The stoRy package: a general overview

Paul Sheridan and Mikael Onsjö

Last updated: February 25, 2018

The `stoRy` package implements 1) the hypergeometric test for over-representation of literary themes in a storyset (a list of stories) relative to a background list of stories, and 2) a recommendation system that takes a user-selected story as input and returns a ranked list of similar stories on the basis of shared themes. The package is currently implemented for the episodes of the Star Trek television franchhise series The Original Series (TOS), The Animated Series (TAS), The Next Generation (TNG), and Voyager (VOY).

1 Installation

The package is hosted on CRAN and can be installed by running the command

```
install.packages("stoRy")
```

Once installed, the package can be loaded by running the standard `library` command

```
library("stoRy")
```

2 Accessing documentation

Each function in the package is documented. Run the command

```
help(package="stoRy")
```

to see a brief overview of the package functions. The package vignette can be accessed by running

```
vignette(package="stoRy")
```

Function help files can be accessed using the usual R command. For example, help on the method `get_enriched_themes` can be obtained with

```
?get_enriched_themes
```

3 Example: View and edit a story's themes and metadata

The `stoRy` package contains themed Star Trek episodes with additional metadata for The Original Series (TOS), The Animated Series (TAS), The Next Generaton (TNG), and Voyager (VOY). A `story` object is created in order to view the themes (and metadata) for a particular episode of interest. For example, to view the data for the classic TOS episode "Arena" (story ID tos1x19), initialize a `story` object as follows

```
story_id <- "tos1x19"
mystory <- story$new(story_id)
print(mystory)
```

Annotated themes are stored in the data frame object

```
mystory$themes
```

Each theme is categorized as being either *central* or *peripheral* to the story; comments along with related characters, aliens, and things are also recorded. Annotated story settings and keywords can be accessed in a similar manner

```
mystory$settings
mystory$keywords
```

Themes can be added and removed as desired. For example, run the following command to add the theme "neo-luddist utopia" as a central theme

```
mystory$add_theme(theme = "neo-luddist utopia", level = "central")
```

The theme can be removed as follows

```
mystory$remove_theme(theme = "neo-luddist utopia")
```

Settings and keywords can be added and removed in a similar manner

```
mystory$add_setting(setting = "mountain")
mystory$remove_setting(setting = "mountain")
mystory$add_keyword(keyword = "Captain Kirk is climbing a mountain")
mystory$remove_keyword(keyword = "Captain Kirk is climbing a mountain")
```

There are 278 themed episodes in total. The raw story IDs can be viewed by loading `sysdata` and running the command

```
load("R/sysdata.rda")
sysdata$RESERVED_STORY_IDS
```

The following command sequence shows the first ten story IDs along with associated metadata

```
load("R/sysdata.rda")
head(sysdata$story_metadata)
```

4 Example: Exploring the themes

It is possible to examine the individual themes contained in the `stoRy` package collection. For example, to examine the theme "utopia" run the following command sequence

```
theme_name <- "utopia"
mytheme <- theme$new(theme_name)
mytheme$print()
```

The output includes the theme definition, sometimes an illustrative example, and its place in a theme hierarchy. In total, there are 1535 different themes that are arranged into four hierarchies: the human condition, society, the pursuit of knowledge, and alternate reality. The theme "utopia" falls withing the society hierarchy. To view "utpoia" and its descendents in a tree format run the command

```
print_tree(mytheme)
```

The themes from each of the four hierarchies can be viewed in the same manner. For example, for an overview of themes in the society hierarchy run the following command sequence

```
theme_name <- "society"
mytheme <- theme$new(theme_name)
print_tree(mytheme, pruneMethod = "dist", limit = 50)
```

Setting `pruneMethod` in this manner ensures that the upper levels in the hierarcy are displayed; the value of `limit` determines how many themes will be displayed.

5 Example: Enriched themes in Klingon episodes

This section is devoted to an example usage of the hypergeometric test for theme over-representation analysis. In the analysis, the hypergoemetric test is applied to identify over-represented themes in Klingon-centric episodes relative to the backgrounds TOS/TAS and TNG, respectively. First, read in the `aliens.smt` storysets file

```
file <- system.file("storysets", "aliens.smt", package = "stoRy")
mystorysets <- storysets$new(file)
print(mystorysets)
```

The `get_enriched_themes` function performs the hypergeometric test to check whether each of the 2129 individual themes is over-represented in a test sotryset relative to the background storyset. The `get_enriched_themes` function applies the test separately to each storyset in the `mystorysets` object relative to a background of all 446 Star Trek television series episodes by default. To test the Klingon storyset against a background of all TOS and TAS series episodes run the command

```
results <- get_enriched_themes(mystorysets,
               test_storysets = "KLINGON",
               background_storyset = c("tos", "tas"))
```

The top twenty enriched themes can be viewed as follows

```
results$KLINGON[1:20,]
```

In the data frame, n is the size of the test storyset, k is the number of stories in the test storyset featuring the associated theme, N is the size of the background storyset, and K is the number of stories in the background storyset featuring the associated theme. The P-value is calculated using the hypergeometric test.

Run the following command sequence to find over-represented themes in Klingon-centric episodes when the TNG series episodes are used as a background storyset

```
results <- get_enriched_themes(mystorysets,
               test_storysets = "KLINGON",
               background_storyset = "tng")
results$KLINGON[1:20,]
```

6 Example: Enriched themes in the Star Trek TV series

Here we demonstrate another example usage of the `get_enriched_themes` hypergeometric test for theme over-representation analysis. This time the hypergoemetric test is applied to identify over-represented themes in each series TOS, TAS, TNG, VOY relative to the background TOS/TAS/TNG/VOY. First, read in the `series.smt` storysets file

```
file <- system.file("storysets", "series.smt", package = "stoRy")
mystorysets <- storysets$new(file)
print(mystorysets)
```

The `get_enriched_themes` function performs the calculation for each series relative to the background TOS/TAS/TNG/VOY by default

```
results <- get_enriched_themes(mystorysets)
```

The results can be accessed as follows

```
results$TOS[1:20,]
results$TAS[1:10,]
results$TNG[1:20,]
results$VOY[1:20,]
```

7 Example: Finding episodes most similar to a selected episode

The function `get_similar_stories` can be used to find episodes similar to a user-selected episode. Take finding episodes similar to the Voyager episode "False Profits" (`voy3x05`) as an example. First, create a story object for the story in question

```
story_id <- "voy3x05"
mystory <- story$new(story_id)
```

The following command evaluates episodes in the default background of TOS/TAS/TNG/VOY according to their similarity to `mystory`. The default cosine similarity function can be changed to either the cosine tf-idf or soft cardinality similarity function (see the `get_similar_stories` help file for details).

```
result <- get_similar_stories(mystory)
```

The top 10 most similar episodes can be accessed as follows

```
result[1:10,]
```