An integrated framework in R for textual sentiment time series aggregation and prediction


'sentometrics' repository: https://github.com/sborms/sentometrics.

Text mining…

... is the process of distilling actionable insights from text.

Our focus is on textual sentiment analysis.
Time series econometrics... 

... is the analysis of quantitative time series data typically in an economic context.

Our focus is on aggregation, econometric modelling and prediction.
Let’s go for a run with the R package ‘sentometrics’

```r
library("sentometrics")

data("usnews", package = "sentometrics")
```

We have a built-in dataset of news articles between 1995 and 2014, from The Wall Street Journal and The Washington Post.

```
<table>
<thead>
<tr>
<th>ID</th>
<th>DATE</th>
<th>TEXT</th>
<th>WSJ</th>
<th>WAPO</th>
<th>ECONOMY</th>
<th>NONECONOMY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1995-01-02</td>
<td>Full text 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1995-01-05</td>
<td>Full text 2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```

Features: relevance/importance indicators & selectors.
Massaging the corpus

Checking the requirements of the corpus.

```r
corpusAll <- sento_corpus(usnews)
```

Subsetting the corpus, using the `quanteda` package.

```r
corpus <- quanteda::corpus_subset(corpusAll, date < "2014-10-01")
```

Adding features (for example: entities, topics, events).

```r
regex <- c("\\bRepublic\[s]?\\b\\bDemocrat\[s]?\\b\\belection\\b\\b[US|U.S.] \[p|P]resident\\b\\bwar\\b")
corpus <- add_features(corpus,
    keywords = list(uncert = "uncertainty", uselect = regex),
    do.binary = TRUE,
    do.regex = c(FALSE, TRUE))
```
Pick the word lists for lexicon-based sentiment analysis

We have English, Dutch and French built-in word lists.

```r
data("lexicons", package = "sentomemtrics")
data("valence", package = "sentometrics")
```

Prepare and check the lexicons.

```r
lex <- setup_lexicons(lexiconsIn = lexicons[c("LM_eng", "HENRY_eng")],
                      valenceIn = valence[["valence_eng"]])
```
From sentiment to time series: aggregation specs

Aggregation of the many sentiment scores...
  ... within documents = document-level sentiment
  ... across documents = time series
  ... across time = smoothed time series

... across lexicons, features and time aggregation schemes

One control function to define all of this.

```r
ctrAgg <- ctr_agg(
  howWithin = "tf-idf",
  howDocs = "proportional",
  howTime = c("equal_weight", "linear", "almon"),
  do.ignoreZeros = TRUE,
  by = "month",
  fill = "zero",
  lag = 12,
  ordersAlm = 1:3,
  do.inverseAlm = TRUE)
```
This one simple function call gives you a wide number of different sentiment time series, or “measures”.

```
sentMeas <- sento_measures(corpus, lexicons = lex, ctr = ctrAgg)
```

The sentiment measures are represented as “lexicon—feature—smoothing”.

```
head(sentMeas[["measures"]][, 1:5])
```

<table>
<thead>
<tr>
<th>date</th>
<th>LM_eng--wsj--equal_weight</th>
<th>LM_eng--wapo--equal_weight</th>
<th>LM_eng--economy--equal_weight</th>
<th>LM_eng--noneconomy--equal_weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 1995-12-01</td>
<td>-0.03038392</td>
<td>-0.03096058</td>
<td>-0.02514323</td>
<td>-0.03072403</td>
</tr>
<tr>
<td>2: 1996-01-01</td>
<td>-0.03074413</td>
<td>-0.03262021</td>
<td>-0.02200173</td>
<td>-0.03485245</td>
</tr>
<tr>
<td>3: 1996-02-01</td>
<td>-0.03349817</td>
<td>-0.03567584</td>
<td>-0.02548210</td>
<td>-0.03746940</td>
</tr>
<tr>
<td>4: 1996-03-01</td>
<td>-0.03106851</td>
<td>-0.03681972</td>
<td>-0.02363359</td>
<td>-0.03776122</td>
</tr>
<tr>
<td>5: 1996-04-01</td>
<td>-0.02889475</td>
<td>-0.03420715</td>
<td>-0.02486474</td>
<td>-0.03497349</td>
</tr>
<tr>
<td>6: 1996-05-01</td>
<td>-0.02873871</td>
<td>-0.03299130</td>
<td>-0.02532216</td>
<td>-0.03381545</td>
</tr>
</tbody>
</table>
Plotting across the three time series dimensions

plot(sentMeas, group = "lexicons")

plot(sentMeas, group = "time")

plot(sentMeas, group = "features")
We try to predict the monthly U.S. EPU index...

The Economic Policy Uncertainty (EPU) index is a partly news-based measure of policy-related economic uncertainty. It is served with the package as a dataset.

http://www.policyuncertainty.com
... using elastic net regularization

We propose to use the **elastic net** regression (relying on *glmnet*), which balances between the LASSO and Ridge regressions through an $\alpha$ parameter. The large number and collinearity of the sentiment measures motivate this choice.

\[
y_{u+h} = \delta + \gamma^T x_u + \beta_1 s_{u1} + \ldots + \beta_p s_{up} + \ldots + \beta_p s_{up} + \epsilon_{u+h}
\]

A straightforward control function defines the model setup.

```r
ctrIter <- ctr_model(
  model = "gaussian",
  type = "BIC",
  h = 1,
  alphas = c(0.3, 0.5, 0.7),
  do.iter = TRUE,
  nSample = 36)
```

Steps 4 – 5
Load the data.

data("epu", package = "sentometrics")
y <- epu[epu["date"] >= sentMeas["measures"][["date"]][1], "index"]

Running the out-of-sample prediction analysis is easy.

out <- sento_model(sentMeas, y, ctr = ctrIter)

We call “attribution” the decomposition of the prediction into one of the underlying sentiment time series dimensions.

attr <- retrieve_attribution(out, sentMeas, do.normalize = FALSE)
Visualizing the out-of-sample prediction and attribution

plot(out)

plot_attributions(attr, group = "features")
Next steps

The package already offers quite some flexibility to develop sentiment time series.

Improvements along:
- Faster and more complex sentiment analysis;
- Interfaces to more types of models;
- More flexible aggregation and modelling.

Purpose?
Become the go-to package for embedding textual sentiment into the prediction of other variables!

If you want to help out, get in touch!