The theory of Delta measures

Stefan Evert

6 March 2015

Contents

1 Data sets

Load relative frequencies and z-scores for the German, English and French data set. For technical reasons, the data structures store the transposed document-term matrices $\mathbf{F}^T$ and $\mathbf{Z}^T$

```
load("data/delta_corpus.rda")
## FreqDE, FreqEN, FreqFR ... text-word matrix with absolute and relative frequencies
## zDE, zEN, zFR          ... standardized (z-transformed) relative frequencies
## goldDE, goldEN, goldFR ... gold standard labels (= author names)
```

- $\mathbf{F}^T$ is available under the names `FreqDE$S`, `FreqEN$S` and `FreqFR$S`
- $\mathbf{Z}^T$ is available under the names `zDE`, `zEN` and `zFR`
- absolute frequencies $n_{D_j} \cdot f_i(D_j)$ can be found in `FreqDE$M`, `FreqEN$M`, `FreqFR$M`

2 Notation

Our notation follows Argamon (2008) and Jannidis et al. (2015):

- given a collection of text documents $D \in \mathcal{D}$
  - $n_{\mathcal{D}} = |\mathcal{D}|$ is the number of texts in the collection
  - n_D is the token count of text D
- the relative frequency of word w_i in text D is denoted by $f_i(D)$
  - not entirely clear that Argamon (2008) really means relative frequency, but everything else doesn't make much sense
  - the number of words taken into consideration as features is denoted by n_w
  - $f_i(D)$ may also be used more generally for the relative frequency of other features such as lemmas, n-grams, POS tags, …

- following Burrows (2002), word frequencies are usually standardized to z-scores

$$z_i(D) = \frac{f_i(D) - \mu_i}{\sigma_i}$$

where μ_i is the mean of f_i and σ_i its sample standard deviation (s.d.) across $\mathcal{D}$
  - note that the features f_i are standardized *individually*, i.e. based on their respective μ_i and σ_i only
  - we will sometimes also use $z_i(D)$ to refer more generally to any scaled relative frequencies
- each text is thus represented by a frequency profile $\mathbf{f}(D) \in \mathbb{R}^{n_w}$ or a vector of z-scores $\mathbf{z}(D) \in \mathbb{R}^{n_w}$
- Burrows Delta corresponds to *Manhattan distance* between z-score vectors:

$$\Delta_B(D, D') = \|\mathbf{z}(D) - \mathbf{z}(D')\|_1$$

  - this is Argamon's simplified version, which omits the factor $1/n_w$ from the original formula
- Quadratic Delta corresponds to squared *Euclidean distance*:

$$\Delta_Q(D, D') = \|\mathbf{z}(D) - \mathbf{z}(D')\|_2^2$$

  - we will often use the equivalent Euclidean distance $\sqrt{\Delta_Q}$ instead
- Cosine Delta corresponds to *angular distance*, which can be computed from cosine similarity

$$\cos \Delta_\angle(D, D') = \frac{\mathbf{z}(D)^T \mathbf{z}(D')}{\|\mathbf{z}(D)\|_2 \|\mathbf{z}(D')\|_2}$$

  - for normalized vectors $\|\mathbf{z}(D)\|_2 = 1$, angular distance corresponds to spherical distance between points on the unit sphere and is equivalent to the Euclidean distance between those points
  - in other words, Cosine Delta is equivalent to Quadratic Delta with an implicit vector normalization
  - computation of Cosine Delta simplifies to $\Delta_\angle(D, D') = \cos^{-1} \mathbf{z}(D)^T \mathbf{z}(D')$ in this case
- these vectors form the columns of a $n_w \times n_\mathcal{D}$ term-document matrix denoted by
  - $\mathbf{F} = (f_{ij})$ where $f_{ij} = f_i(D_j)$ is the relative frequency of w_i in the j-th text, and
  - $\mathbf{Z} = (z_{ij})$ where $z_{ij} = z_i(D_j)$ is the corresponding standardized z-score
- some measures suggested by Argamon (2008) make use of the covariance matrix $\mathbf{S} = (\sigma_{ij}) = \mathrm{Cov}(\mathbf{F}^T)$ of the variables f_i
  - $\mathbf{S}$ can be computed from the centered matrix $\bar{\mathbf{F}} = (f_{ij} - \mu_i)$ as a cross-product $\mathbf{S} = \frac{1}{n_\mathcal{D}-1} \bar{\mathbf{F}} \bar{\mathbf{F}}^T$
  - the diagonal elements of $\mathbf{S}$ correspond to the variances of the individual variables: $\sigma_{ii} = (\sigma_i)^2$
  - note that Argamon (p. 140) specifies equations for *population* covariances (with denominator $n_\mathcal{D}$) rather than the more appropriate *sample* covariances (with denominator $n_\mathcal{D} - 1$)
  - the corresponding cross-product of $\mathbf{Z}$ yields the correlation matrix $\mathrm{Cor}(\mathbf{F}^T) = \frac{1}{n_\mathcal{D}-1} \mathbf{Z}\mathbf{Z}^T$ with elements $\sigma_{ij} / \sqrt{\sigma_i \sigma_j}$; all diagonal elements are equal to 1, i.e. $\mathrm{diag}(\mathrm{Cor}(\mathbf{F}^T)) = \mathbf{1}_{n_w}$

