

Text clustering with string kernels in R

Alexandros Karatzoglou¹ Ingo Feinerer²

¹Department of Statistics and Probability Theory,
Technische Universität Wien, Austria

²Department of Statistics and Mathematics,
Wirtschaftsuniversität Wien, Austria

30th Annual GfKI-Conference, 2006

Outline

- 1 Introduction
- 2 Statistical Environment
- 3 Theoretical Background
- 4 Experimental Setup
- 5 Results
- 6 Summary

Motivation

- Direct computation in the feature space
- Text mining:
 - Use comprehensive framework and infrastructure in R
 - Use string kernels and kernel methods
- Text clustering:
 - Which methods work best?
 - Benchmark performance and running time

R Infrastructure

- Environment for Statistical Computing
- Highly extensible
 - via packages
 - C linkage for computationally-intensive tasks (like string kernels)
- Effective data handling
- Open Source

kernlab R Package

- Extensible S4 package for kernel-based ML methods in R
- Contains :
 - functions for computing kernel expressions
 - kernel methods for regression and classification (SVM, RVM, Gaussian Processes)
 - clustering (kernel k-means, spectral clustering)
 - ranking and kernel PCA

textmin R Package

- Text mining framework
- Tailored for
 - Plain texts, articles and papers
 - Web documents (XML, SGML, ...)
 - Surveys
- Methods for
 - Clustering
 - Classification
 - Visualisation

Kernel Methods

- Use an implicit mapping $\Phi : X \rightarrow H$ into a high dimensional feature space
- learning takes place in this space and the data only appears inside dot products $\langle \Phi(x), \Phi(x') \rangle$
- kernel function is used to calculate the dot product

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

- Support Vector Machine a popular kernel method

(Kernel) k -means Clustering

- Classical k -means:

$$\mathcal{D}(\{\pi_c\}_{c=1}^k) = \sum_{c=1}^k \sum_{a_i \in \pi_c} |a_i - m_c|^2$$

- Kernel k -means:

$$\mathcal{D}(\{\pi_c\}_{c=1}^k) = \sum_{c=1}^k \sum_{a_i \in \pi_c} |\phi(a_i) - m_c|^2$$

Spectral Clustering

- Embed data points into the subspace of the K eigenvectors of an affinity/kernel matrix
- Cluster embedded points using k -means (Ng et al, Shi and Malik)
- Performance is better because embedded points form tight clusters

String Kernels

- String Kernels:

$$k(x, x') = \sum_{s \sqsubseteq x, s' \sqsubseteq x'} \lambda_s \delta_{s, s'} = \sum_{s \in A} \text{num}_s(x) \text{num}_s(x') \lambda_s$$

- We consider the case where $\lambda_s = 0$ for all $|s| \neq n$ (string or spectrum kernel)
- and where $\lambda_s = 0$ for all $|s| > n$ (full string kernel).

Reuters News Corpus

Reuters-21578 XML

- Set of originally 21578 news articles
- Three topics used:
 - ① acq (shortened): ≈ 1000 documents
 - ② corn: ≈ 240 documents
 - ③ crude: ≈ 580 documents
- Removal of empty documents and white space
- Conversion to lower case
- Final working set of ≈ 1720 documents

Spectral Clustering

- λ values: 0.2, 0.5, 0.8
- String length values: 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14
- String vs. full string kernel
- Spectral clustering with 3 clusters
- 10 runs

Kernel k -means Clustering

- Kernel 3-means (3 topics) clustering
- String length values: 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14
- String vs. full string kernel
- 10 runs

k-means Clustering

- Preprocessing:
 - White space removal
 - Punctuation marks removal
 - Stopword removal
 - Stemming
- Term-document matrix
 - term frequency (tf) weighting
 - term frequency inverse document frequency (tf-idf) weighting
- Classical 3-means (3 topics) clustering
- 10 runs

