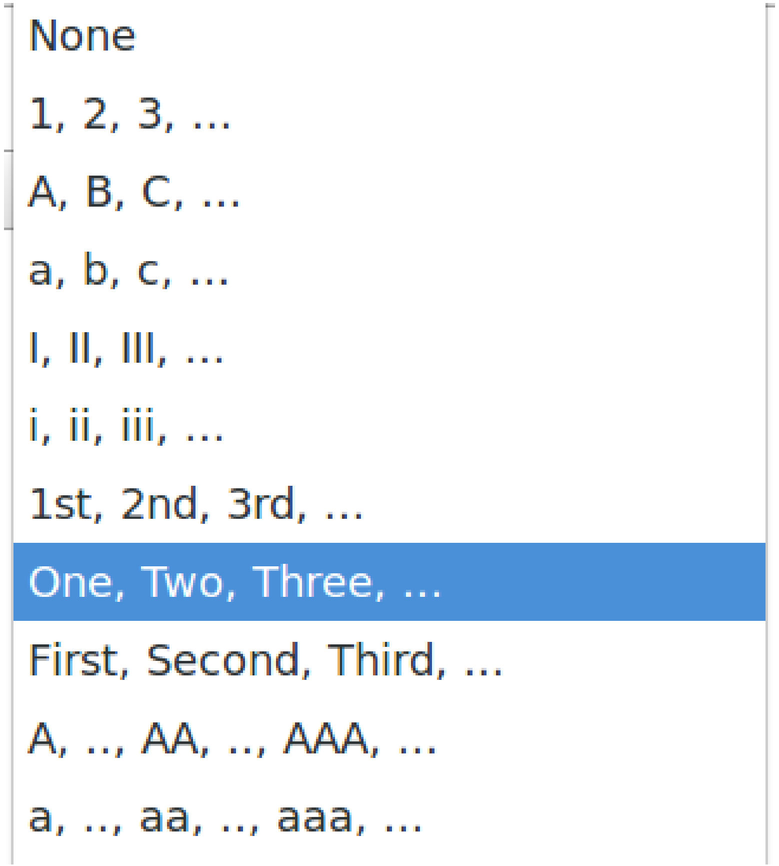
– ordinal indicators

– cardinal names

– ordinal names

DOCX import/export

Screencast: ☞Usage



None

1, 2, 3, ...

A, B, C, ...

a, b, c, ...

I, II, III, ...

i, ii, iii, ...

1st, 2nd, 3rd, ...

One, Two, Three, ...

First, Second, Third, ...

A, .., AA, .., AAA, ...

a, .., aa, .., aaa, ...