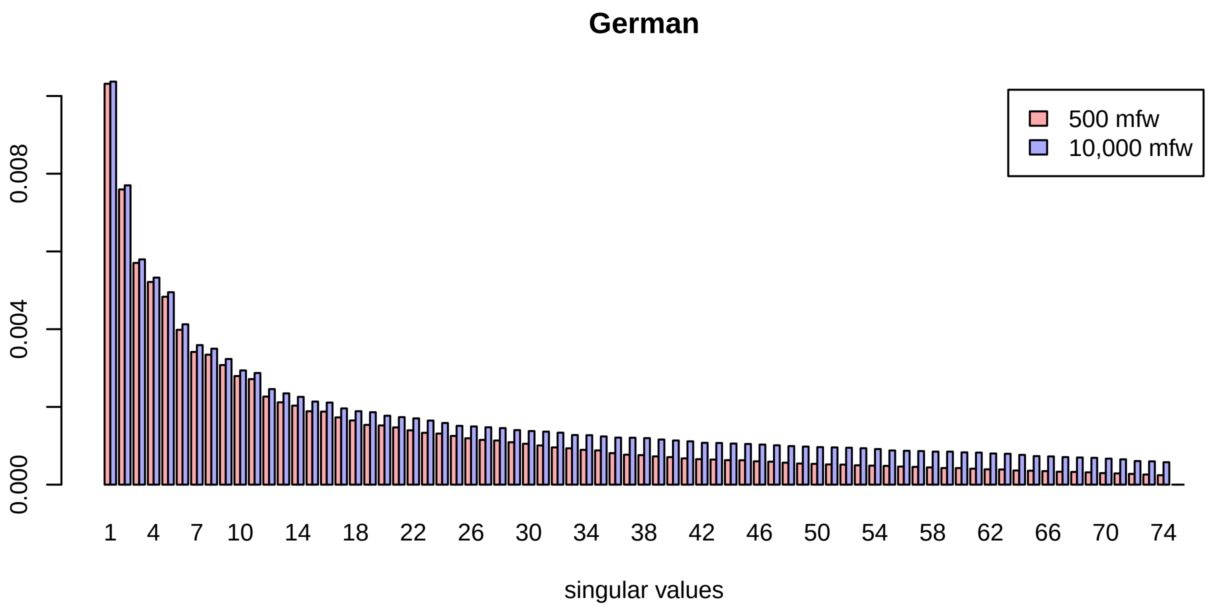
3 Quadratic Delta

4 Rotated Delta

- Rotated Delta Δ_R has been shown to perform very poorly, even though it has the most convincing mathematical justification of the various Delta measures (Argamon 2008)
- intuitively, it performs a standardization similar to the z-transformation, but takes correlations between the word frequencies into account; the scaling factors are $1/\lambda_i$ instead of $1/\sigma_i$
- if some of the λ_i are very small, this might give too much weight to "noise" dimensions
- we can obtain the singular values λ_i from a Principal Component Analysis and visualize them in the form of a barplot; note that the principal dimensions and the distribution of singular values depends on the number n_w of features

```
res500 <- prcomp(FreqDE$S[, 1:500], center=TRUE) # based on relative frequencies
res10k <- prcomp(FreqDE$S[, 1:10000], center=TRUE)
```

```r
barplot(rbind(res500$sdev, res10k$sdev), beside=TRUE, names=1:75, space=c(0,.5),
        col=c("#FFAAAA", "#AAAAFF"), main="German", xlab="singular values")
legend("topright", inset=.02, legend=c("500 mfw", "10,000 mfw"),
       fill=c("#FFAAAA", "#AAAAFF"))
```