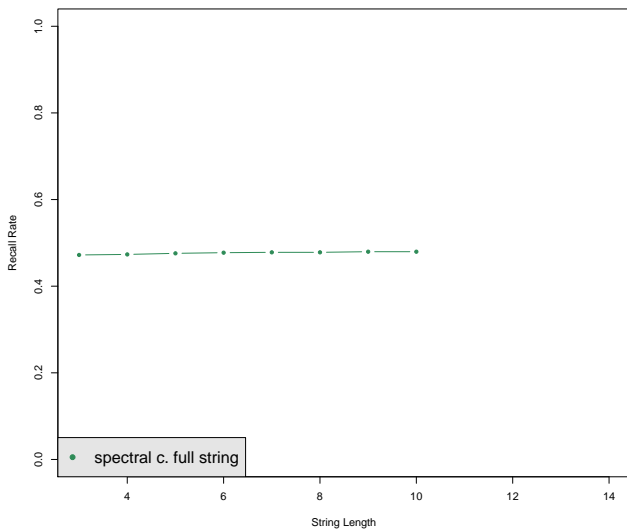
Performance Measures

- Recall rate

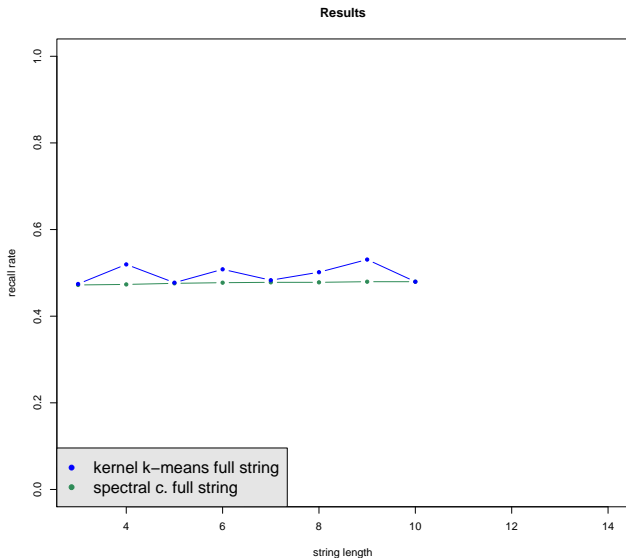
$$R = \frac{\sum_{\Gamma=1}^k n_{\gamma\Gamma}}{\sum_{\Gamma=1}^k N_{\Gamma}}$$

