Approximate text matching with the stringdist package

Mark van der Loo

Statistics Netherlands
mark.vanderloo@gmail.com

Keywords: approximate string matching

Comparing text strings in terms of distance functions is a common and fundamental task in many statistical text-processing applications. Thus far, string distance functionality has been somewhat scattered around *R* and its extension packages, leaving users with inconsistent interfaces and encoding handling. The newly developed **stringdist** package is designed to offer an easy to use interface to several popular string distance algorithms which have been re-implemented in *C* for this purpose. The package offers distances based on counting *q*-grams, edit-based distances, and some lesser known heuristic distance functions [1].

For example, to compute the true Damerau-Levenshtein distance between two strings, one uses the `stringdist` function as follows.

```
> library(stringdist)
> stringdist('leia', 'leela', method='dl')
```

```
[1] 2
```

The distance of two corresponds to two edit operations necessary to turn `'leia'` into `'leela'`. For example: replace `'i'` with `'e'` and insert an `'l'`.

For approximate dictionary lookup one may use the `amatch` function:

```
> companions <- c('adric', 'ace', 'leia')
> amatch('leela', companions, method='dl', maxDist=2)
```

```
[1] 3
```

Here, `'leela'` matches with the third element of `companions` since it is both the closest match and its DL-distance is less than or equal to `maxDist`.

In this presentation I will review the string distance algorithms offered by the package, show how to apply them and point out some particularities related to special values (`NA`) and character encoding.

References

[1] M.P.J. van der Loo (2014). *The stringdist package for approximate string matching*. Accepted for publication in the R Journal.