A modern replacement for spell(1)

Abhinav Upadhyay <abhinav@NetBSD.org>

AsiaBSDCon 2017

Problems with old spell

Problems with old spell

- Very ancient - dates back to Unix version 7

Problems with old spell

- Very ancient - dates back to Unix version 7
- Uses inflection rules to do spell check

Problems with old spell

- Very ancient - dates back to Unix version 7
- Uses inflection rules to do spell check
- Rules not 100% accurate, for example mis-spellings like *appled*, *coffeed*, *undoubt*, *repremanded, undoubtedlys* are not caught

Problems with old spell

- Very ancient - dates back to Unix version 7
- Uses inflection rules to do spell check
- Rules not 100% accurate, for example mis-spellings like *appled*, *coffeed*, *undoubt*, *repremanded, undoubtedlys* are not caught
- Rules only work for English language

Problems with old spell

- Very ancient - dates back to Unix version 7
- Uses inflection rules to do spell check
- Rules not 100% accurate, for example mis-spellings like *appled*, *coffeed*, *undoubt*, *repremanded, undoubtedlys* are not caught
- Rules only work for English language
- No support for spell corrections

Problems with old spell

- Very ancient - dates back to Unix version 7
- Uses inflection rules to do spell check
- Rules not 100% accurate, for example mis-spellings like *appled*, *coffeed*, *undoubt*, *repremanded, undoubtedlys* are not caught
- Rules only work for English language
- No support for spell corrections
- No library interface for other applications to add spell check support - shells, pkgin, pkg_add, apropos could benefit

A modern spell(1)

A modern spell(1)

- Uses a expanded dictionary instead of inflection rules (size of new dictionary 5.1M compared to 2.4 M of the old dictionary)

A modern spell(1)

- Uses a expanded dictionary instead of inflection rules
- Levenshtein distance and soundex techniques used for finding possible corrections

A modern spell(1)

- Uses a expanded dictionary instead of inflection rules
- Levenshtein distance and soundex techniques used for finding possible corrections
- Also support for n-gram models to do context sensitive corrections and grammar checks (experimental WIP)

A modern spell(1)

- Uses a expanded dictionary instead of inflection rules
- Levenshtein distance and soundex techniques used for finding possible corrections
- Also support for n-gram models to do context sensitive corrections and grammar checks (experimental WIP)
- A tool to parse any corpus and generate dictionary to do application specific spell check

A modern spell(1)

- Uses a expanded dictionary instead of inflection rules
- Levenshtein distance and soundex techniques used for finding possible corrections
- Also support for n-gram models to do context sensitive corrections and grammar checks (experimental WIP)
- A tool to parse any corpus and generate dictionary to do localized or application specific spell check
- The core spell checking and correction functionality available as a reusable library

How spell correction works

- Levenshtein distance - minimum number of edits required to convert one string into another
- Generate all possible words at distance 1 or 2 and see which ones of them are in the dictionary
- Lower weight to corrections involving a change in the 1st character or replacement of a character given
- Higher weight to corrections having the same soundex code
- On no match at distance 1, same process done at distance 2
- If still no match, word having the same soundex code with minimum edit distance selected.

Support for other languages

- Levenshtein distance is language agnostic
- Dictionary for any language can be generated and used for spell checking
- But before that some work needed to add support for wide chars

Comparison with GNU aspell

- Total number of tests: 3945
- Matches at first place: 91.33% (aspell 74%)
- Matches at positions 1-5: 95.26% (aspell 96.6%)
- Matches at positions 1-10: 95.59% (aspell 98.2%)
- Matches at positions 1-25: 95.77% (aspell 99%)
- Matches at positions 1-50: 95.84% (aspell 99.2%)
- Matches at 1-100: 95.92% (aspell 99.2%)

Questions?

Thank you :-)