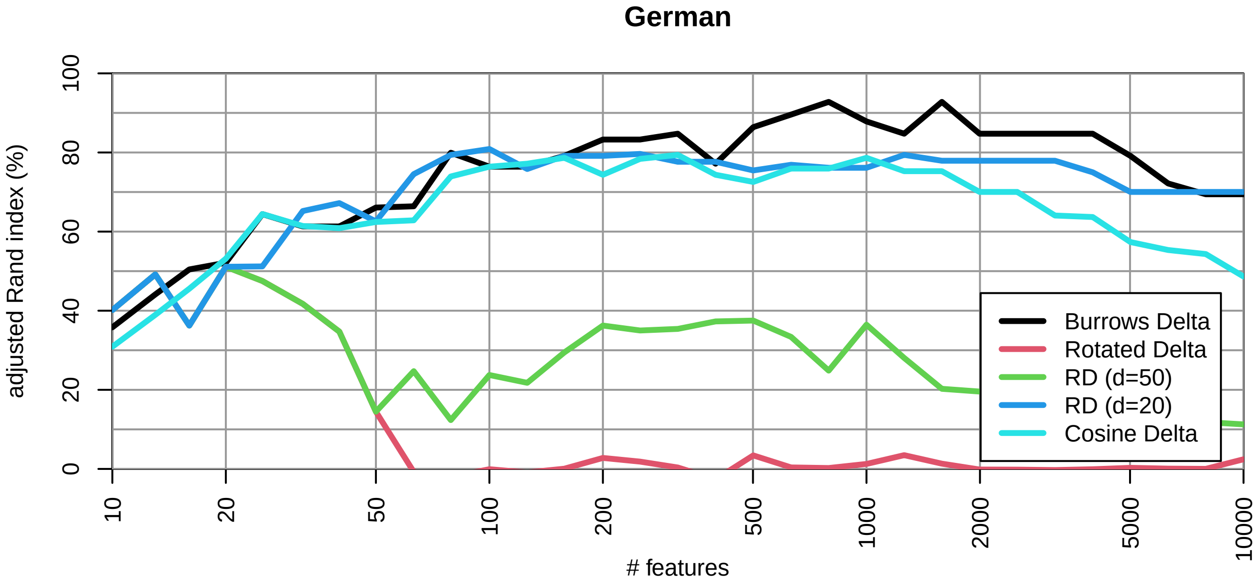


- notice that $\lambda_{75} \approx 0$ (because the 75 profile vectors span a 74-dimensional subspace); if it is naively included in the whitening, it might introduce a substantial amount of random noise
- a large part of the variance is captured by the first $d \approx 20$ principal coordinates; if only few features are used (e.g. $n_w = 500$), the remaining coordinates contribute relatively little to text distances; with a higher-dimensional feature space (e.g. $n_w = 10000$), the structure becomes more complex and variance spreads to higher singular values
- truncated PCA is often seen as a "noise reduction" technique, especially in distributional semantics
- the intuition is that the later principal coordinates with small singular values contain mostly noise, whereas the first coordinates capture the main structure of the data set
- truncating the transformed vectors to $d < n_{\mathcal{D}-1}$ dimensions should thus make distances (i.e. Rotated Delta) more meaningful and improve authorship attribution
- the evaluation plots below show that this is not the case at all: Δ_R performs very poorly, as in previous evaluations; truncation does indeed improve results, but it still remains inferior to Δ_B and $\Delta_\angle$

