Burrows Delta: The effects of vector normalization

Stefan Evert

7 April 2015

Contents

1 Data sets and setup

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`

Evaluation steps and corresponding grid for the plots:

```
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")
}
```

2 Vector normalization improves Delta measures

2.1 Cosine and normalized Quadratic Delta

Cosine distance is equivalent to Euclidean distance between L_2-normalized vectors. In other words, the difference in performance between Δ_Q and $\Delta_\angle$ results from vector normalization rather than a genuinely different approach to measuring distances. As a confirmation, here is the evaluation of the German data with an additional line for Δ_Q on normalized vectors (note that the new line hides the identical line of $\Delta_\angle$).

```
plot(1, 100, type="n", log="x", xlim=range(n.vals), ylim=c(0,100),
     xlab="# features", ylab="adjusted Rand index (%)", main="German (z-scores)",
     xaxs="i", yaxs="i", las=3, xaxp=c(range(n.vals), 3))
draw.grid()
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, method="manhattan")$adj.rand, lwd=3, col=1)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, method="euclidean")$adj.rand, lwd=3, col=2)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, method="cosine")$adj.rand, lwd=3, col=3)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, meth="eucl", norm="eucl")$adj.rand, lwd=3, col=4)
legend("bottomright", inset=.02, bg="white", lwd=3, col=1:4,
       legend=c("Burrows Delta", "Quadratic Delta", "Cosine Delta", "QD (normalized)"))
```

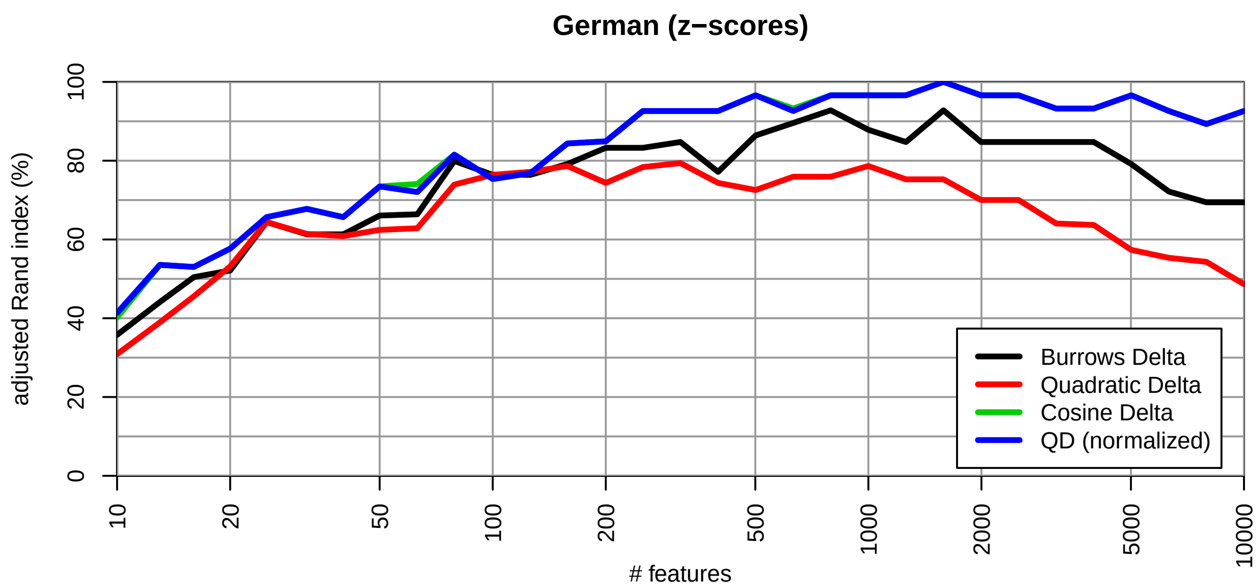
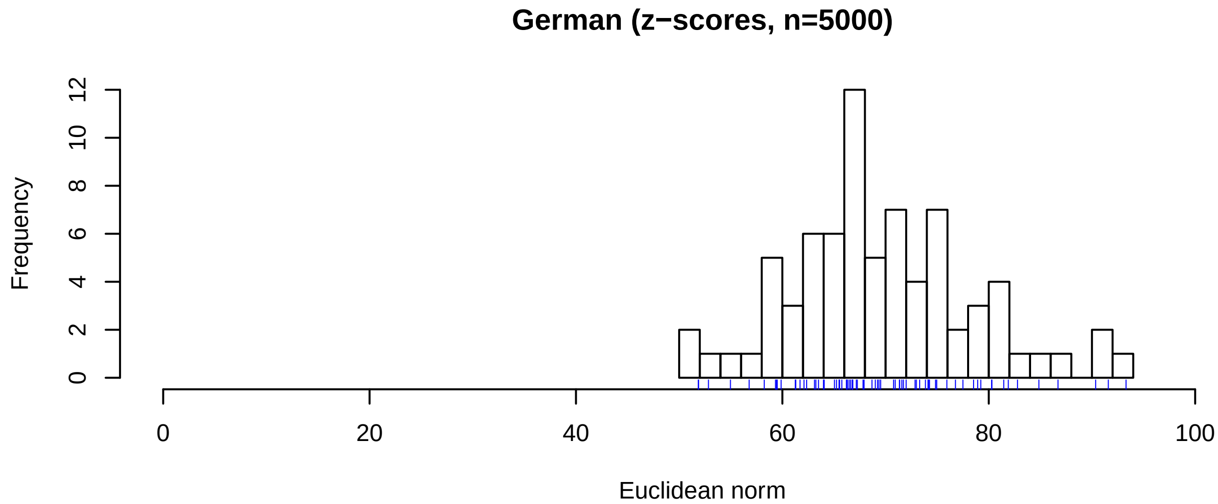


Figure 1:

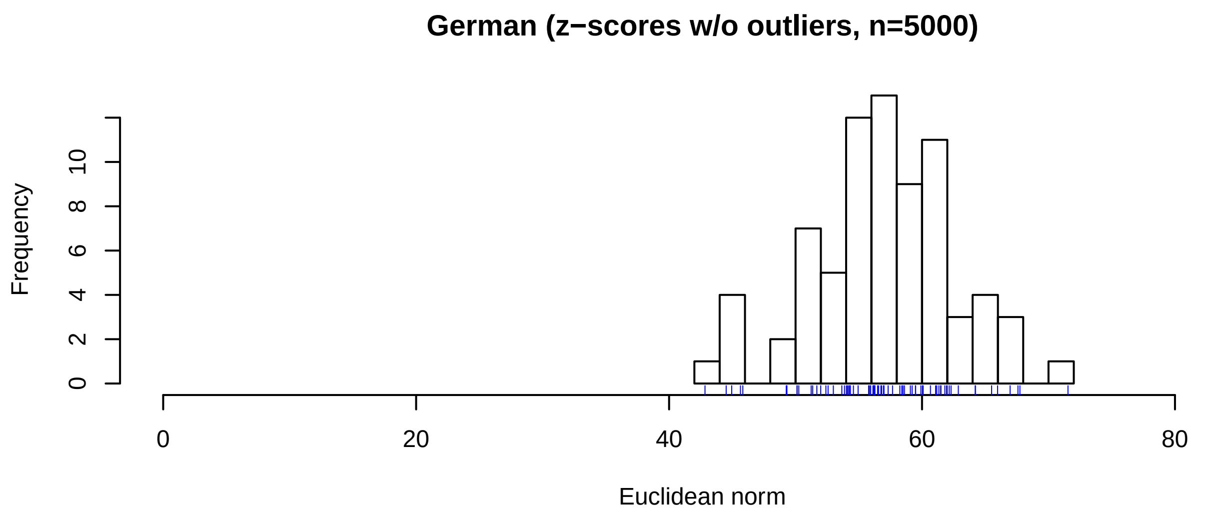
This can only happen if there are considerable differences in the lengths of different vectors, which is confirmed by a histogram plot. We compute the histogram for $n_w = 5000$ where Δ_Q already performs much worse than the robust alternatives:

```
x <- rowNorms(zDE[, 1:5000], method="euclidean")
hist(x, breaks=20, xlim=c(0, 1.5 * mean(x)),
     xlab="Euclidean norm", main="German (z-scores, n=5000)")
rug(x, col=4)
```



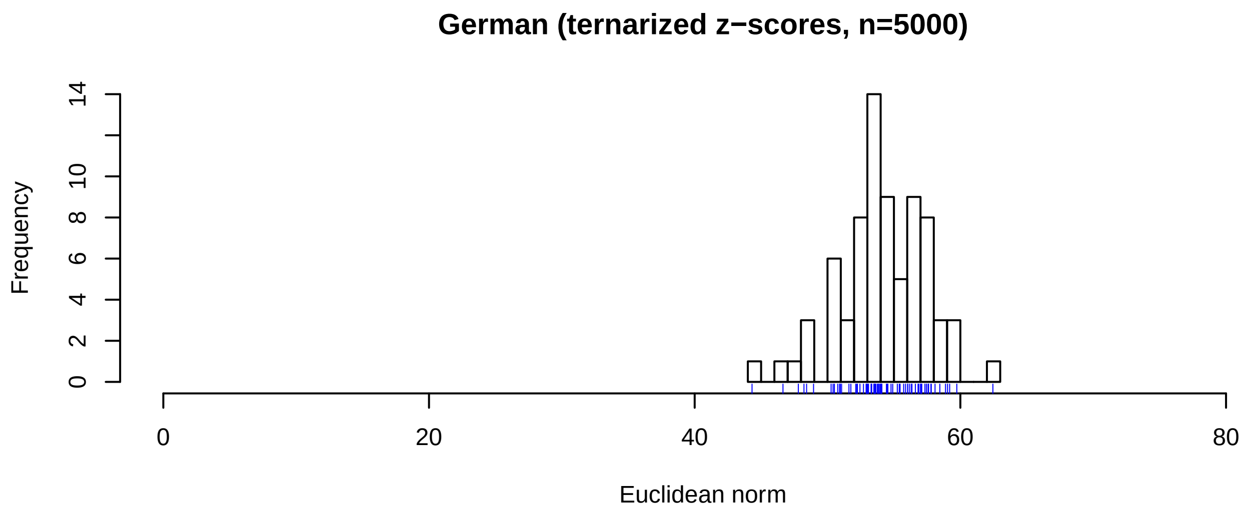
This histogram shows a considerable – though not dramatic – spread of the vector lengths, which might indeed have an effect on clustering by Euclidean distance. Truncating outlier z-scores ($|z| > 2$) improves the distribution, especially for a few extremely texts with very large L_2 norm.

```
x <- rowNorms(clamp(zDE[, 1:5000], min=-2, max=2), method="euclidean")
hist(x, breaks=20, xlim=c(0, 1.5 * mean(x)),
     xlab="Euclidean norm", main="German (z-scores w/o outliers, n=5000)")
rug(x, col=4)
```



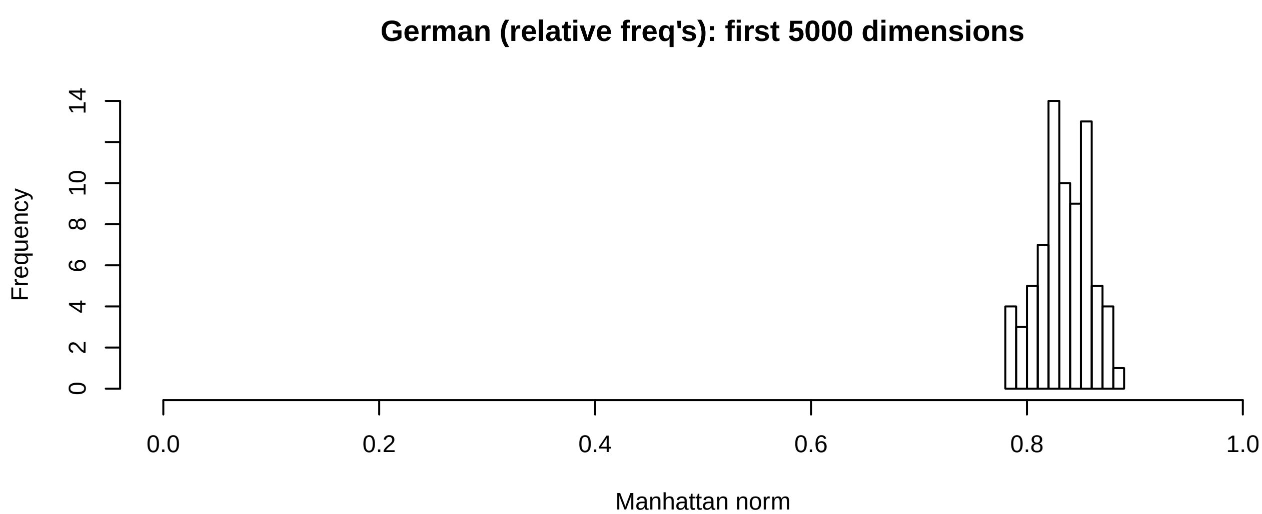
Ternarization results in a much narrower spread, with only two or three outlier texts, providing support for the hypothesis that the pattern of positive and negative deviations contains the "author signal" while the amplitude of these deviations is a nuisance factor.

```
x <- rowNorms(ternarize(zDE[, 1:5000], neutral.p=1/3), method="euclidean")
hist(x, breaks=20, xlim=c(0, 1.5 * mean(x)),
     xlab="Euclidean norm", main="German (ternarized z-scores, n=5000)")
rug(x, col=4)
```



