Using the sylly Package for Hyphenation and Syllable Count

m.eik michalke

October 2, 2017

1 Hyphenation

The method `hyphen()` takes vectors of character strings (i.e., single words) and applies an hyphenation algorithm (Liang, 1983) to each word. This algorithm was originally developed for automatic word hyphenation in LaTeX, and is gracefully misused here to fulfill a slightly different service.[1]

`hyphen()` needs a set of hyphenation patterns for each language it should analyze. If you're lucky, there's already a pre-built package[2] for your language of interest that you only need to install and load. These packages are called `sylly.XX`, where `XX` is a two letter abbreviation for the particular language. For instance, `sylly.de` adds support for German, whereas `sylly.en` adds support for English:

```
> sampleText <- c("This", "is", "a", "rather", "stupid", "demonstration")
> library(sylly.en)
> hyph.txt.en <- hyphen(sampleText, hyph.pattern="en")
```

1.1 Alternative output formats

The method has a parameter called `as` which defines the object class of the returned results. It defaults to the S4 class `kRp.hyphen`. In addition to the hyphenated tokens, it includes various statistics and metadata, like the language of the text. These objects were designed to integrate seamlessly with the methods and functions of the `koRpus` package.

[1] The `hyphen()` method was originally implemented as part of the `koRpus` package, but was later split off into its own package, which is `sylly`. `koRpus` adds further `hyphen()` methods so they can be used on tokenized and POS tagged objects directly.

[2] Officially supported packages can be found in the `l10n` repository: `https://undocumeantit.github.io/repos`

When all you need is the actual data frame with hyphenated text, you could call `hyphenText()` on the `kRp.hyphen` object. But you could also set `as="data.frame"` accordinly in the first place. Alternatively, using te shortcut method `hyphen_df()` instead of `hyphen()` will also return a simple data frame.

If even you're only interested in the numeric results, you can set `as="numeric"` (or `hyphen_c()`), which will strip down the results to just the numeric vector of syllables.

2 Support new languages

Should there be no package for your language, you can import pattern files from the LaTeX sources[3] and use the result as `hyph.pattern`:[4]

```
> url.is.pattern <- url("http://tug.ctan.org/tex-archive/language/hyph-
utf8/tex/generic/hyph-utf8/patterns/txt/hyph-is.pat.txt")
> hyph.is <- read.hyph.pat(url.is.pattern, lang="is")
> close(url.is.pattern)
> hyph.txt.is <- hyphen(icelandicSampleText, hyph.pattern=hyph.is)
```

3 Correcting errors

`hyphen()` might not produce perfect results. As a rule of thumb, if in doubt it seems to behave rather conservative, that is, is might underestimate the real number of syllables in a text.

Depending on your use case, the more accurate the end results should be, the less you should rely on automatic hyphenation alone. But it sure is a good starting point, for there is a function called `correct.hyph()` to help you clean these results of errors later on. The most comfortable way to do this is to call `hyphenText(hyph.txt.en)`, which will get you a data frame with two colums, `word` (the hyphenated words) and `syll` (the number of syllables), in a spread sheet editor:[5]

```
>  hyphenText(hyph.txt.en)

   syll     word
[...]
20    1    first
21    1    place
22    1  primary
```

[3] Look for `*.pat.txt` files at `http://tug.ctan.org/tex-archive/language/hyph-utf8/tex/generic/hyph-utf8/patterns/txt/`

[4] You can also use the private method `sylly:::sylly_langpack()` to generate an R package skeleton for this language, but it requires you to look at the `sylly` source code, as the commented code is the only documentation. The results of this method are optimized to be packaged with `roxyPackage` (`https://github.com/unDocUMeantIt/roxyPackage`). In this combination, generating new language packages can almost be automatized.

[5] For example, this can be comfortably done with RKWard: `http://rkward.kde.org`

```
23    2 de-fense
24    1      and
[...]
```

You can then manually correct wrong hyphenations by removing or inserting "-" as hyphenation indicators, and call the function without further arguments, which will cause it to recount all syllables:

```
> hyph.txt.en <- correct.hyph(hyph.txt.en)
```

Of course the function can also be used to alter entries on its own:

```
> hyph.txt.en <- correct.hyph(hyph.txt.en, word="primary",
+ hyphen="pri-ma-ry")

Changed

   syll    word
22    1 primary

  into

   syll      word
22    3 pri-ma-ry
```

References

Liang, F. M. (1983). *Word Hy-phen-a-tion by Com-put-er* (Unpublished doctoral dissertation). Stanford University, Dept. Computer Science, Stanford.