NatNum12 modifier

Numbertext numbering functions

– money, year, gender, affixation…

New NatNum12 for number format codes

[NatNum12 USD]0

[NatNum12 cardinal-feminine]0

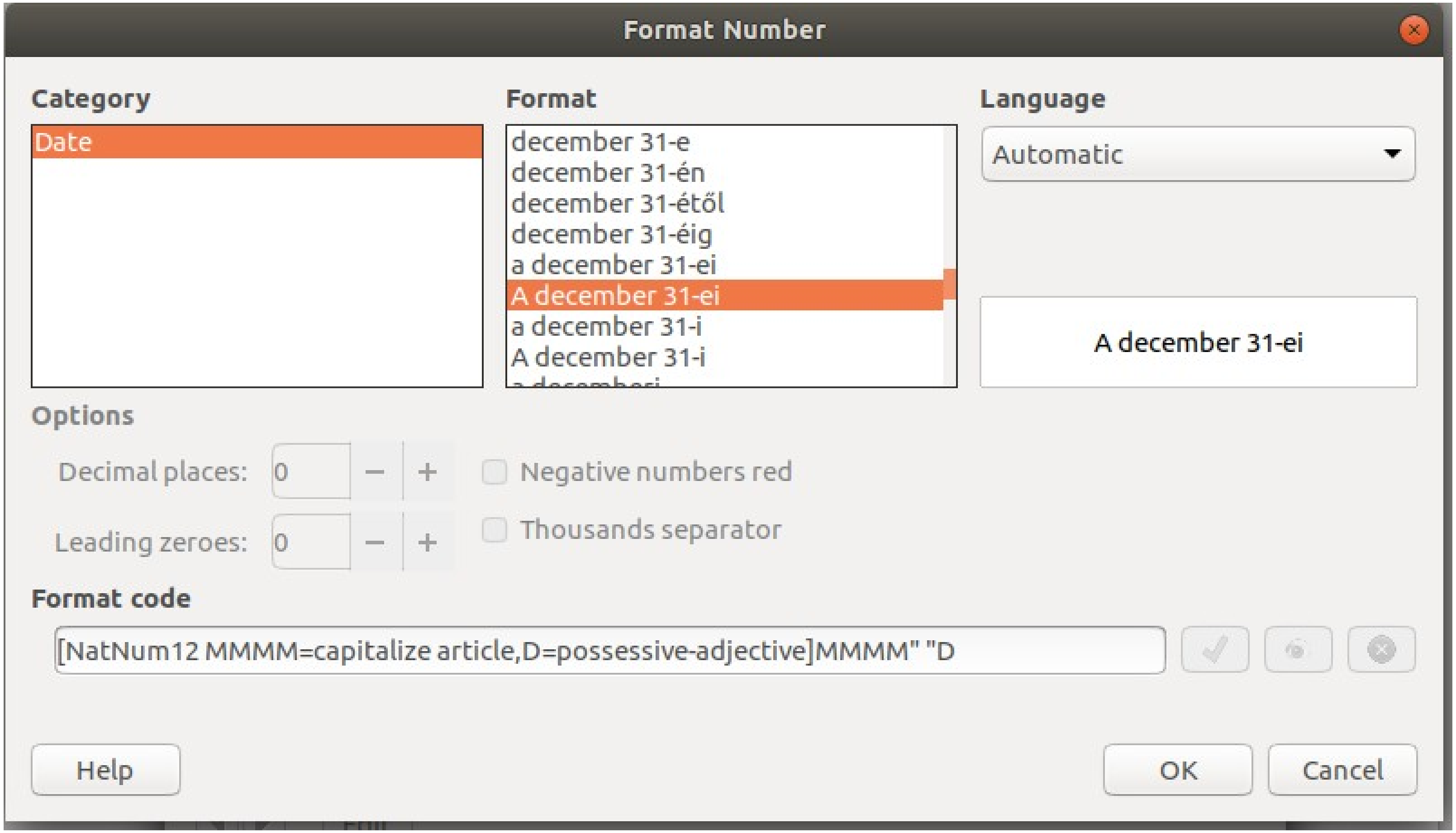
[NatNum12 capitalize year]0

☞LibreOffice Help

NatNum12 in dates

Support formats of affix-rich languages

☞in Hungarian (code: ☞LO, numbertext)



Category: Date

Format:
- december 31-e
- december 31-én
- december 31-étől
- december 31-éig
- a december 31-ei
- A december 31-ei
- a december 31-i
- A december 31-i
- a decemberi

Language: Automatic

A december 31-ei

Options

Decimal places: 0

Leading zeroes: 0

□ Negative numbers red

□ Thousands separator

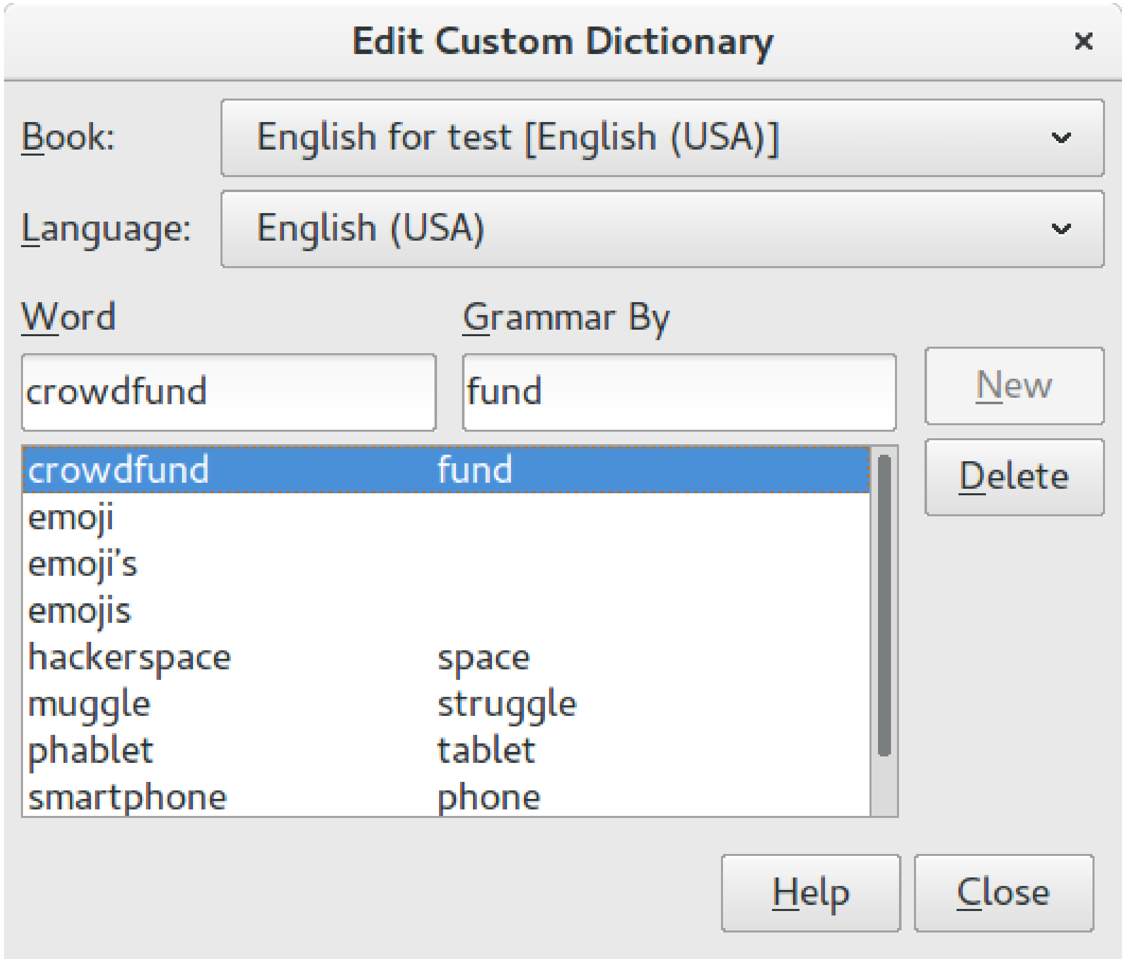
Format code: [NatNum12 MMMM=capitalize article,D=possessive-adjective]MMMM" "D