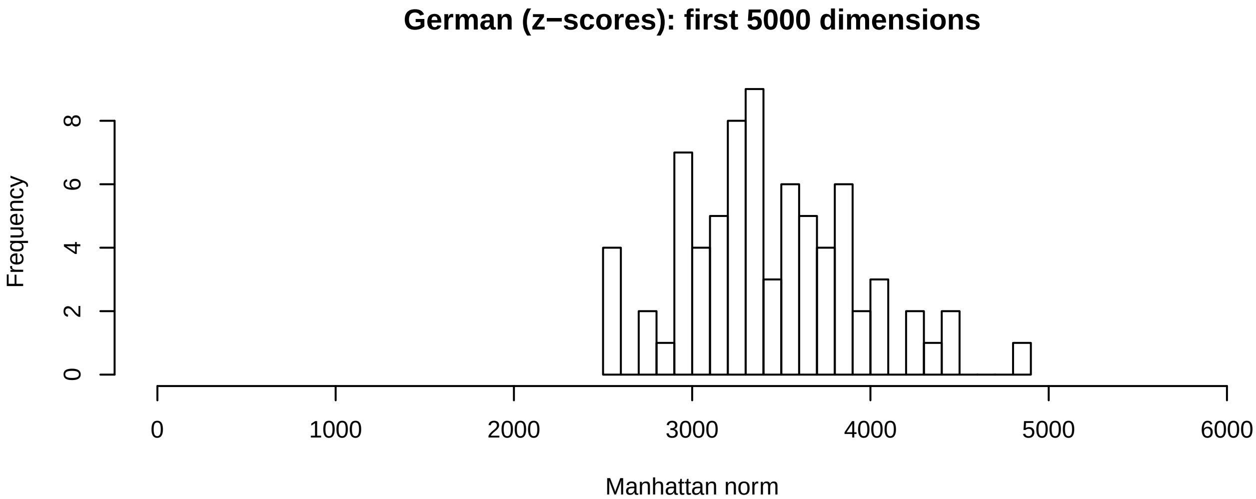
One might think that vectors of relative frequencies should be normalized with respect to the Manhattan norm (i.e. L_1-normalized), but this is no longer the case if we select the first n_w dimensions.

```
hist(rowNorms(FreqDE$S[, 1:5000], method="manhattan"), breaks=10, xlim=c(0,1),
     xlab="Manhattan norm", main="German (relative freq's): first 5000 dimensions")
```



Most of the variability is introduced by the transformation to z-scores, however, which undoes the L_1-normalization:

```
hist(rowNorms(zDE[, 1:5000], method="manhattan"), breaks=20, xlim=c(0,6000),
     xlab="Manhattan norm", main="German (z-scores): first 5000 dimensions")
```



Unsurprisingly, the effect of normalization on Δ_Q is confirmed by the results obtained on the English and French data sets:

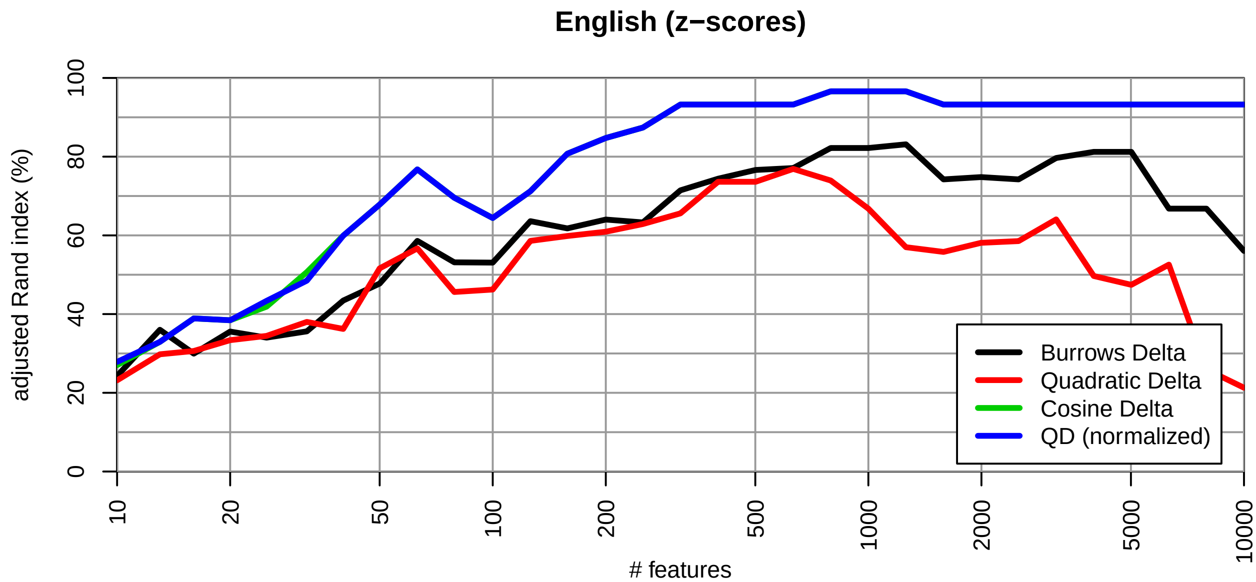
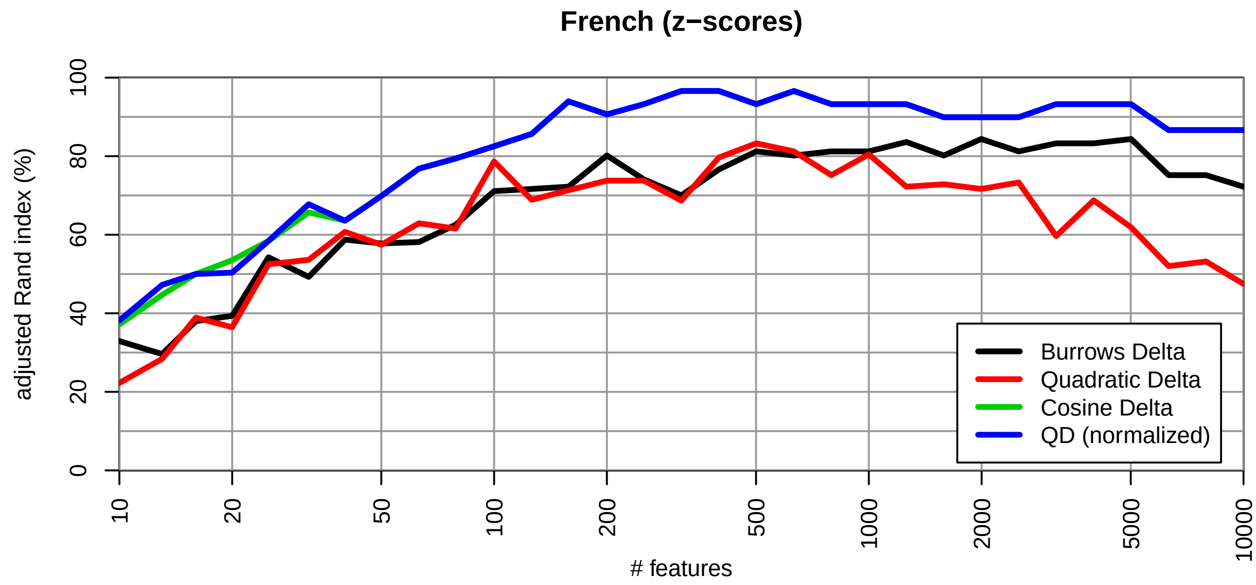


Figure 2:



2.2 Normalization for Burrows Delta

Normalization also improves Burrows Delta, which is then almost on par with Cosine Delta. Interestingly, Δ_B with (inappropriate) Euclidean normalization is even a little better on average than with the mathematically sensible Manhattan normalization. The combination of Δ_Q with L_1-normalization also works quite well, but is inferior to Cosine Delta for German.

```
plot(1, 100, type="n", log="x", xlim=range(n.vals), ylim=c(0,100),
     xlab="# features", ylab="adjusted Rand index (%)", main="German (z-scores)",
     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")$adj.rand, lwd=3, col=1)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, method="euclidean")$adj.rand, lwd=3, col=2)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, meth="cosine")$adj.rand, lwd=3, col=3)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, meth="manh", norm="manh")$adj.rand, lwd=3, col=4)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, meth="manh", norm="eucl")$adj.rand, lwd=3, col=8)
lines(n.vals, evaluate(zDE, goldDE, n=n.vals, meth="eucl", norm="manh")$adj.rand, lwd=3, col=6)
legend("bottomright", inset=.02, bg="white", lwd=3, col=c(1:4, 8, 6),
       legend=c("Burrows Delta", "Quadratic Delta", "Cosine Delta",
                "Burrows Delta / L1", "Burrows Delta / L2", "Quadratic Delta / L1"))
```

We can also look at the effect of normalization on a wider range of p-norms (which we might call L_p-Delta, or perhaps simply Δ_p). In order to keep the display readable, we need to make three separate plots for unnormalized as well as L_1- and L_2-normalized vectors.

```
for (run in list(c("not ", NA), c("L2-", "euclidean"), c("L1-", "manhattan"), c("L4-", "minkowski"))) {
  title <- sprintf("German (z-scores, %snormalized)", run[1])
  plot(1, 100, type="n", log="x", xlim=range(n.vals), ylim=c(0,100),
       xlab="# features", ylab="adjusted Rand index (%)", main=title,
       xaxs="i", yaxs="i", las=3, xaxp=c(range(n.vals), 3))
  draw.grid()
  lines(n.vals, evaluate(zDE, goldDE, n=n.vals, norm=run[2], norm.p=4, method="manh")$adj.rand, lwd=3,
```

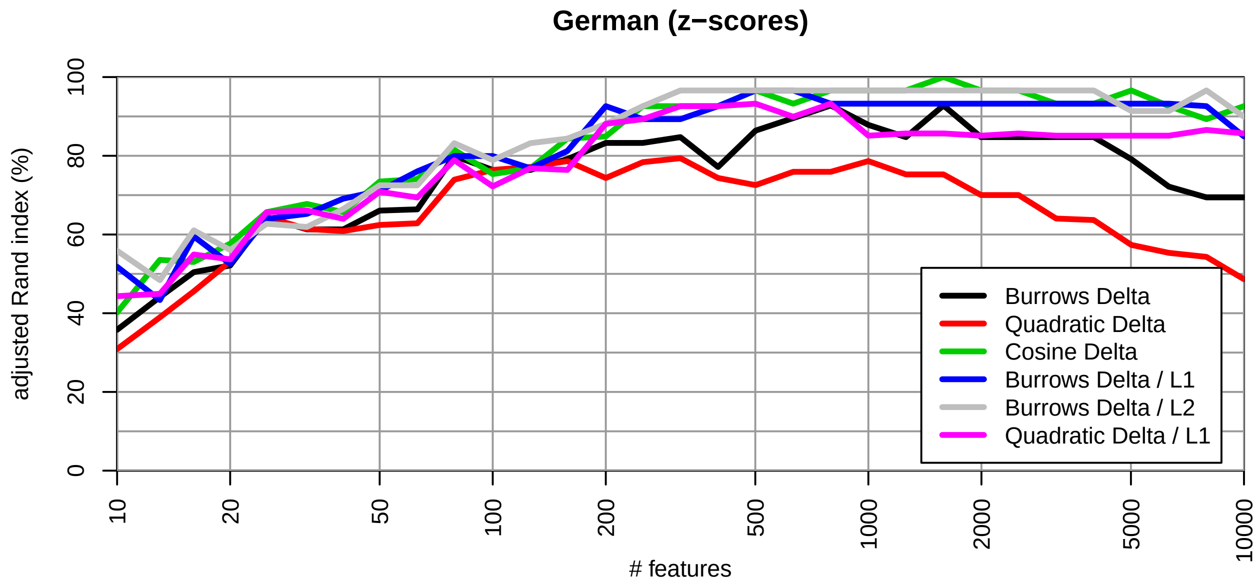


Figure 3:

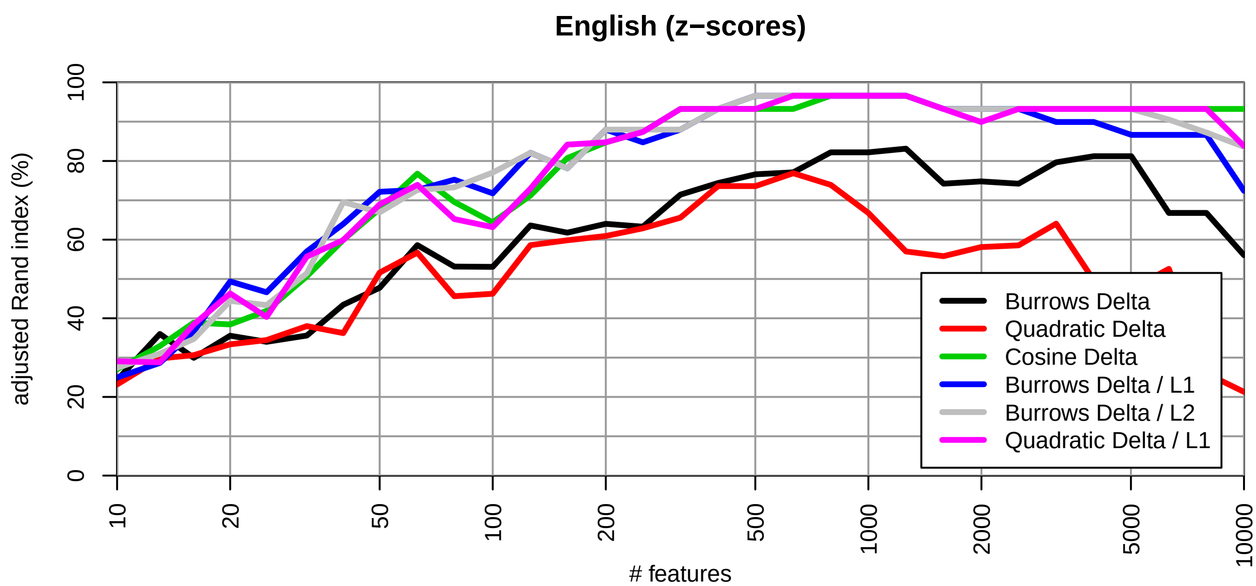


Figure 4:

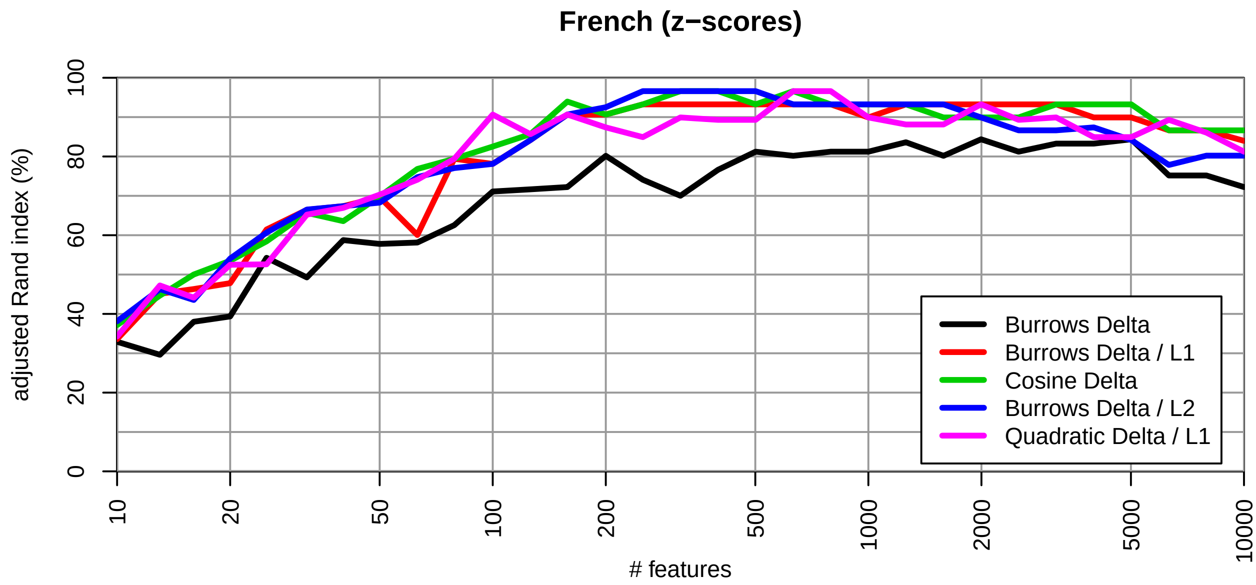
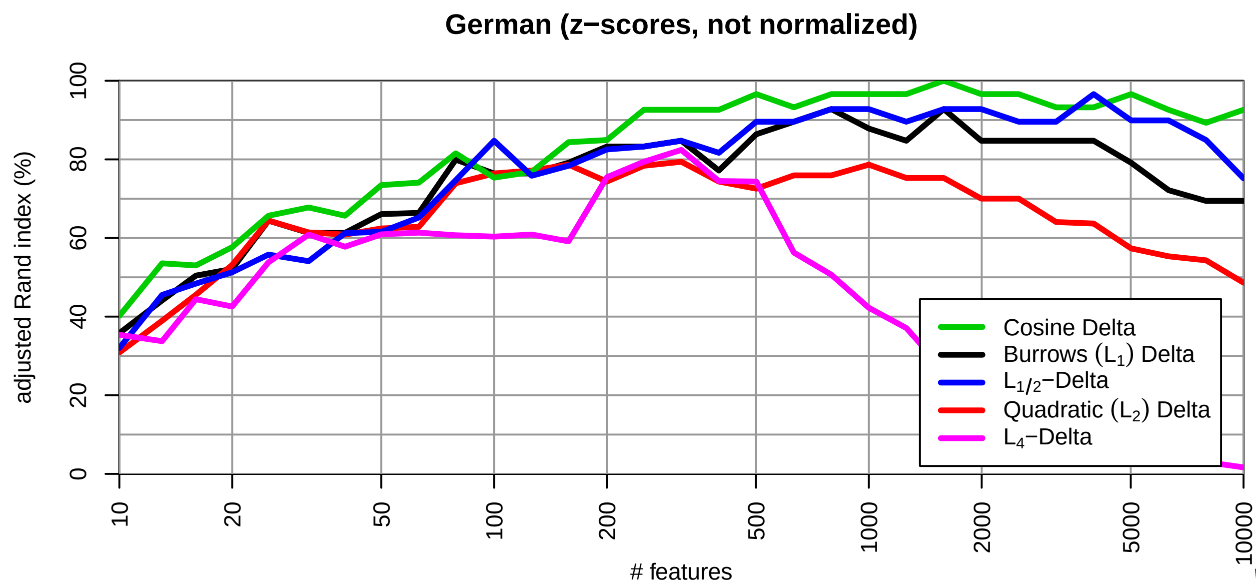


Figure 5:

```
  lines(n.vals, evaluate(zDE, goldDE, n=n.vals, norm=run[2], norm.p=4, method="eucl")$adj.rand, lwd=3, c
  lines(n.vals, evaluate(zDE, goldDE, n=n.vals, norm=run[2], norm.p=4, method="cosine")$adj.rand, lwd=3
  lines(n.vals, evaluate(zDE, goldDE, n=n.vals, norm=run[2], norm.p=4, method="mink", p=0.5)$adj.rand, l
  lines(n.vals, evaluate(zDE, goldDE, n=n.vals, norm=run[2], norm.p=4, method="mink", p=4)$adj.rand, lwd
  legend("bottomright", inset=.02, bg="white", lwd=3, col=c(3,1,4,2,6),
         legend=expression("Cosine Delta", "Burrows "*(L[1])*" Delta", L[1/2]*"-Delta",
                           "Quadratic "*(L[2])*" Delta", L[4]*"-Delta"))
}
```



!!!

2.3 Conclusion

Normalization improves both Δ_B and Δ_Q considerably. It also makes both measures robust wrt. the number of features and (at least partly) the feature selection strategy. It appears that vector normalization is the most important factor for successful authorship attribution with Delta measures. Surprisingly, it does not seem to play a role which norm is applied, even if it does not match the distance metric.

Why normalization has such a beneficial effect – and why it seems to be a much more important factor than the distance measure used – is still a mystery, though.

3 Vector length as deviation from the norm

One speculative explanation is that each author has a characteristic stylistic profile of deviations from the "norm", i.e. which words are used more frequently and which are used less frequently than the average. However, this profile is not expressed to the same degree in all texts (e.g. because of constraints imposed by sub-genre, editor, "tone" of the narrative). In this case, authorial style is conveyed by the *pattern* of deviations from the average, not by the overall *magnitude* of these deviations. Since the latter is directly connected with the length of the feature vector, normalization removes the irrelevant magnitude information and thus brings out authorial style more clearly.

If this explanation holds true, texts from the same author should exhibit considerable differences in vector length. Otherwise the degree of deviation from the norm would be a characteristic aspect of this author's style and thus improve authorship attribution. In order to test this hypothesis, we plot average deviation – directly related to vector length – for each text. We compute separate values for positive and negative deviations (as a rough indicator of the stylistic profiles) and use them as coordinates of a scatterplot. The averages can be computed according to L_1 or L_2 norms, but this should not make a substantial difference.

The `mean.deviation()` function computes separate means over the positive and negative values in each row of matrix `M`; it returns a two-column matrix. Note that values of the opposite sign are substituted by 0 and included in the average, so the values returned correspond to the total negative and positive mass. The average is computed according to a Minkowski p-norm, usually with `p=1` (L_1, Manhattan) or `p=2` (L_2, Euclidean).

```
mean.deviation <- function (M, n=NA, p=2) {
  if (!is.na(n) && n < ncol(M)) M <- M[, 1:n]
  M.plus <- pmax(M, 0)
  M.minus <- pmin(M, 0)
  mean.plus <- (rowSums(abs(M.plus) ^ p) / ncol(M)) ^ (1/p)
  mean.minus <- (rowSums(abs(M.minus) ^ p) / ncol(M)) ^ (1/p)
  res <- cbind(mean.plus, mean.minus)
  colnames(res) <- c("positive", "negative")
  rownames(res) <- rownames(M)
  res
}
```

```
md.grid <- function (p=1, l=(0:20)/10) {
  if (p == 1) {
    for (r in l) abline(r, -1, col="grey70", lwd=(if (r == 1) 3 else 1))
  } else {
    phi <- seq(0, .5, length.out=50)
    xy <- cbind(cospi(phi), sinpi(phi))
    r <- rowSums(xy ^ p) ^ (1/p)
    xy <- scaleMargins(xy, rows=1 / r)
    for (r in l) lines(r * xy, col="grey70", lwd=(if (r == 1) 3 else 1))
```

```
  }
  abline(0, 1, lty="dashed", col="grey70")
}
author.col <- rep(c(1:4, 6), 5) # colours and symbols for 24 authors in the scatterplot
author.pch <- rep(1:5, each=5)
```

```
md1.DE <- mean.deviation(zDE, n=1500, p=1)
authorDE <- as.factor(goldDE)
plot(0, 0, type="n", xlim=c(0,.62), xaxs="i", ylim=c(.2,.62), yaxs="i",
     xlab="mean positive score", ylab="mean negative score", main="German (L1, 1500 mfw)")
md.grid(p=1)
points(md1.DE, col=author.col[authorDE], pch=author.pch[authorDE])
legend("topleft", inset=.02, legend=levels(authorDE),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
```

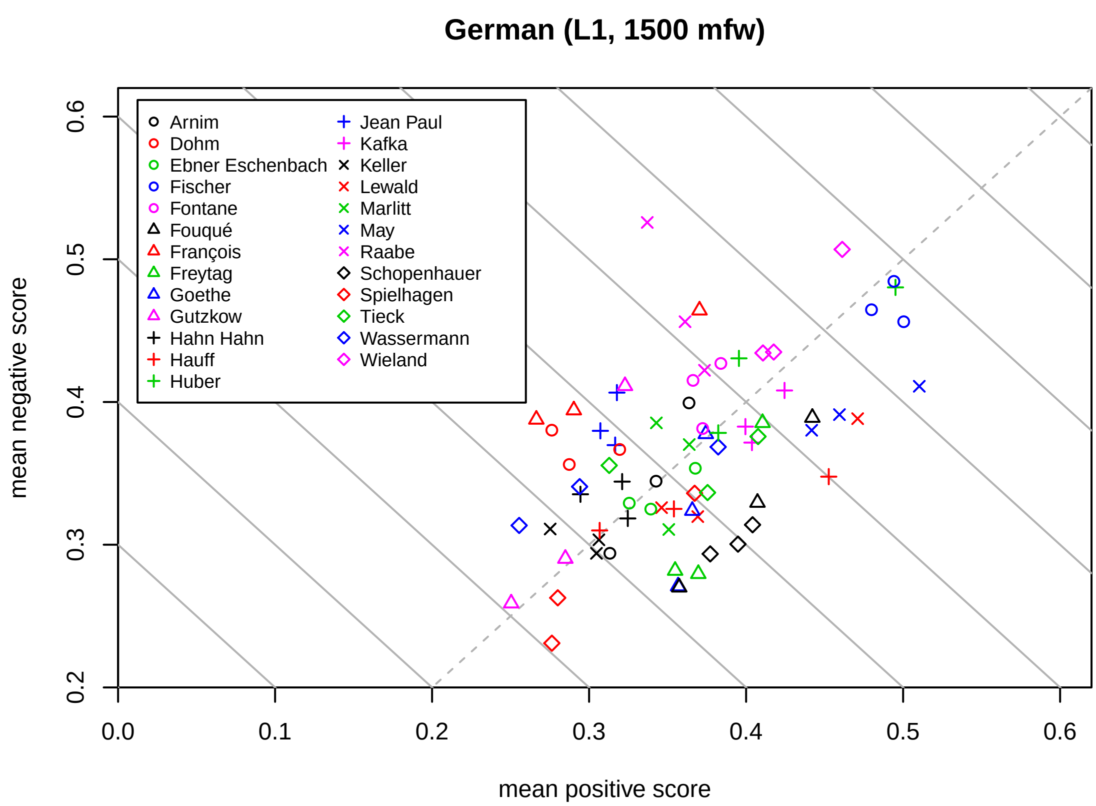


Figure 6:

```
md1.DE <- mean.deviation(zDE, n=750, p=1)
plot(0, 0, type="n", xlim=c(0,.62), xaxs="i", ylim=c(.2,.62), yaxs="i",
     xlab="mean positive score", ylab="mean negative score", main="German (L1, 750 mfw)")
md.grid(p=1)
points(md1.DE, col=author.col[authorDE], pch=author.pch[authorDE])
legend("topleft", inset=.02, legend=levels(authorDE),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
```