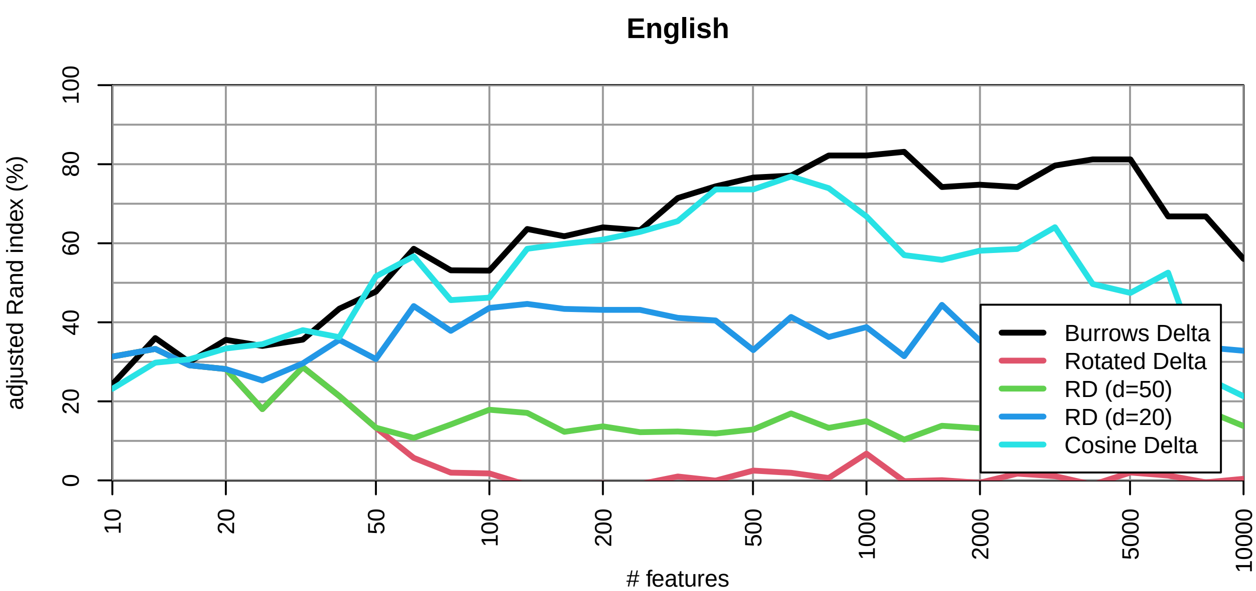
```r
n.vals <- round(10 ^ seq(1, 4, .1)) # logarithmic steps
draw.grid <- function () { # corresponding grid for plot region
  abline(h=seq(0, 100, 10), col="grey60")
  abline(v=c(10,20,50,100,200,500,1000,2000,5000,10000), col="grey60")
}

plot(1, 100, type="n", log="x", xlim=range(n.vals), ylim=c(0,100),
     xlab="# features", ylab="adjusted Rand index (%)", main="German",
     xaxs="i", yaxs="i", las=3, xaxp=c(range(n.vals), 3))
draw.grid()
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, meth="manh")$adj.rand, lwd=3, col=1)
lines(n.vals, evaluate(FreqDE$S, goldDE, n=n.vals, meth="eucl", pca=74)$adj.rand, lwd=3, col=2)
lines(n.vals, evaluate(FreqDE$S, goldDE, n=n.vals, meth="eucl", pca=50)$adj.rand, lwd=3, col=3)
```