“Grammar By”...

More extensible spell checking

Affixation & compounding of user dictionary words: #113739

by an arbitrary model word



Edit Custom Dictionary

Book: English for test [English (USA)]

Language: English (USA)

Word	Grammar By
crowdfund	fund
crowdfund	fund
emoji	
emoji's	
emojis	
hackerspace	space
muggle	struggle
phablet	tablet
smartphone	phone

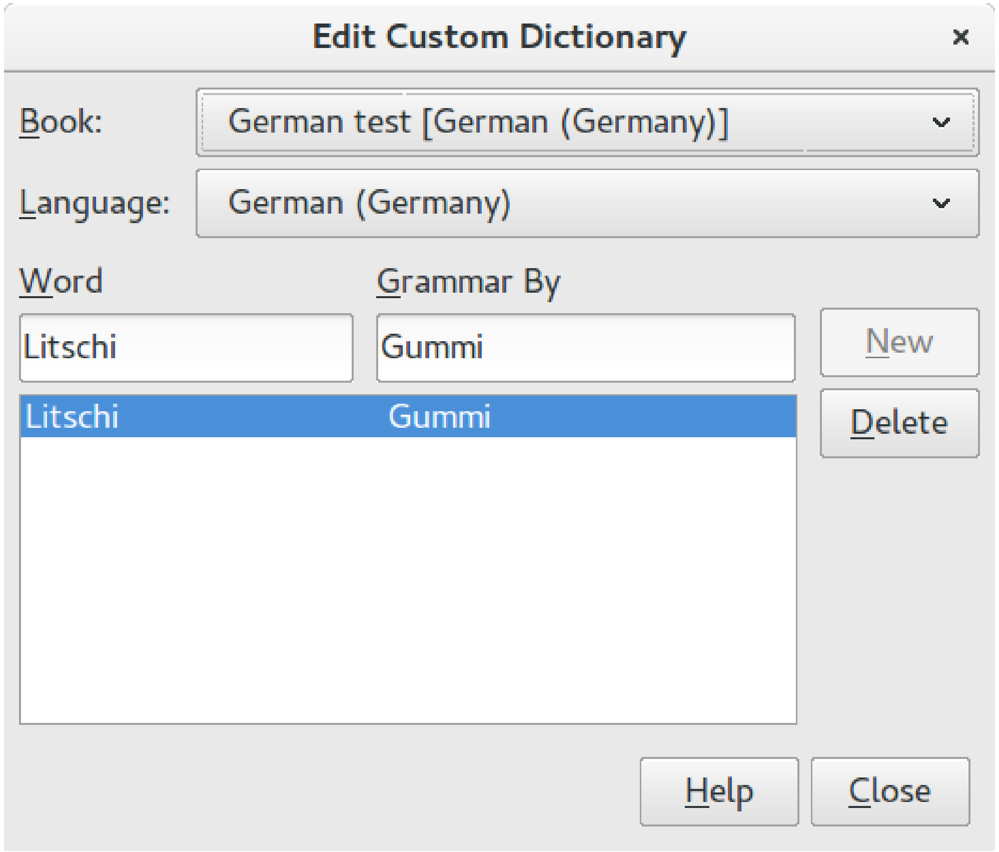
For custom dictionaries

☞LibO 6.0 release notes

Screencasts:
☞English,
☞German

Works with every Hunspell dictionaries with affixation or compounding

Using model words, custom dictionaries with affixation or compounding don’t depend from the changes of the main spelling dictionary