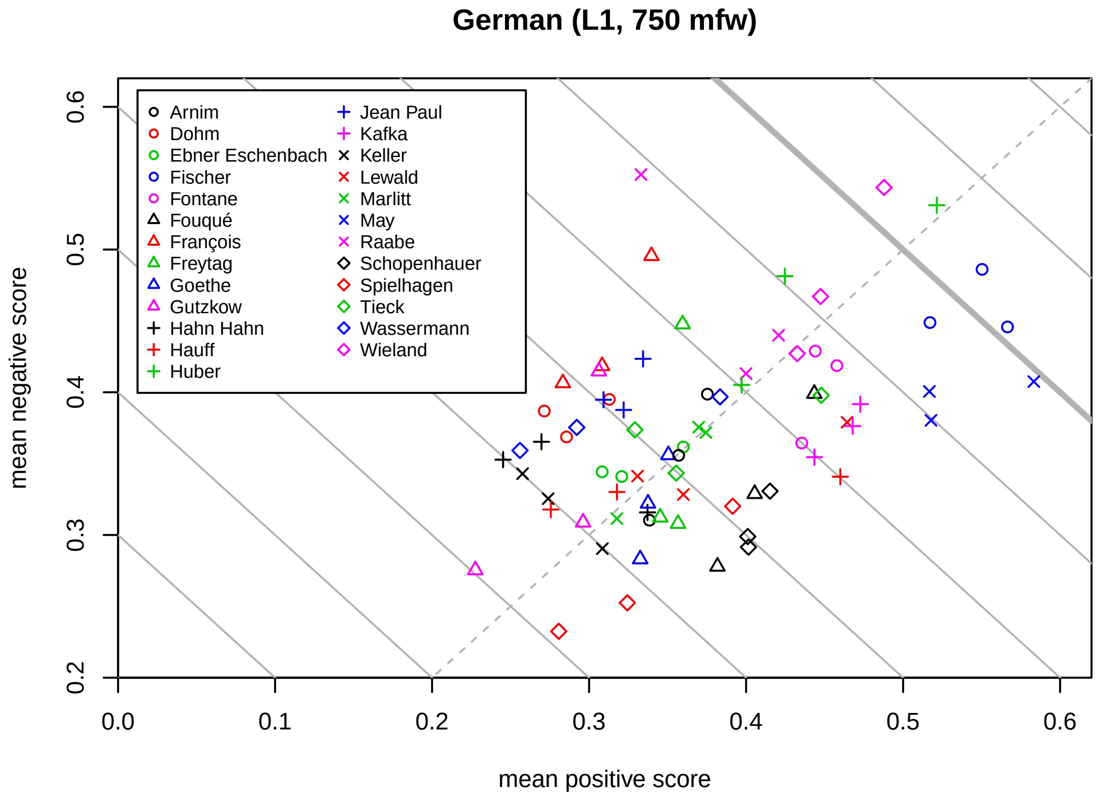


Figure 7:

```
md1.DE <- mean.deviation(zDE, n=10000, p=1)
plot(0, 0, type="n", xlim=c(0,.62), xaxs="i", ylim=c(.2,.62), yaxs="i",
     xlab="mean positive score", ylab="mean negative score", main="German (L1, 10000 mfw)")
md.grid(p=1)
points(md1.DE, col=author.col[authorDE], pch=author.pch[authorDE])
legend("topleft", inset=.02, legend=levels(authorDE),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
```

```
md2.DE <- mean.deviation(zDE, n=150, p=2)
plot(0, 0, type="n", xlim=c(0,1.2), xaxs="i", ylim=c(.2,1.0), yaxs="i",
     xlab="mean positive score", ylab="mean negative score", main="German (L2, 150 mfw)")
md.grid(p=2)
points(md2.DE, col=author.col[authorDE], pch=author.pch[authorDE])
legend("topleft", inset=.02, legend=levels(authorDE),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
```

```
md2.DE <- mean.deviation(zDE, n=500, p=2)
plot(0, 0, type="n", xlim=c(0,1.2), xaxs="i", ylim=c(.2,1.0), yaxs="i",
     xlab="mean positive score", ylab="mean negative score", main="German (L2, 500 mfw)")
md.grid(p=2)
points(md2.DE, col=author.col[authorDE], pch=author.pch[authorDE])
legend("topleft", inset=.02, legend=levels(authorDE),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
```

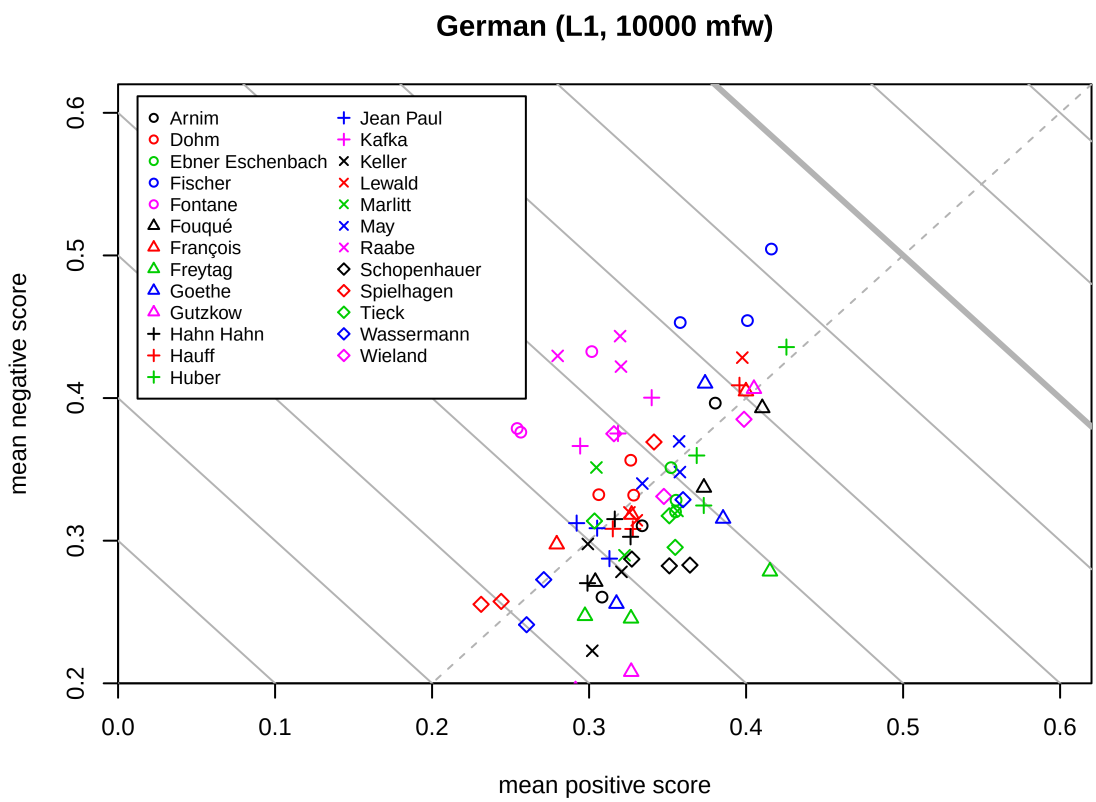


Figure 8:

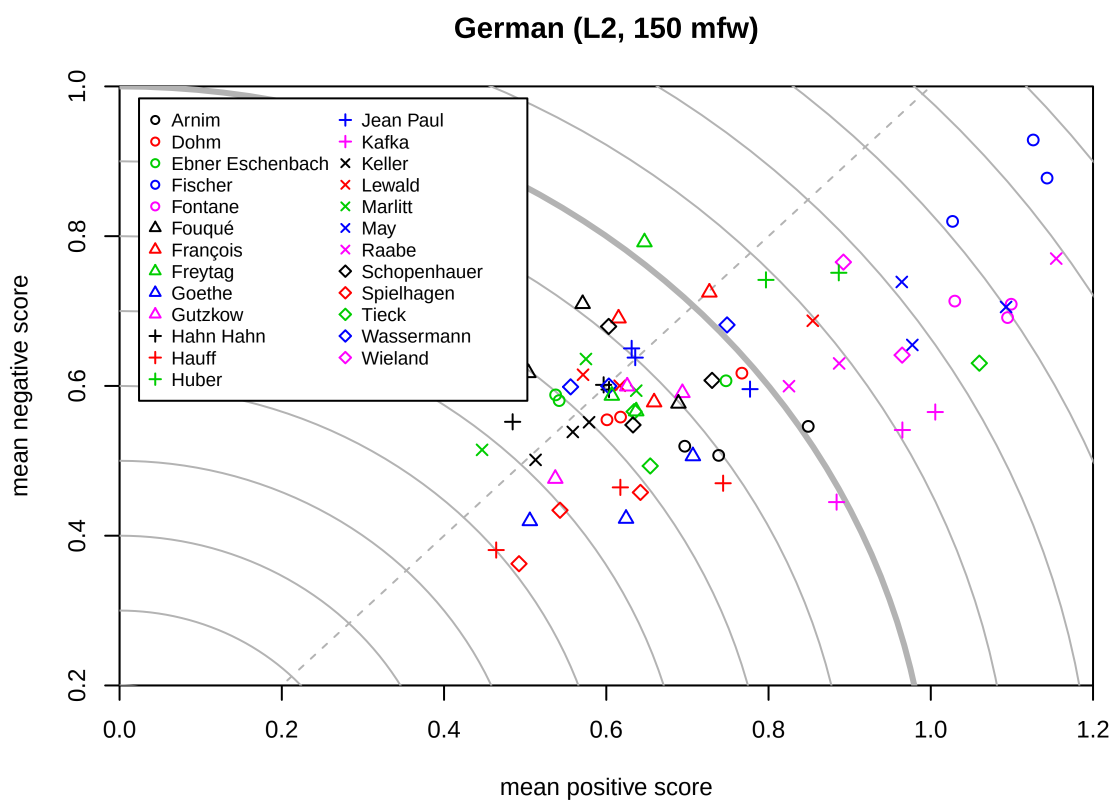
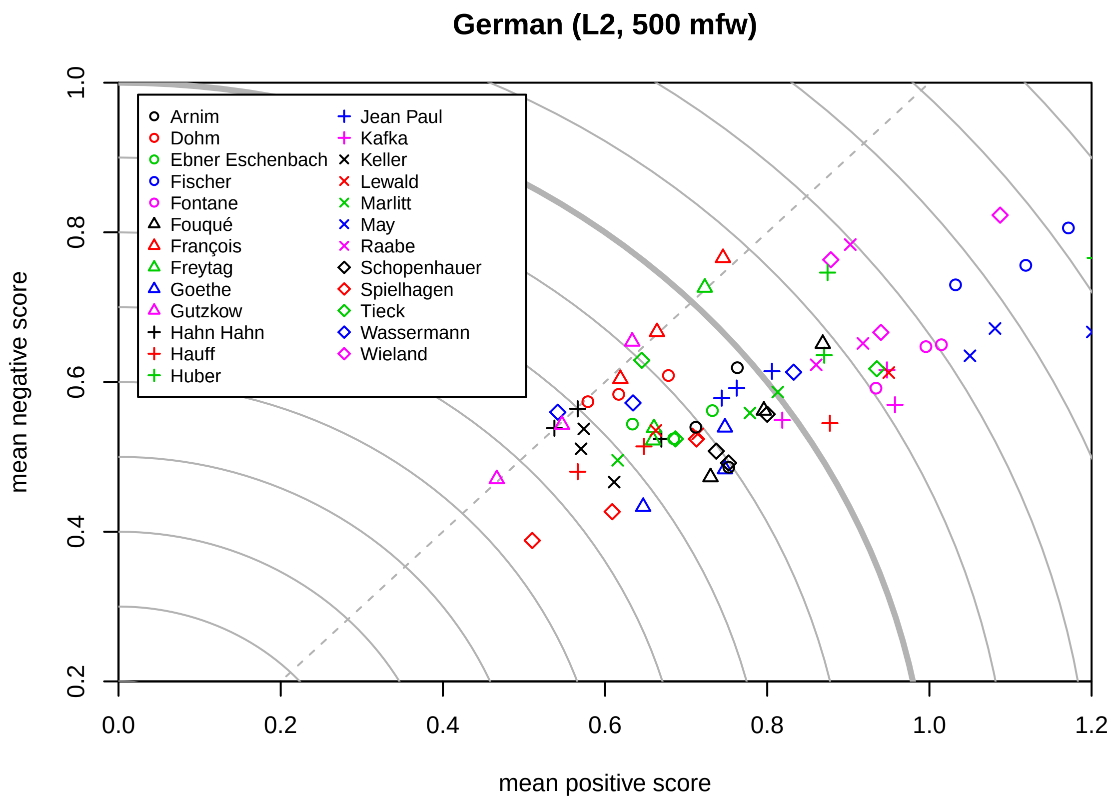