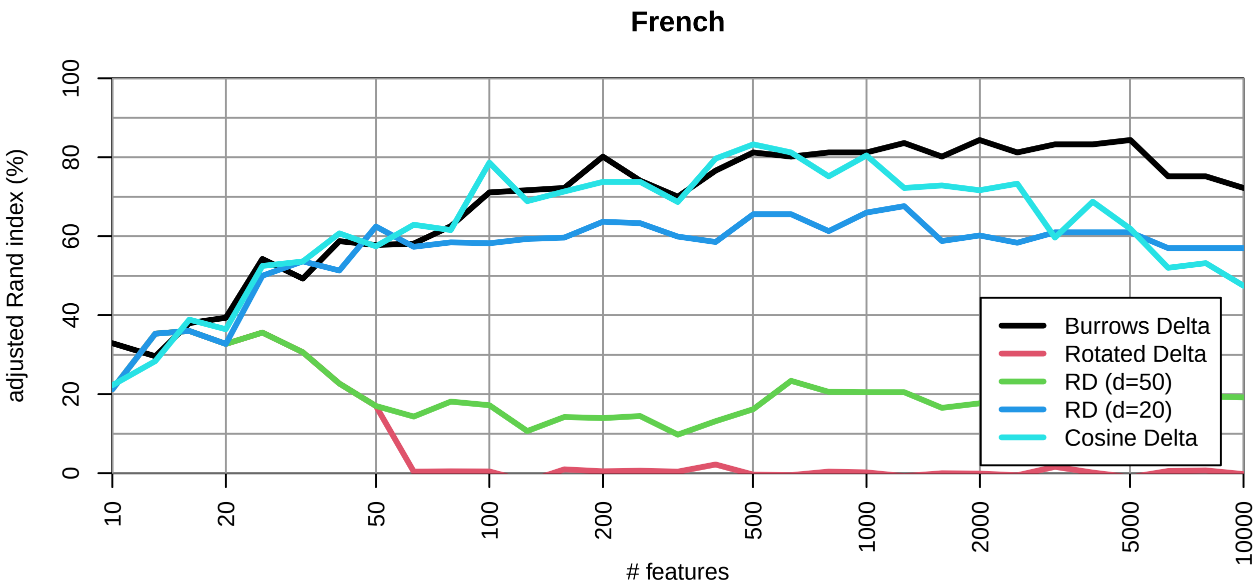
```
lines(n.vals, evaluate(FreqDE$S, goldDE, n=n.vals, meth="eucl", pca=20)$adj.rand, lwd=3, col=4)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, meth="eucl", norm="eucl")$adj.rand, lwd=3, col=5)
legend("bottomright", inset=.02, bg="white", lwd=3, col=1:5,
       legend=c("Burrows Delta", "Rotated Delta", "RD (d=50)", "RD (d=20)", "Cosine Delta"))
```




```
plot(1, 100, type="n", log="x", xlim=range(n.vals), ylim=c(0,100),
     xlab="# features", ylab="adjusted Rand index (%)", main="English",
     xaxs="i", yaxs="i", las=3, xaxp=c(range(n.vals), 3))
draw.grid()
lines(n.vals, evaluate(zEN, goldEN, n=n.vals, meth="manh")$adj.rand, lwd=3, col=1)
lines(n.vals, evaluate(FreqEN$S, goldEN, n=n.vals, meth="eucl", pca=74)$adj.rand, lwd=3, col=2)
lines(n.vals, evaluate(FreqEN$S, goldEN, n=n.vals, meth="eucl", pca=50)$adj.rand, lwd=3, col=3)
lines(n.vals, evaluate(FreqEN$S, goldEN, n=n.vals, meth="eucl", pca=20)$adj.rand, lwd=3, col=4)
lines(n.vals, evaluate(zEN, goldEN, n=n.vals, meth="eucl", norm="eucl")$adj.rand, lwd=3, col=5)
legend("bottomright", inset=.02, bg="white", lwd=3, col=1:5,
       legend=c("Burrows Delta", "Rotated Delta", "RD (d=50)", "RD (d=20)", "Cosine Delta"))
```