Edit Custom Dictionary

Book: German test [German (Germany)]

Language: German (Germany)

Word	Grammar By
Litschi	Gummi

Extra dictionaries

E.g. extras/source/wordbook/hu_Akh11.dic (old Hungarian orthography)

Shipped with LO (only with hu-HU localization) ☞patch

New feature of the user dictionary format: field **title**

“Grammar by” syntax: *new_word==model*

```
OOoUserDict1
lang: hu-HU
type: positive
title: Régi helyesírás (AkH. 11.)
---
Adriennel
Adriennek
bedekker==bédekker
chili==csili
chips==caries
...
```

LIBOCON
TIRANA 2018

Special dictionaries

For checking medical, legal, scientific etc. documents

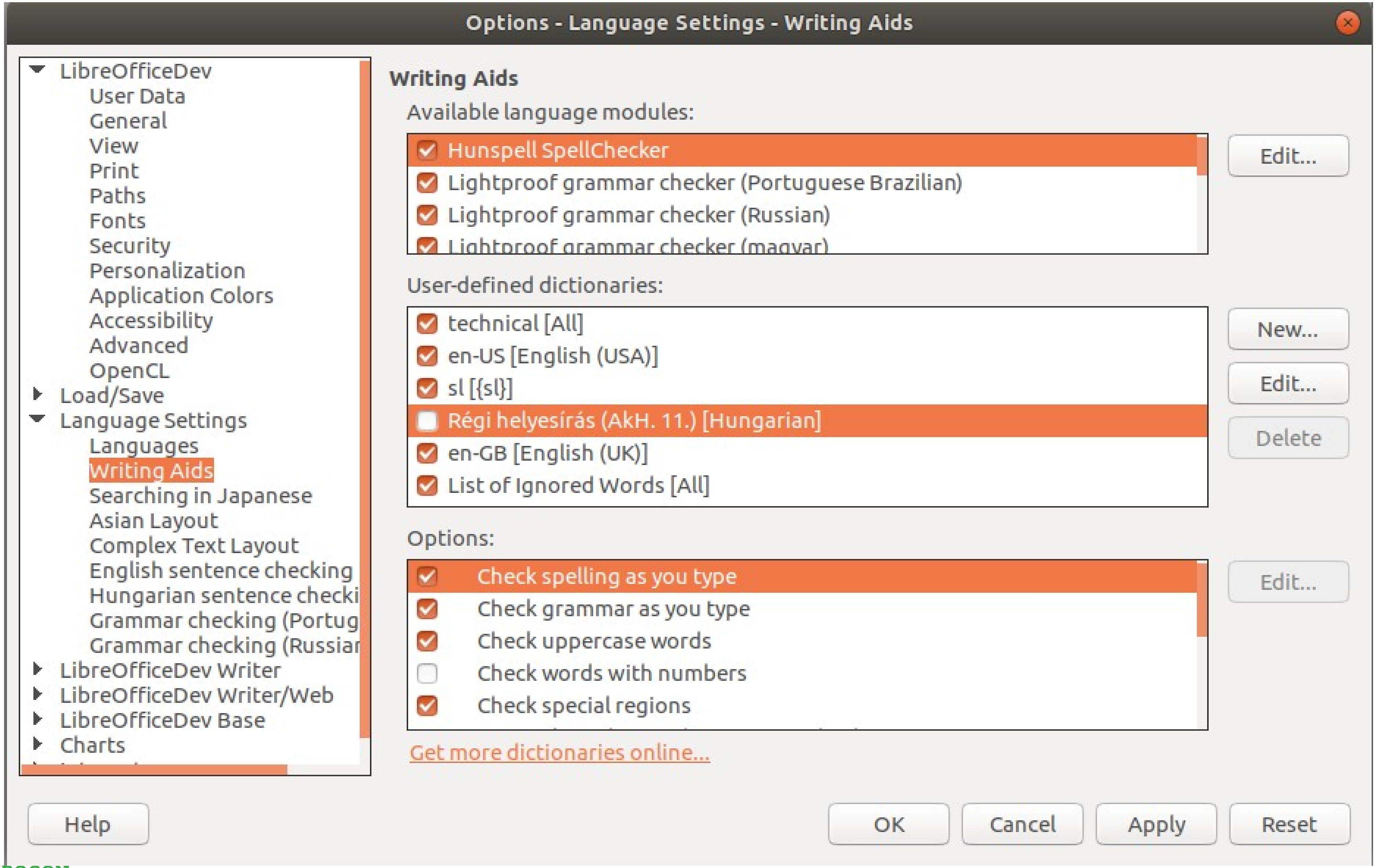
They shouldn’t be default dictionaries:

