Using the koRpus Package for Corpus Analysis

m.eik michalke

July 8, 2015

The R package `tm.plugin.koRpus` is an extension to the `koRpus` package, enhancing its usability for actual corpus analysis. It adds new classes and methods to `koRpus`, which are designed to work with complete text corpora in both `koRpus` and `tm` formats. This vignette gives you a quick overview.

1 What is tm.plugin.koRpus?

While the `koRpus` package focusses mostly on analysis steps of individual texts, `tm.plugin.koRpus` adds several new object classes and respective methods, which can be used to analyse complete text corpora in a single step. These classes are also a first step to combine object classes of both, the `koRpus` and `tm` packages.

There are three basic classes, which are hierarchically nested:

- class `kRp.topicCorpus` holds a list (named by topics) of objects of
  - class `kRp.sourcesCorpus`, which in its `sources` slot holds a list of objects of
    - class `kRp.Corpus`, which in turn contains objects of both `koRpus` and `tm` classes.

The idea behind this is to be able to categorize corpora on at least two levels. The default assumes that these levels are different *sources* and different *topics*, but apart from this naming (which is coded into the classes) you can actually use this for whatever levels you like.

If you don't need these levels, you can just use the function `simpleCorpus()` to create objects of class `kRp.Corpus`. It represents a flat corpus of texts. To distinguish texts which came from different sources, use the function `sourcesCorpus()`, which will generate sub-corpora for each source given. And one level higher up, use the function `topicCorpus()`, to sort `kRp.sourcesCorpus` objects by different topics. Objects of this class will only be valid if there are texts of each topic from each source.

2 Tokenizing corpora

As with `koRpus`, the first step for text analysis is tokenizing and possibly POS tagging. This step is performed by the functions mentioned above, `simpleCorpus()`, `sourcesCorpus()`, or `topicCorpus()`, respectively. The package includes four sample texts taken from Wikipedia[1] in its `tests` directory we can use for an elaborate demonstration:

```
> library(tm.plugin.koRpus)
> # set tha root path to the sample files
> sampleRoot <- file.path(path.package("tm.plugin.koRpus"), "tests", "testthat", "samples")
> # now we can define the topics (names of the vector elements)
> # and their main path
> samplePaths <- c(
>   C3S=file.path(sampleRoot, "C3S"),
>   GEMA=file.path(sampleRoot, "GEMA")
> )
> # we also define the sources
> sampleSources <- c(
>   wpa="Wikipedia_alt",
>   wpn="Wikipedia_neu"
> )
> # and finally, we can tokenize all texts
> sampleTexts <- topicCorpus(paths=samplePaths, sources=sampleSources, tagger="tokenize", lan

processing topic "C3S", source "Wikipedia_alt", 1 texts...
processing topic "C3S", source "Wikipedia_neu", 1 texts...
processing topic "GEMA", source "Wikipedia_alt", 1 texts...
processing topic "GEMA", source "Wikipedia_neu", 1 texts...
```

3 Analysing corpora

After this, we can analyse the corpus by calling the provided methods, for instance lexical diversity:

```
> sampleTexts <- lex.div(sampleTexts, char=FALSE, quiet=TRUE)
> corpusSummary(sampleTexts)
```

[1] see the file `tests/testthat/samples/License_of_sample_texts.txt` for details