Figure 9:




```
md2.DE <- mean.deviation(zDE, n=1500, p=2)
plot(0, 0, type="n", xlim=c(0,1.2), xaxs="i", ylim=c(.2,1.0), yaxs="i",
     xlab="mean positive score", ylab="mean negative score", main="German (L2, 1500 mfw)")
md.grid(p=2)
points(md2.DE, col=author.col[authorDE], pch=author.pch[authorDE])
legend("topleft", inset=.02, legend=levels(authorDE),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
```

```
md2.DE <- mean.deviation(zDE, n=5000, p=2)
plot(0, 0, type="n", xlim=c(0,1.2), xaxs="i", ylim=c(.2,1.0), yaxs="i",
     xlab="mean positive score", ylab="mean negative score", main="German (L2, 5000 mfw)")
md.grid(p=2)
points(md2.DE, col=author.col[authorDE], pch=author.pch[authorDE])
legend("topleft", inset=.02, legend=levels(authorDE),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
```

And some plots for the English data. We use a helper function to generate a sufficient number of plots for a small animation.

```
mean.dev.plot.EN <- function (n, p=2) {
  md2.EN <- mean.deviation(zEN, n=n, p=p)
  authorEN <- as.factor(goldEN)
  xlim <- if (p==2) c(0,1.2) else c(-0.2,0.8)
  ylim <- if (p==2) c(0.2,1) else c(0,0.8)
  plot(0, 0, type="n", xlim=xlim, xaxs="i", ylim=ylim, yaxs="i",
       xlab="mean positive score", ylab="mean negative score",
```

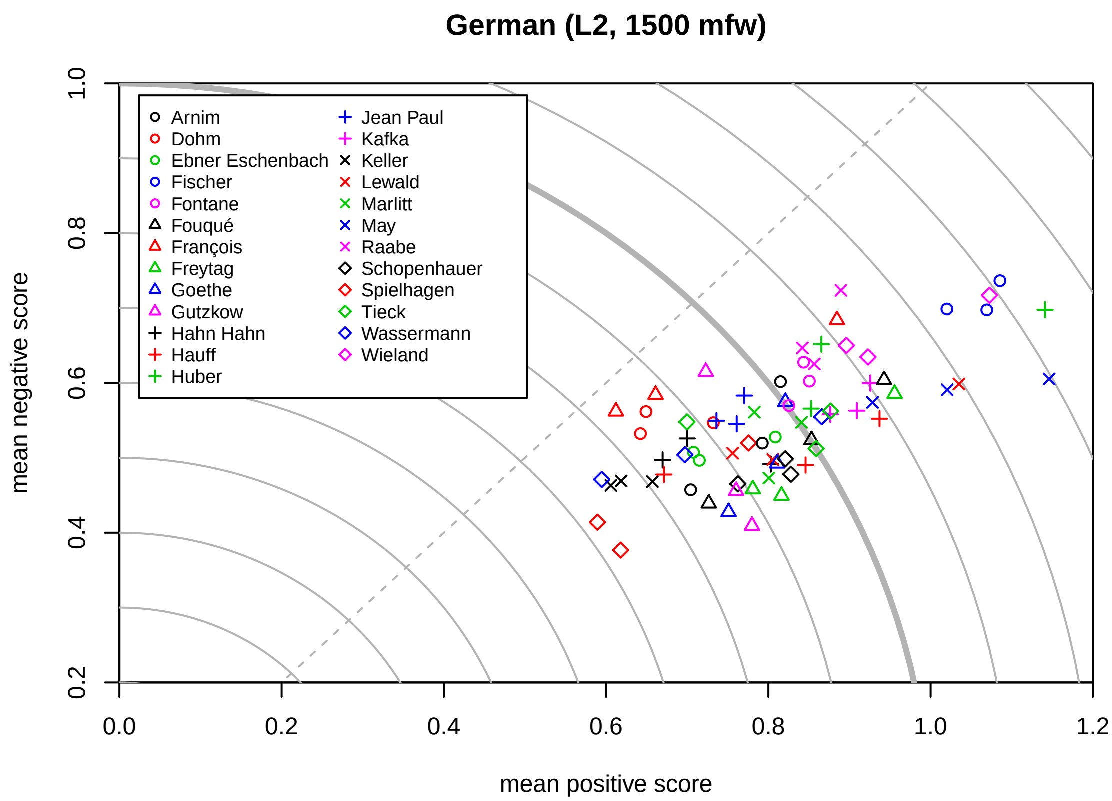


Figure 10:

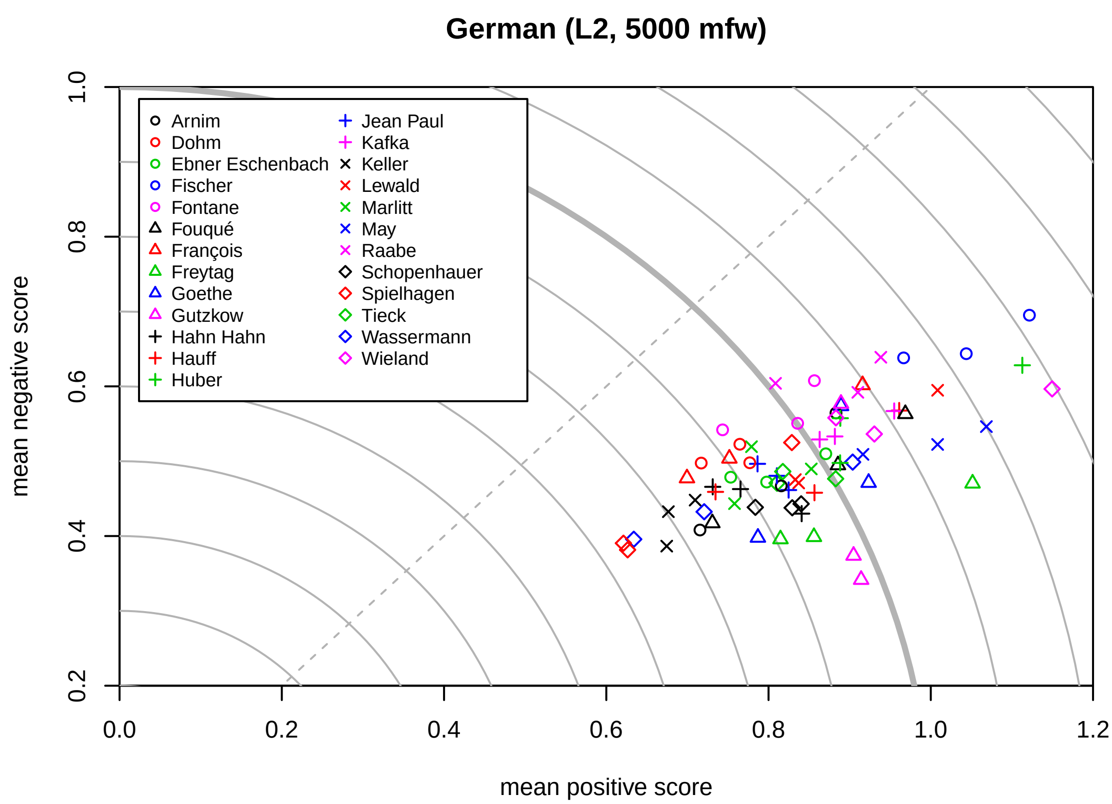


Figure 11:

```
       main=sprintf("English (L%g, %d mfw)", p, n))
md.grid(p=p)
points(md2.EN, col=author.col[authorEN], pch=author.pch[authorEN])
legend("topleft", inset=.02, legend=levels(authorEN),
       col=author.col, pch=author.pch, cex=0.75, ncol=2, bg="white")
}
```

```
mean.dev.plot.EN(500, 2)
```

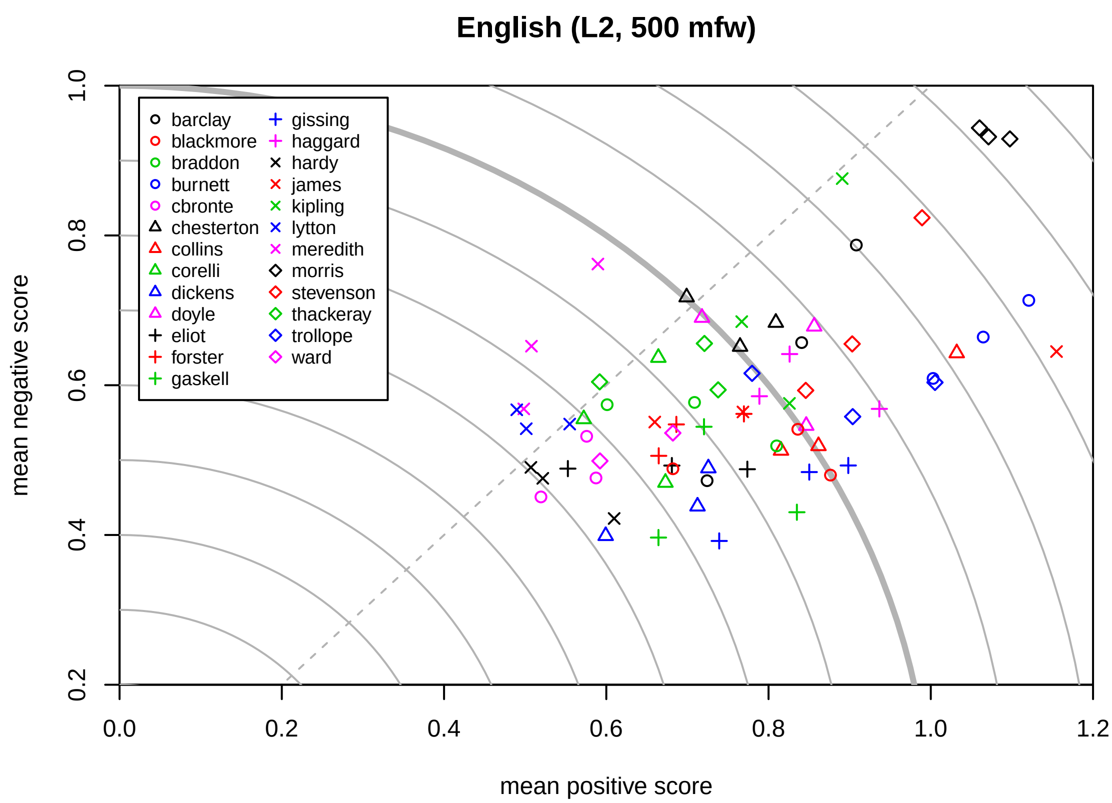


Figure 12:

```
mean.dev.plot.EN(600, 2)
```

```
mean.dev.plot.EN(700, 2)
```

```
mean.dev.plot.EN(800, 2)
```

```
mean.dev.plot.EN(1000, 2)
```

```
mean.dev.plot.EN(1250, 2)
```

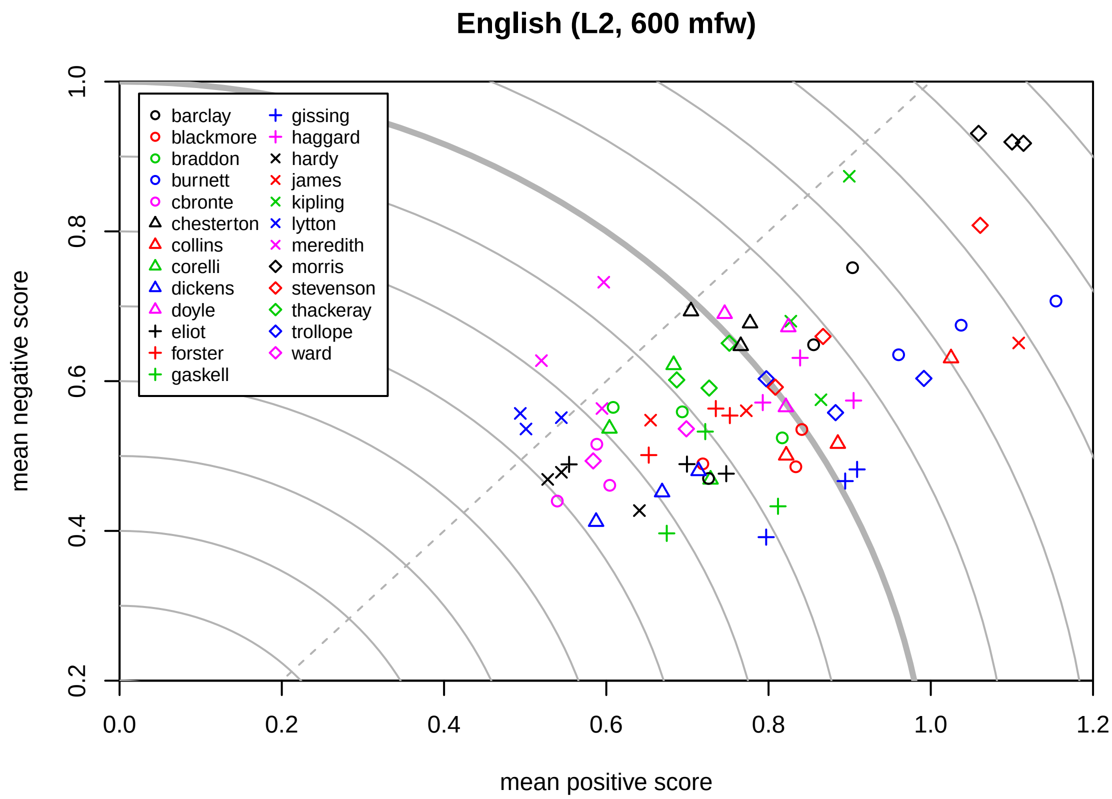


Figure 13:

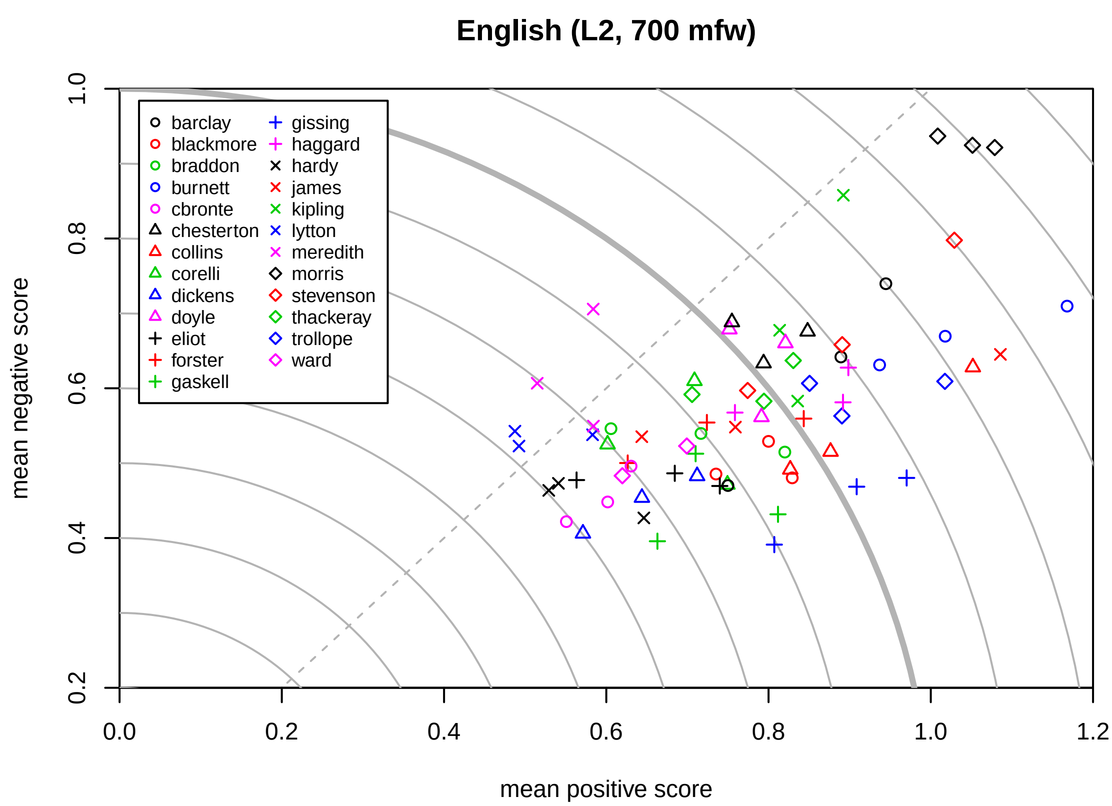


Figure 14:

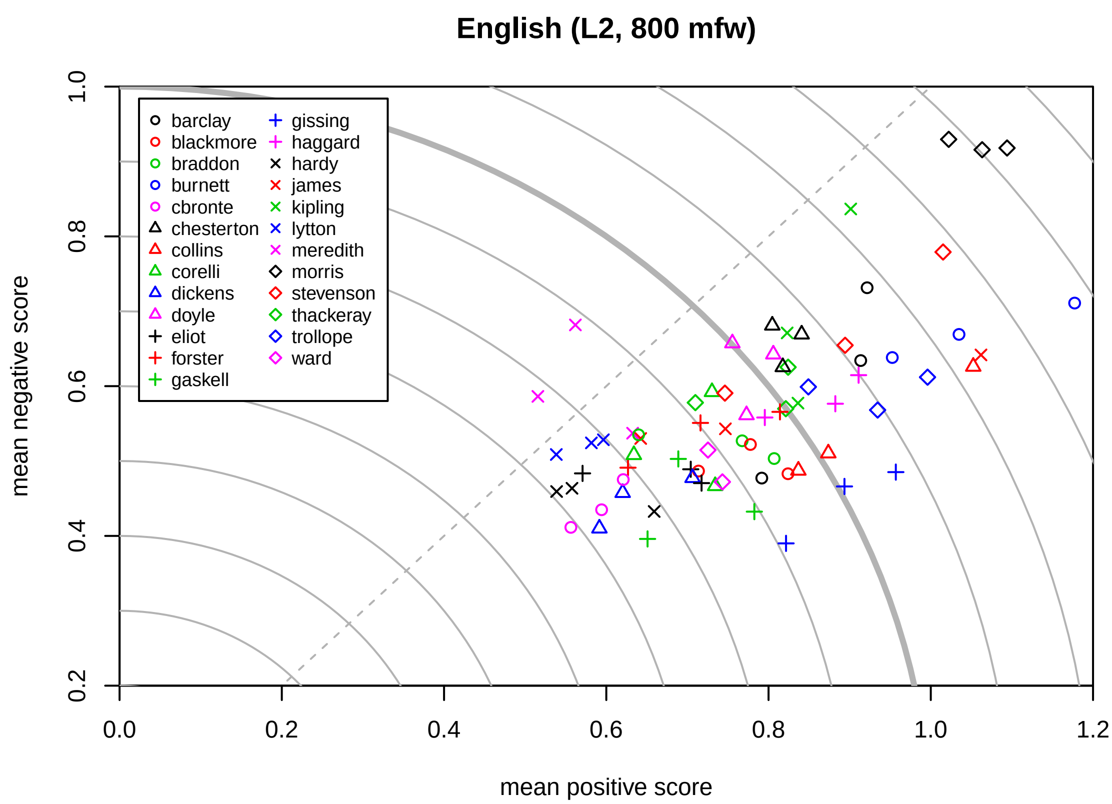


Figure 15:

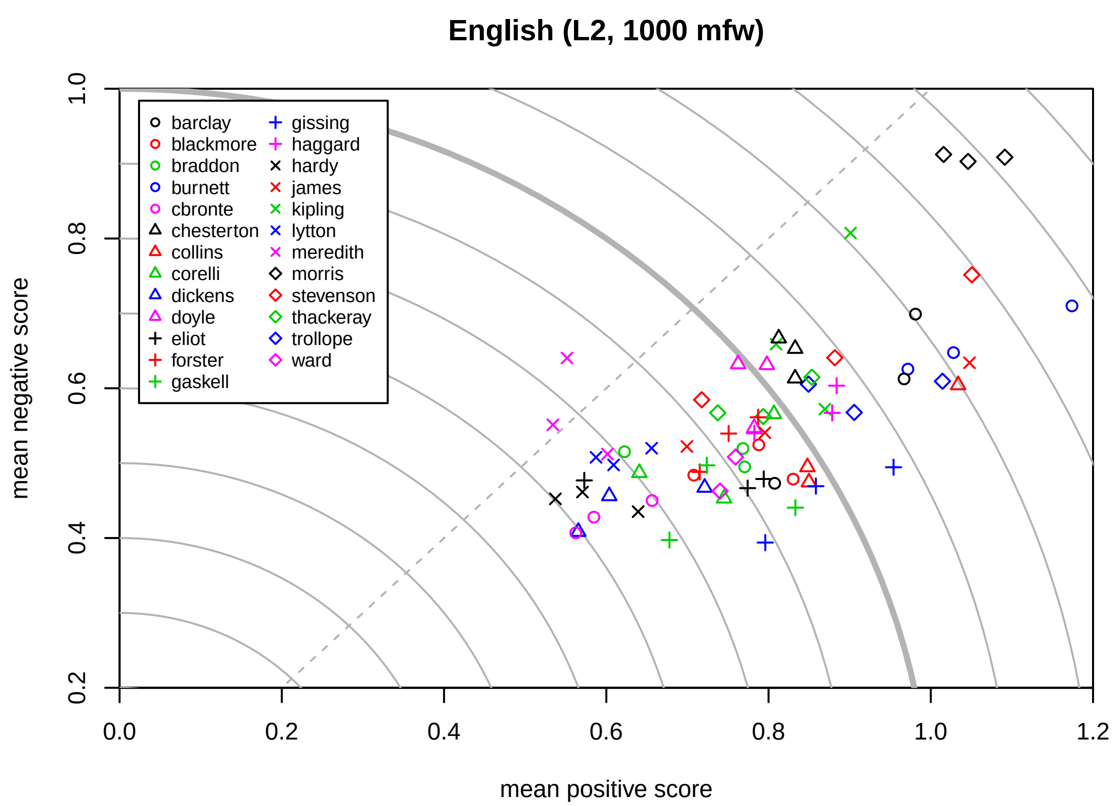


Figure 16:

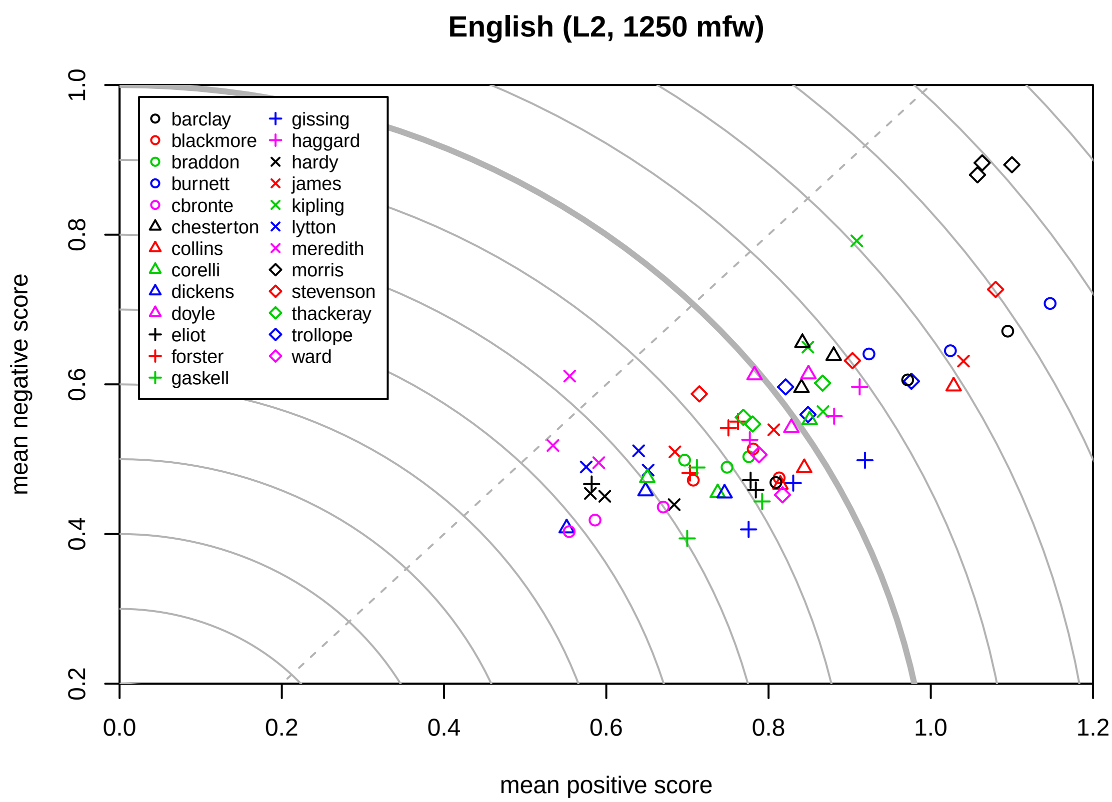


Figure 17:

```r
mean.dev.plot.EN(1500, 2)
```

```r
mean.dev.plot.EN(2000, 2)
```

```r
mean.dev.plot.EN(500, 1)
```

```r
mean.dev.plot.EN(600, 1)
```

```r
mean.dev.plot.EN(700, 1)
```

```r
mean.dev.plot.EN(800, 1)
```

```r
mean.dev.plot.EN(1000, 1)
```

```r
mean.dev.plot.EN(1250, 1)
```