– Their orthography differ from the common

– Suggesting special words can annoy users

Proposed solution: ship nondefault custom dictionaries with LibreOffice, reusing the existing interface for switchable dictionaries

Switchable by users



Writing Aids

Available language modules:

- ☑ Hunspell SpellChecker
- ☑ Lightproof grammar checker (Portuguese Brazilian)
- ☑ Lightproof grammar checker (Russian)
- ☑ Lightproof grammar checker (magyar)

User-defined dictionaries:

- ☑ technical [All]
- ☑ en-US [English (USA)]
- ☑ sl [{sl}]
- ☐ Régi helyesírás (AkH. 11.) [Hungarian]
- ☑ en-GB [English (UK)]
- ☑ List of Ignored Words [All]

Options:

- ☑ Check spelling as you type
- ☑ Check grammar as you type
- ☑ Check uppercase words
- ☐ Check words with numbers
- ☑ Check special regions

Hunspell spell checker

Hunspell 1.7

Suggestion: no freezing & slowing down

Limit overgeneration of suggestions

Word pair support as in real dictionaries

Custom suggestions

Improved handling of OpenDocument

Other fixes for several languages

Release with test files and manual: ☞v1.7.0

No freezing

Guaranteed suggestion time (<0.5 sec) also with languages with compounding or complex morphology

Balanced multi-level time limits

Fix also possible combinatorial explosions by overlapping words in the recursive compound word segmentation algorithm

Check with long misspellings of Dutch, German, Hungarian, Turkish etc.

Without limitation

Faster suggestions allowed **removing artificial limitation** for affix-rich languages

Hunspell can fix words with maximal affix count (prefixes and both derivative and inflectional suffix groups) again:

le|fikszál|ás|ára → *le|fixál|ás|ára*

[right time to check ☞doubleaffixcompress]

Less strangeness

No (often strange) n-gram and compound word suggestions, if “good” suggestions exist, i.e. uppercase, REP, ph: or dictionary word pair suggestions

Don’t suggest capitalized dictionary words for lower case misspellings in **n-gram** suggestions (except PHONE usage; or in the case of German, where not only proper nouns are capitalized; or the capitalized word has special pronunciation)

And don’t suggest if the difference of lengths of misspellings and suggestions is 5 or more characters.

And less...

A dictionary word pair has got top priority in suggestions for writing the word pair without spaces (in the case of NOSPLITSUGS affix file option, too), and this is the only suggestion, if there is no other “good” suggestion (more on the next slide).

Also dictionary word pairs separated by dash instead of space get priority in two-word suggestion (in languages with Latin letters or with dash TRY character):

scotfree → only *scot-free* (no *scot free*, *scooter*, etc.)

A lot easier “a lot”…

List the frequently misspelled two-word expressions simply in the .dic file – as in a traditional spelling dictionary – to get the single correct suggestion, for example *alot* → *a lot*:

```
------------ example.dic -------------

...

a lot
```

Custom suggestions

Dic file field “ph:” used not only in low-priority n-gram, but in **high priority custom suggestions**, too, like *etcetra* → *and so on, et cetera*; or *andsoon* → *and soon, and so on*:

------------ in example.dic -------------

et cetera ph:etcetra

and soon ph:andsoon

and so on ph:andsoon ph:etcetra

ph: within words

> ph: definitions handled exactly as “REP” definitions now, i.e. they are recognized within **affixed forms and compounds**. For example, we get *jists* → *gists* suggestion at the first place, not only *jist* → *gist* with the following single definition:
>
> *------------ in example.dic ------------*
>
> *gist ph:jist*

ph: more syntax

Asterisk at the end of ph: pattern results shortening of the replacement pattern. In this example, we define the shortened *hep* → *happ* replacement rule, resulting the *hepier* → *happier* correction:

------------ in example.dic ------------

*happy ph:hepy**

ph: and more…

Narrowing the previous rule, or in more complex cases, we can add a REP rule in the ph: definition, using the ASCII arrow pattern “->”. Here the *hepi* → *happi* rule among the default *hepy* → *happy*:

------------ in example.dic -------------

happy ph:hepy ph:hepi->happi

Syllable duplication

Better correction of syllable duplication

Not only ABABA → ABA (for example nutrITITIon → nutrITIon), but the simpler ABAB → AB pattern is recognized (in non-starting position), for example,

regretTETEd -> regretTEd

(helping affix-rich languages especially)

Fix BREAK

It’s possible to forbid compound forms recognized by BREAK word breaking rule by adding the bad compounds to the dictionary with FORBIDDENWORD flags, e.g. to reject the bad Hungarian word form *“n-t”*:

n-t → *n-et* (accusative of the Hungarian letter *n)*

Recognize words in BREAK compounds when they already contain word break characters, for example, “e-mail” is a dictionary word with the word break character n-dash, and it wasn’t accepted before in BREAK compounds in some cases and languages:

e-mail-küldés → breaking as *e-mail|küldés*, too, not only the old *e|mail-küldés*, to recognize the solely correct *“e-mail”.*

Fix special casing

Allow dotted İ in dictionary, and disable bad capitalization of i. (Dictionary words weren’t recognized with dotted İ, but dictionary words with the letter i were recognized with dotted İ, too.)