- Internal validity (Silhoutte plot)
- Timings

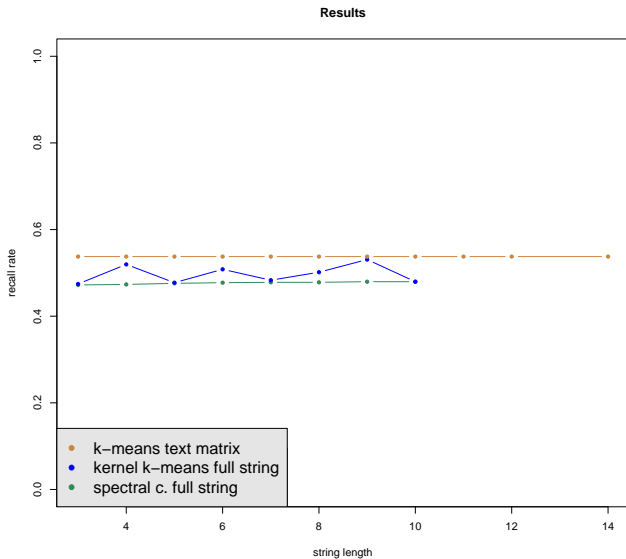
Recall Rate for Reuters-21578 Sample



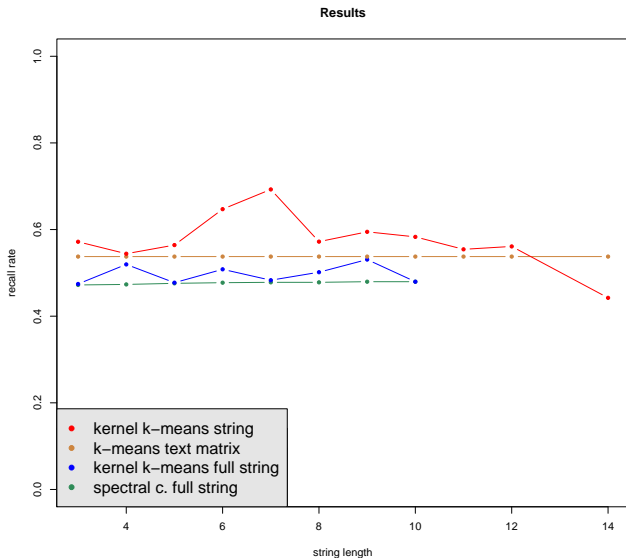
Recall Rate for Reuters-21578 Sample



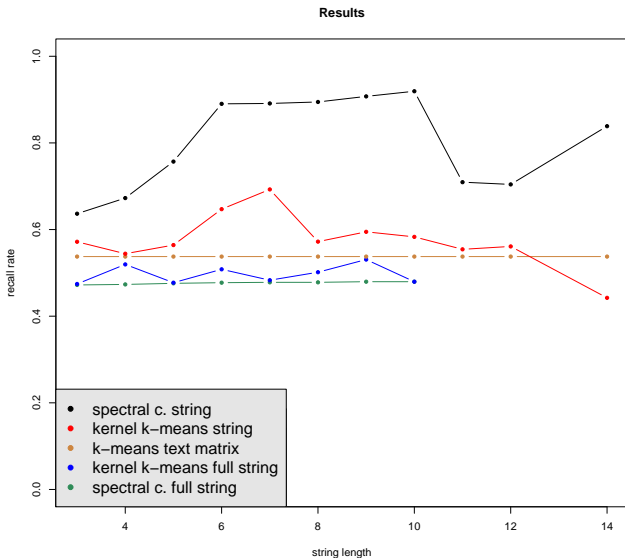
Recall Rate for Reuters-21578 Sample



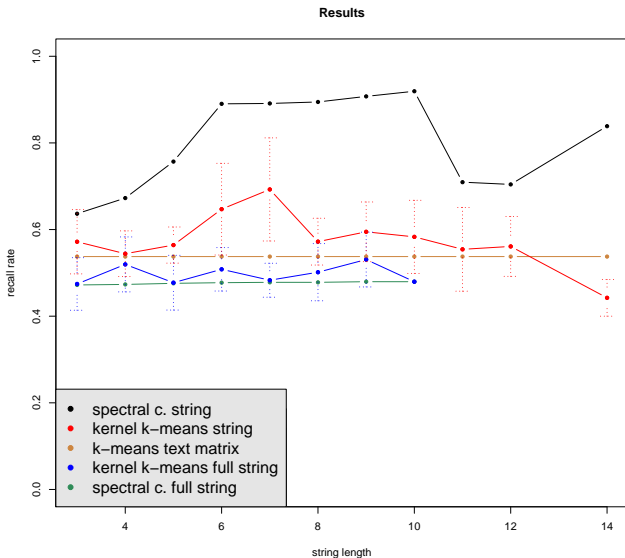
Recall Rate for Reuters-21578 Sample



Recall Rate for Reuters-21578 Sample



Recall Rate for Reuters-21578 Sample



Silhouette Plot

Silhouette plot for spectral clustering

$n = 1717$

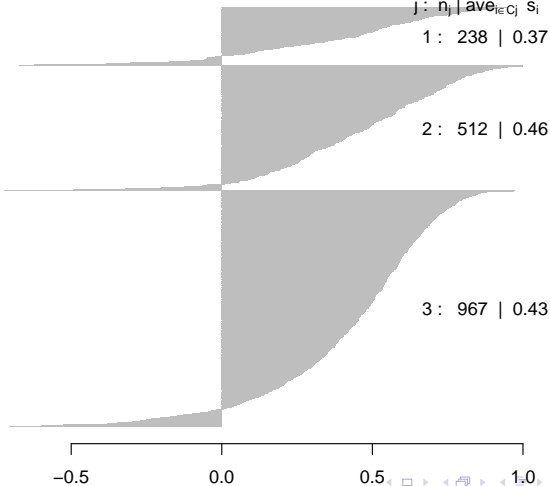
3 clusters C_j

$j: n_j \mid \text{ave}_{i \in C_j} s_i$

1: 238 | 0.37

2: 512 | 0.46

3: 967 | 0.43



Timings

- Experiments run on a Linux 2.6 GHz Pentium 4

kernel matrix calculations	$\approx 2\text{h}$
spectral clustering	$\approx 20\text{ sec.}$
kernel k-means	$\approx 30\text{ sec.}$
term matrix k-means	$\approx 40\text{ sec.}$

Summary

- Introduced a text processing framework in R
- We evaluated the performance of various text clustering methods
- Spectral clustering along with string kernels seems promising

- Outlook
 - Kernel matrix reduction for spectral clustering
 - Fast implementations of string kernels with suffix trees
 - Determine optimal string length and λ parameters