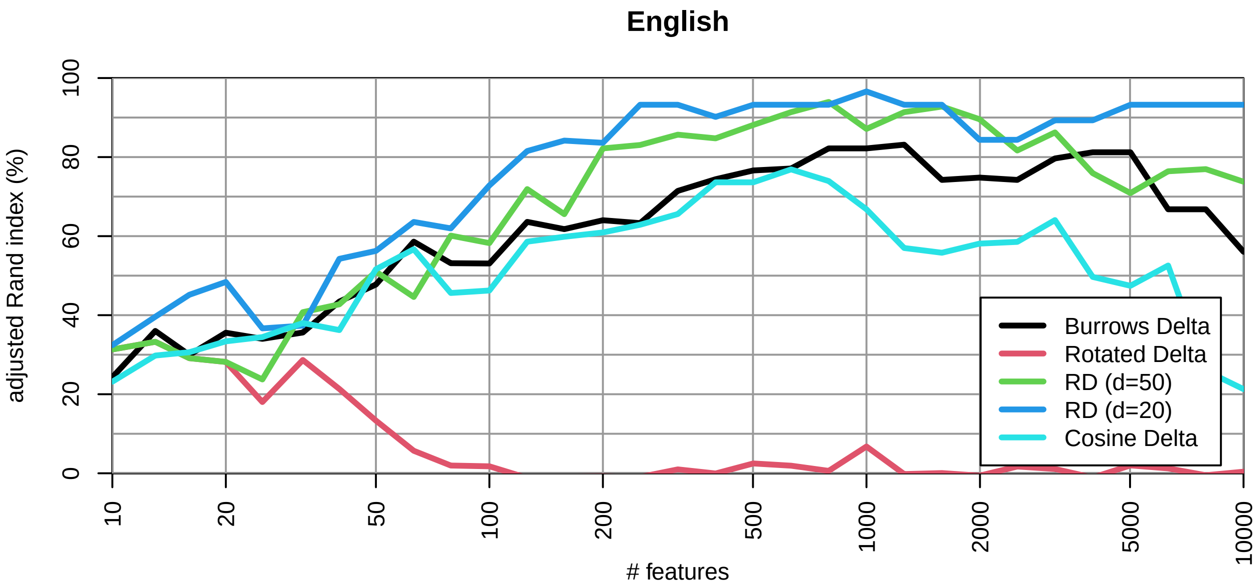


```r
plot(1, 100, type="n", log="x", xlim=range(n.vals), ylim=c(0,100),
     xlab="# features", ylab="adjusted Rand index (%)", main="French",
     xaxs="i", yaxs="i", las=3, xaxp=c(range(n.vals), 3))
draw.grid()
lines(n.vals, evaluate(zFR, goldFR, n=n.vals, meth="manh")$adj.rand, lwd=3, col=1)
lines(n.vals, evaluate(FreqFR$S, goldFR, n=n.vals, meth="eucl", pca=74)$adj.rand, lwd=3, col=2)
lines(n.vals, evaluate(FreqFR$S, goldFR, n=n.vals, meth="eucl", pca=50)$adj.rand, lwd=3, col=3)
lines(n.vals, evaluate(FreqFR$S, goldFR, n=n.vals, meth="eucl", pca=20)$adj.rand, lwd=3, col=4)
lines(n.vals, evaluate(zFR, goldFR, n=n.vals, meth="eucl", norm="eucl")$adj.rand, lwd=3, col=5)
legend("bottomright", inset=.02, bg="white", lwd=3, col=1:5,
       legend=c("Burrows Delta", "Rotated Delta", "RD (d=50)", "RD (d=20)", "Cosine Delta"))
```



- **TODO:** some random experiments
- PCA as additional "noise reduction" on Cosine Delta could be promising, but only if you get the number d of latent dimensions exactly right

```r
plot(1, 100, type="n", log="x", xlim=range(n.vals), ylim=c(0,100),
     xlab="# features", ylab="adjusted Rand index (%)", main="English",
     xaxs="i", yaxs="i", las=3, xaxp=c(range(n.vals), 3))
draw.grid()
lines(n.vals, evaluate(zEN, goldEN, n=n.vals, meth="manh")$adj.rand, lwd=3, col=1)
lines(n.vals, evaluate(FreqEN$S, goldEN, n=n.vals, meth="eucl", pca=74)$adj.rand, lwd=3, col=2)
lines(n.vals, evaluate(zEN, goldEN, n=n.vals, meth="eucl", pca=20)$adj.rand, lwd=3, col=3)
lines(n.vals, evaluate(zEN, goldEN, n=n.vals, meth="eucl", pca=30, prenorm=TRUE, norm="euclidean")$adj.r
lines(n.vals, evaluate(zEN, goldEN, n=n.vals, meth="eucl", norm="eucl")$adj.rand, lwd=3, col=5)
legend("bottomright", inset=.02, bg="white", lwd=3, col=1:5,
       legend=c("Burrows Delta", "Rotated Delta", "RD (d=50)", "RD (d=20)", "Cosine Delta"))
```



5 Problematic assumptions

TODO: compare Rotated Delta with and without whitening (the latter is similar to LSA-style approaches); screeplot of singular values; truncate small singular values before whitening, or use some form of "soft" whitening

References

Argamon, Shlomo. 2008. "Interpreting Burrows's Delta: Geometric and Probabilistic Foundations." *Literary and Linguistic Computing* 23 (2): 131–47. https://doi.org/10.1093/llc/fqn003.

Burrows, John. 2002. "'Delta': A Measure of Stylistic Difference and a Guide to Likely Authorship." *Literary and Linguistic Computing* 17 (3): 267–87. https://doi.org/10.1093/llc/17.3.267.

Jannidis, Fotis, Steffen Pielström, Christof Schöch, and Thorsten Vitt. 2015. "Improving Burrows' Delta. An Empirical Evaluation of Text Distance Measures." In *Proceedings of the Digital Humanities Conference 2015*. Sydney, Australia.