~~İzmir~~, İtem → İzmir, ~~İtem~~

Extend dotless ı and dotted İ rules to Crimean Tatar language to support its special casing of ı/I, i/İ (as in Azeri and Turkish).

Command line

Remove huge amounts of false alarms during filtering ODF documents by ignoring <text:span> elements (added by text modifications frequently in LibreOffice). This fixed the fast custom dictionary creation and usage:

```
$ hunspell -l corrected_doc.odt >my_new_words.txt
$ hunspell -p my_new_words.txt -l new_doc.odt
```

List filenames during filtering multiple files in command-line:

```
$ hunspell -l *.odt
a.odt: misspelling
b.odt: egzample
```

Best practices

LIBOCON TIRANA 2018

Best practices

Collect typical misspellings (typos, phonetic replacements, obsolete forms) and use TRY, MAP, REP, SET UTF-8 and **improved ph:**

Test the coverage of your dictionary on word frequency lists and Wikipedia

See Hunspell manual, test examples and existing dictionaries ☞v1.7.0, ☞dictionaries

Extend extras/ (AutoCorrect) ☞example

“Small is beautiful”

Minor changes with big benefits

Default smart apostrophe (#38395)

L10n AutoCorrect rules, e.g. in Greek:

– σ (sigma) to ς (final sigma) (#116387)

Keyword replacement in AutoCorrect:

– :sigma: → σ, cm:^2: → cm^2, :hand: → 🖐

Example: Hungarian

Special word breaking(s)

Cross-references with a/az articles

Field recognition in grammar checking

Hunspell extensions for Hungarian

Pronunciation based input improvements

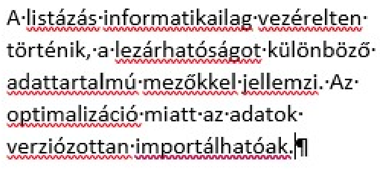
Predefined date formats with affixation

Improved word and sentence sorting

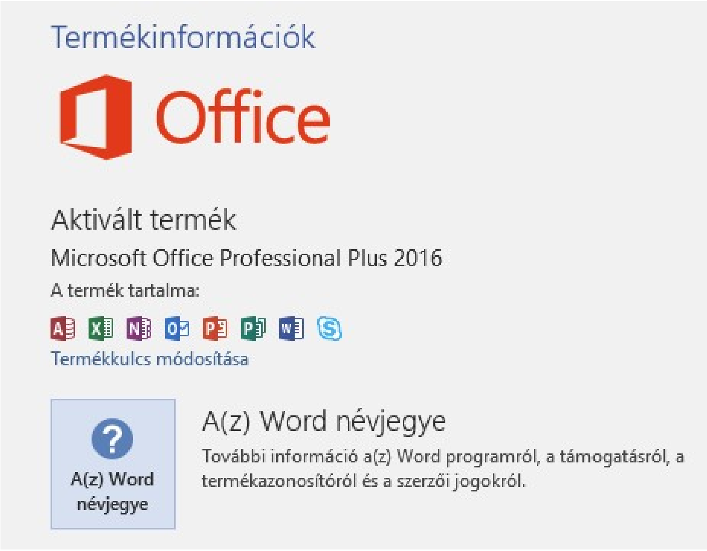
Better spell checking

Better coverage: words, Unicode, morphology *(example and screenshots by Balázs Úr)*

Better suggestions: newest feature: lots of REP, ph: phonetic transcriptions = **new input method for foreign words**, e.g. *ofsór → offshore*



A·listázás·informatikailag·vezérelten·történik,·a·lezárhatóságot·különböző·adattartalmú·mezőkkel·jellemzi.·Az·optimalizáció·miatt·az·adatok·verziózottan·importálhatóak.¶



Termékinformációk

Office

Aktivált termék

Microsoft Office Professional Plus 2016

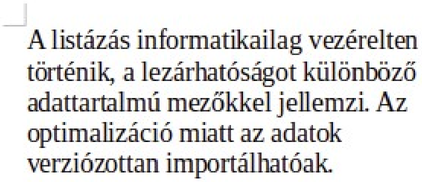
A termék tartalma:

Termékkulcs módosítása

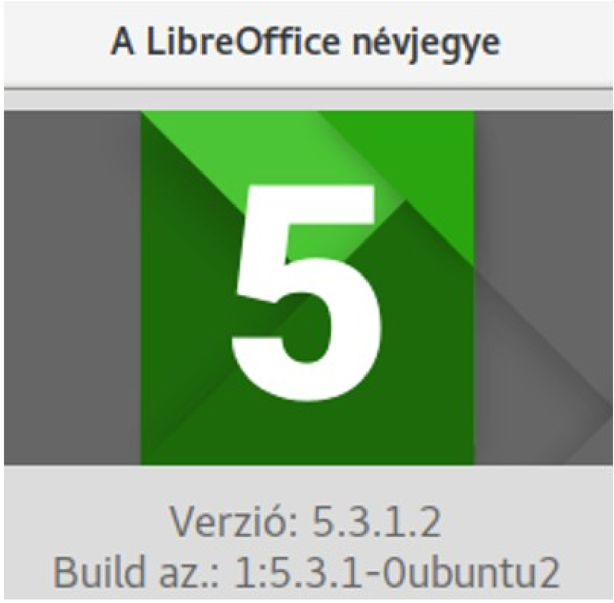
A(z) Word névjegye

A(z) Word névjegye

További információ a(z) Word programról, a támogatásról, a termékazonosítóról és a szerzői jogokról.



A listázás informatikailag vezérelten történik, a lezárhatóságot különböző adattartalmú mezőkkel jellemzi. Az optimalizáció miatt az adatok verziózottan importálhatóak.



A LibreOffice névjegye

Verzió: 5.3.1.2

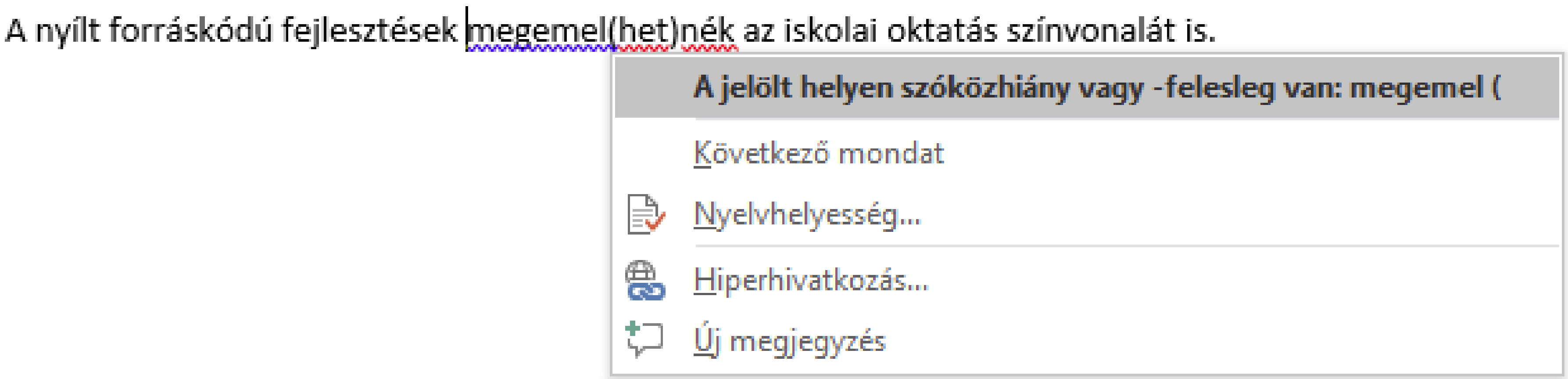
Build az.: 1:5.3.1-0ubuntu2

Special word break

Extended word breaking rules & IGNORE affix file parameter (#116072, ☞patch)

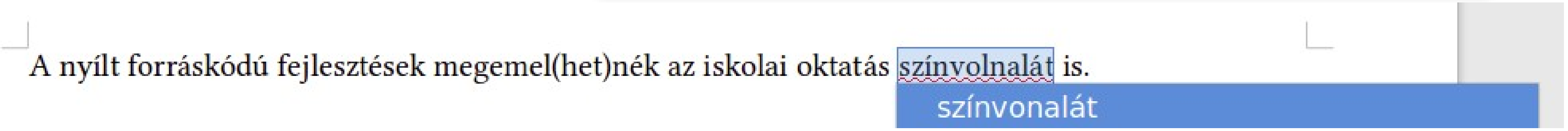
Example word: *megemel(het)nék*

3 false alarms in Word, zero in Writer:



A nyílt forráskódú fejlesztések megemel(het)nék az iskolai oktatás színvonalát is.

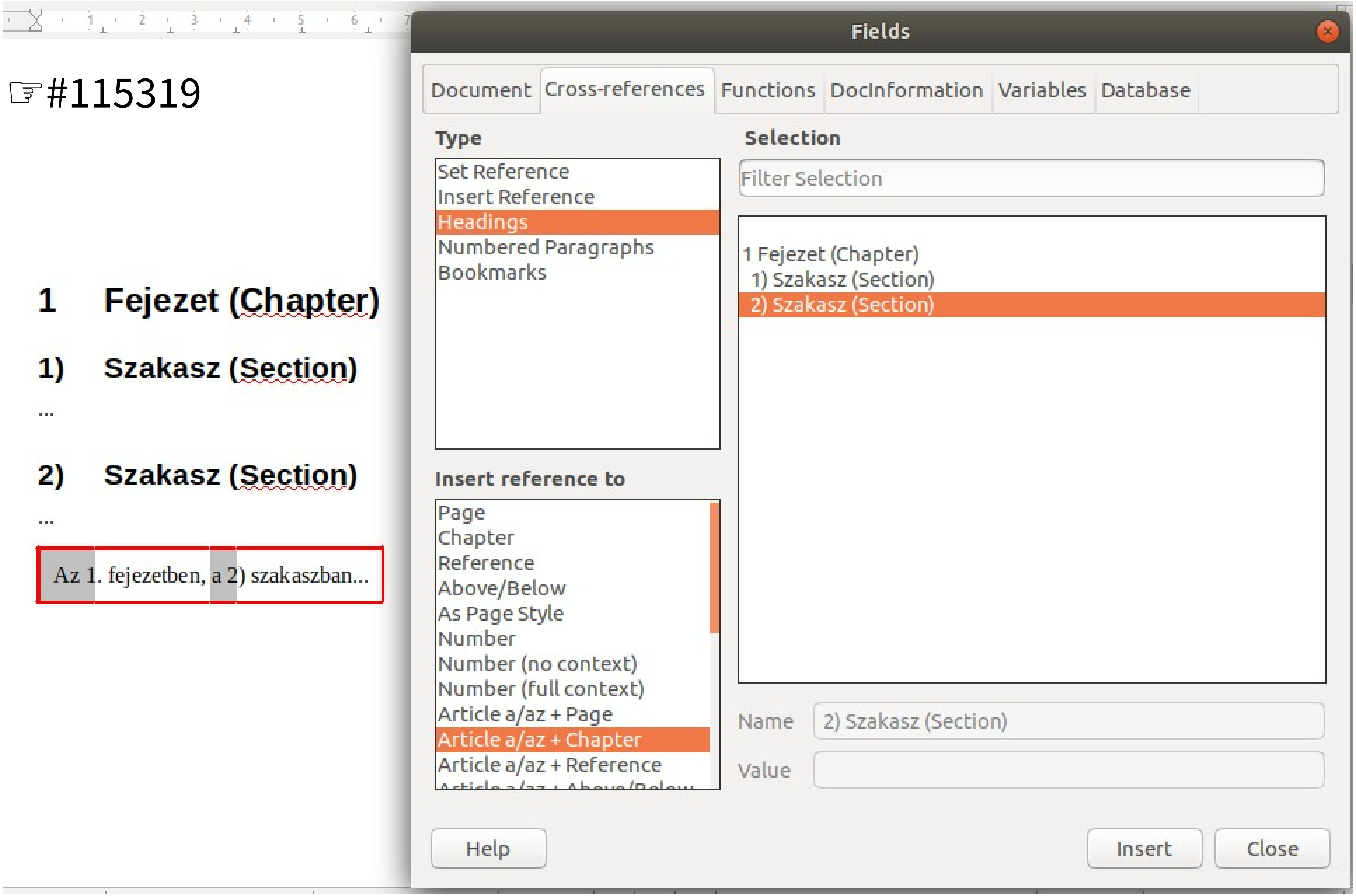
A jelölt helyen szóközhiány vagy -felesleg van: megemel (



A nyílt forráskódú fejlesztések megemel(het)nék az iskolai oktatás színvolnalát is.

Article a/az selection

☞#115319



1 Fejezet (Chapter)

1) Szakasz (Section)

...

2) Szakasz (Section)

...

Az 1. fejezetben, a 2) szakaszban...

Name 2) Szakasz (Section)

Value

Check field content

Improved fix for ☞#69416: cross-reference field content is not removed any more for spell checking (only footnote numbering), also a ZWSP character added to it to recognize field content in grammar checkers to avoid false alarms

☞Lightproof patch for Hungarian

To check consistency of actual page/chapter/etc. numberings in cross-references with their articles and affixes (a typical problem of several languages):

Az 4. oldalon → A 4. oldalon, A 5. oldalon → Az 5. oldalon,

a)-ben → a)-ban, b)-ban → b)-ben

SUMMARY

LibreOffice has got excellent language technology features, it’s worth to know and use them for your language!

Acknowledgements

Co-developers of NatNum12:

Mike Kaganski, Eike Rathke

Numbertext contributors:

http://numbertext.github.io/#team

Primary sponsor of my developments:

FSF.hu Foundation, Hungary