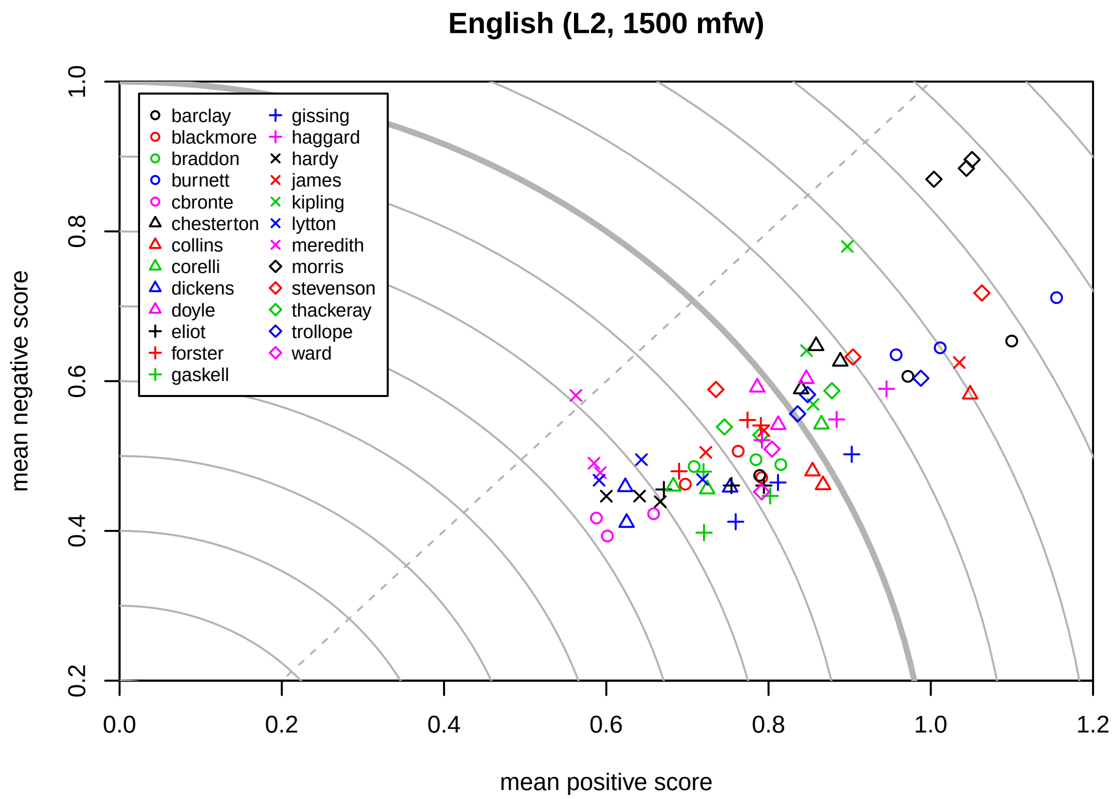


Figure 18:

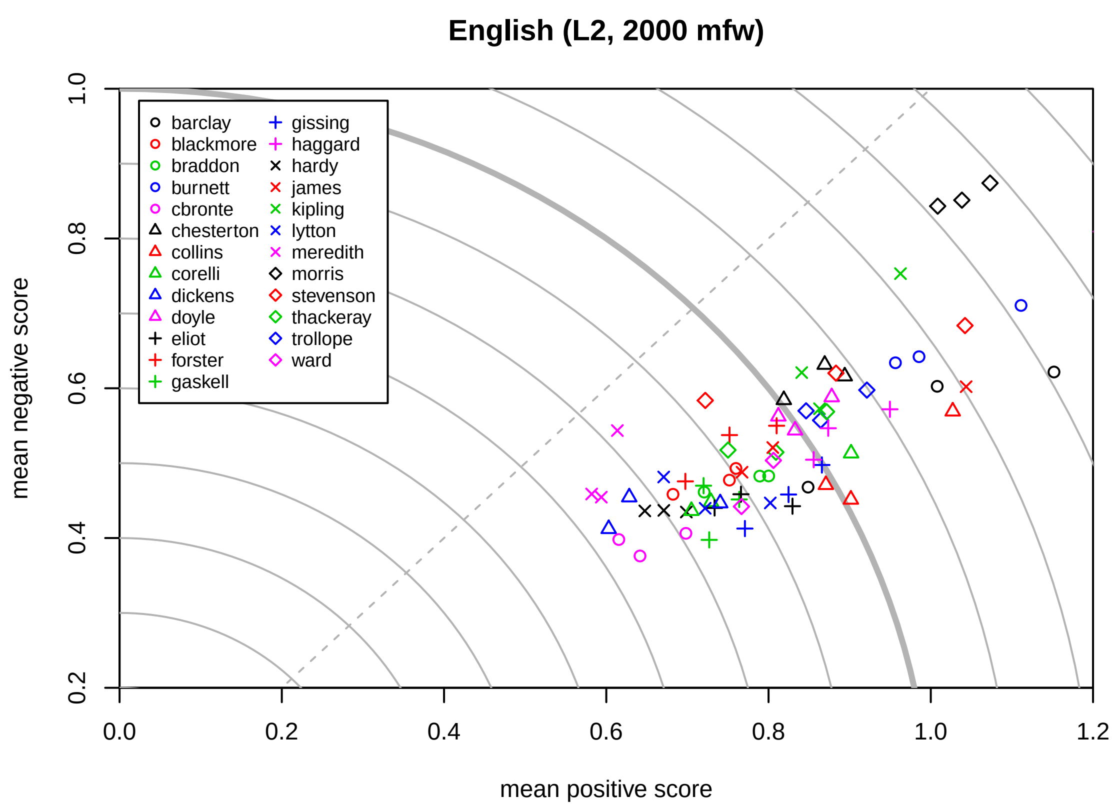


Figure 19:

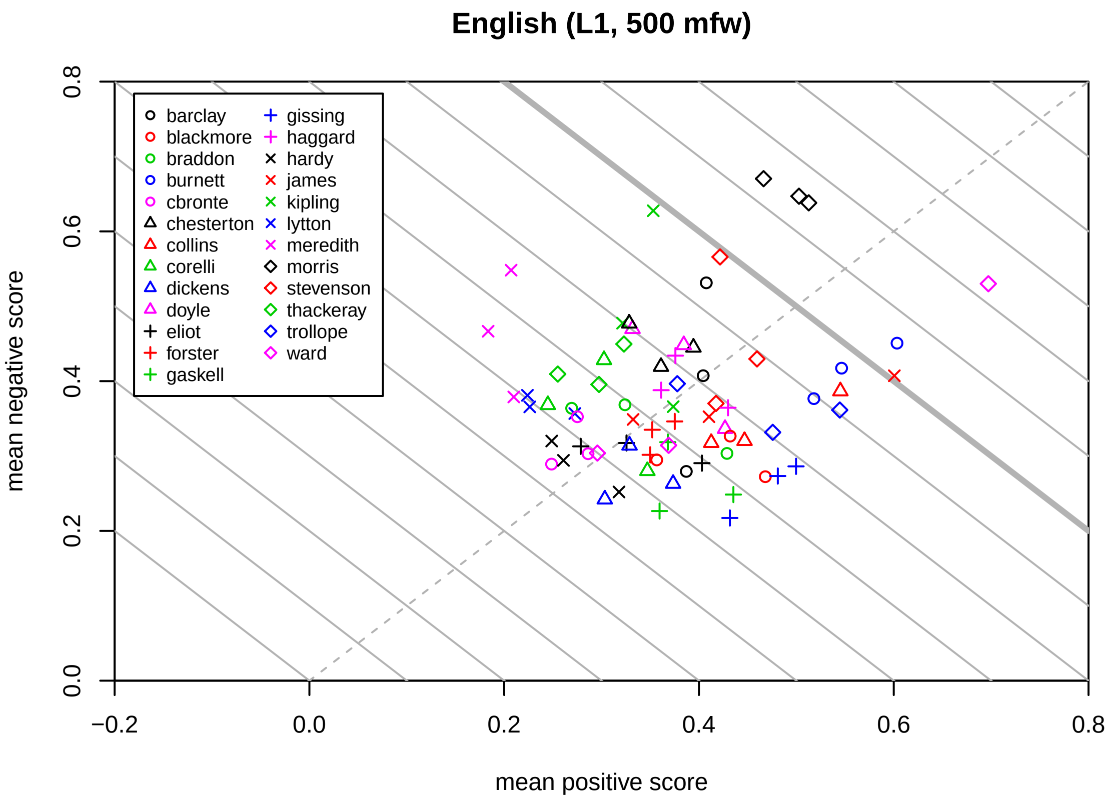


Figure 20:

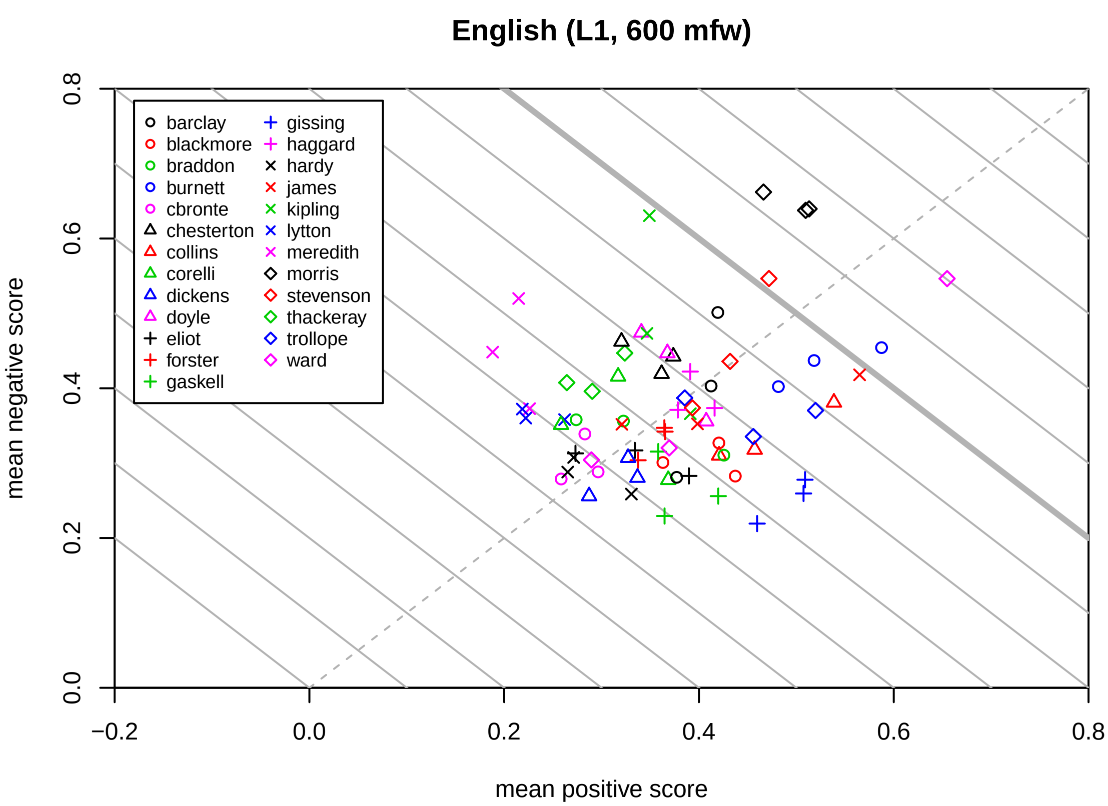


Figure 21:

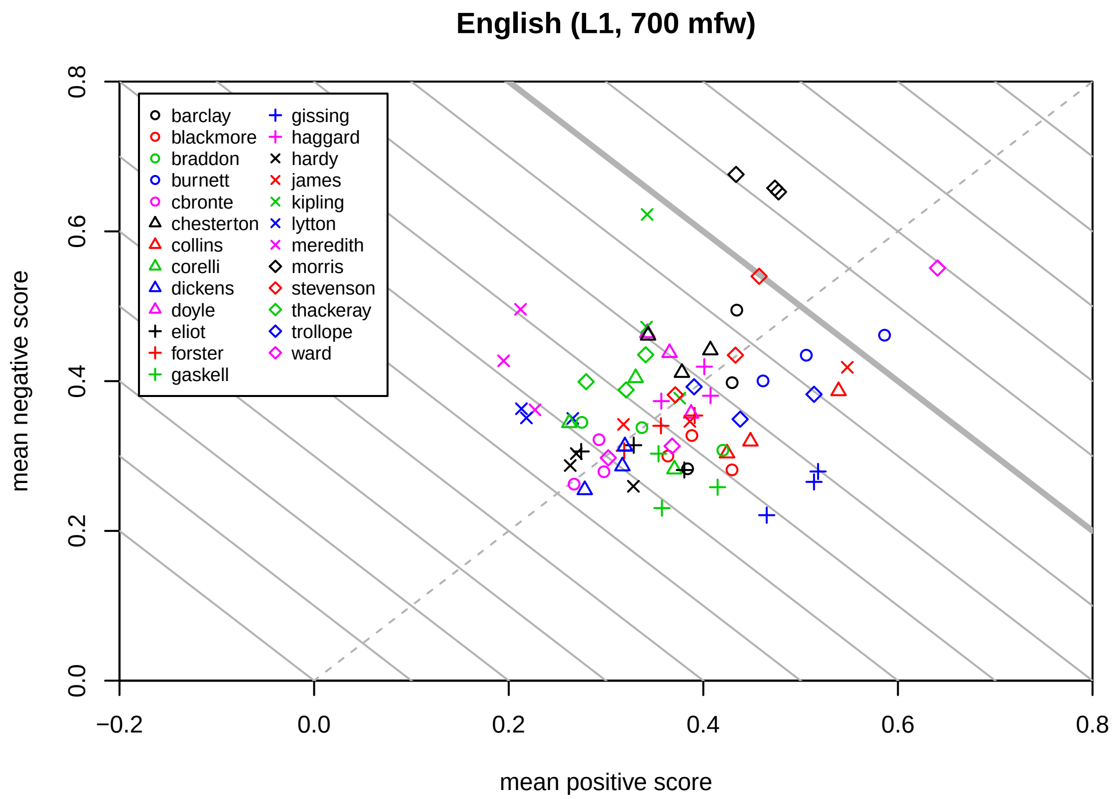


Figure 22:

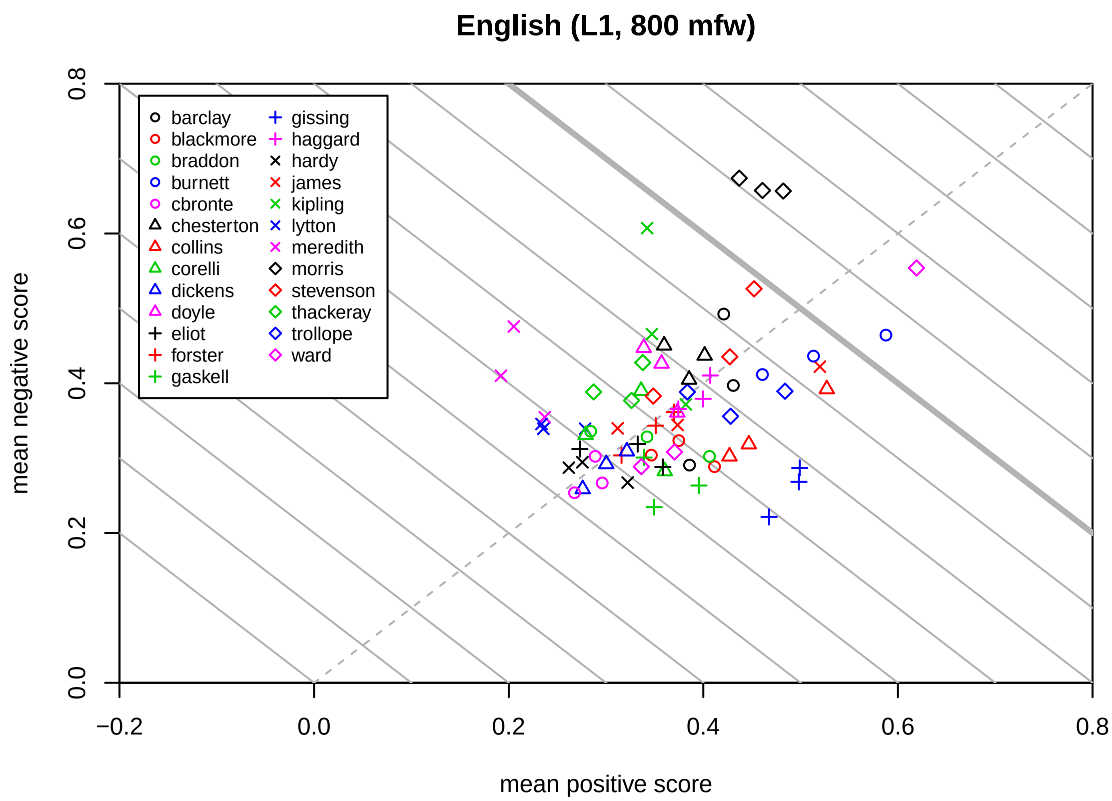


Figure 23:

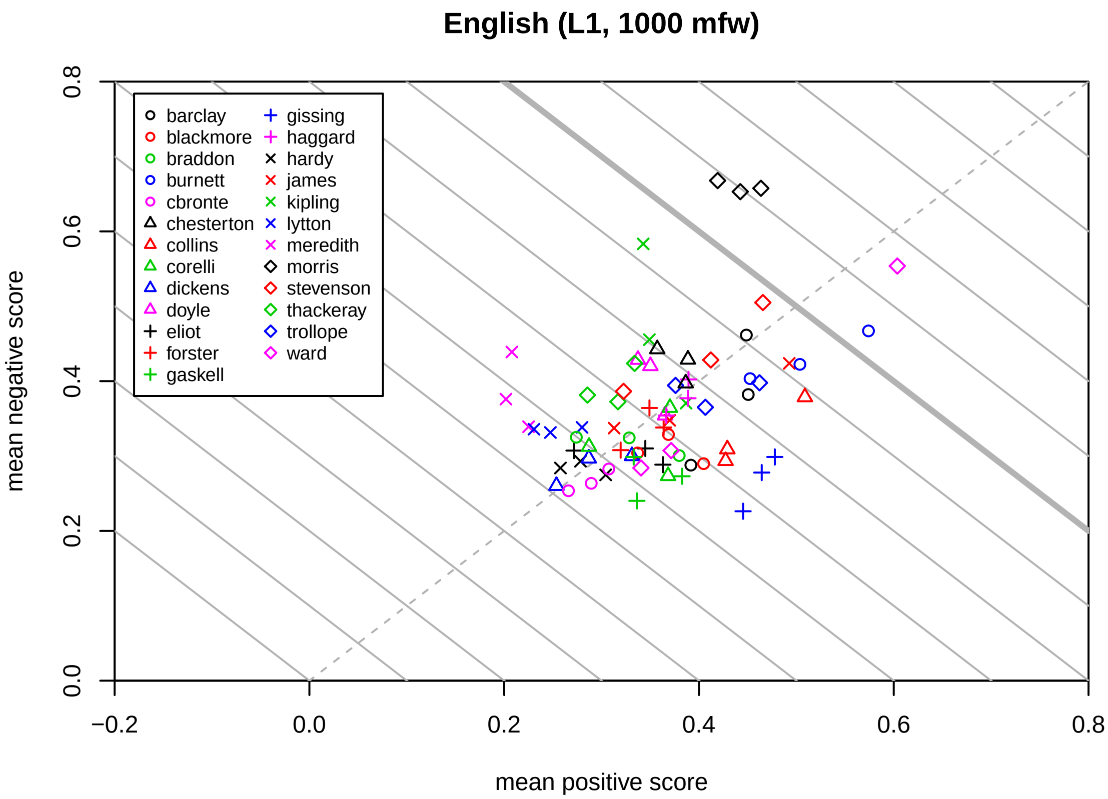


Figure 24:

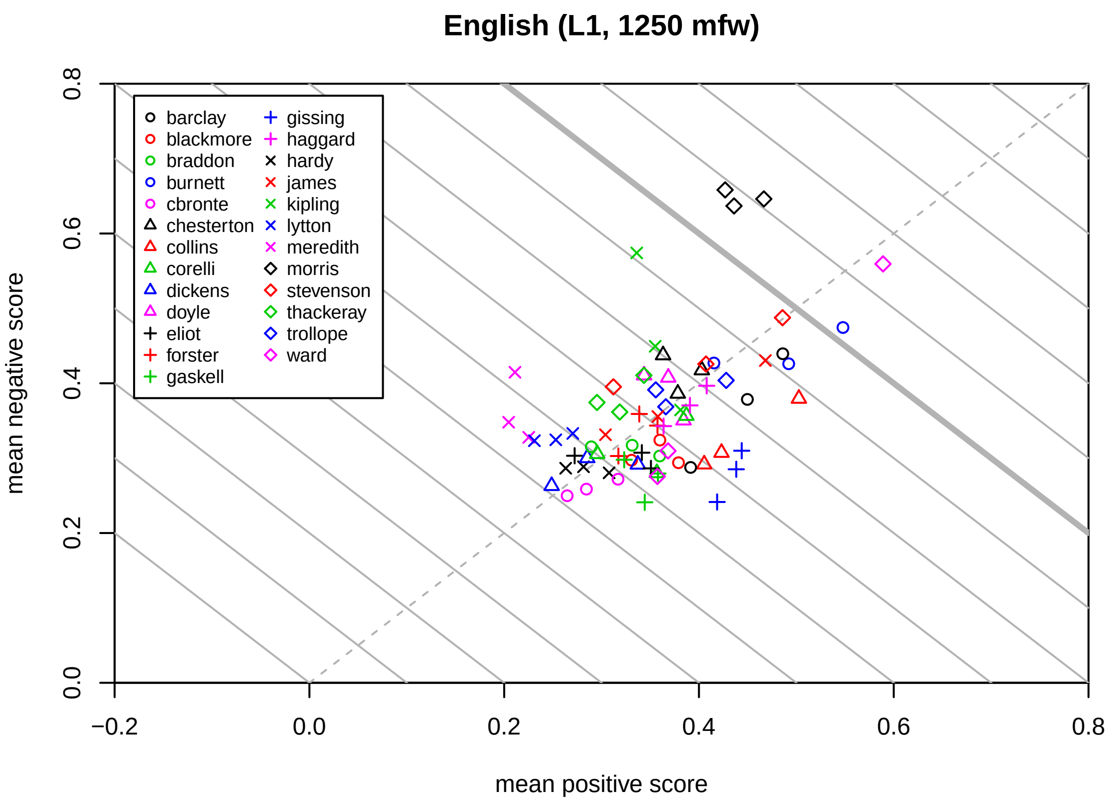


Figure 25:

```r
mean.dev.plot.EN(1500, 1)
```

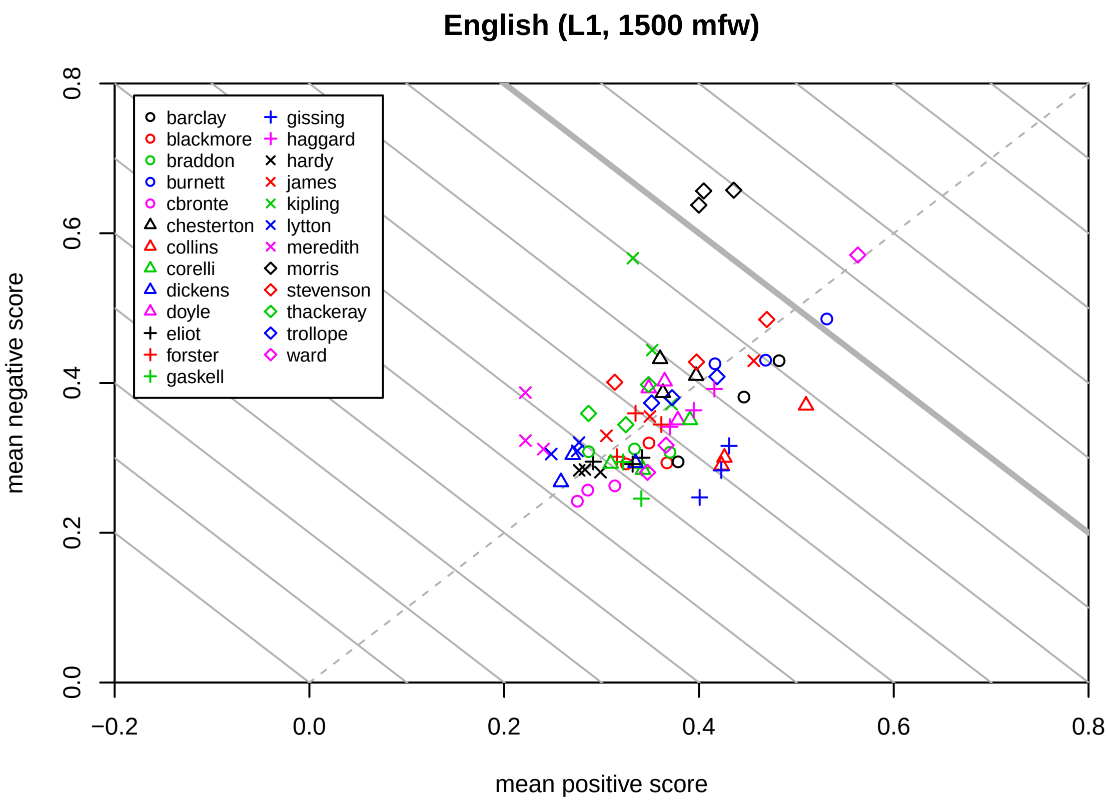


Figure 26:

```r
mean.dev.plot.EN(2000, 1)
```

TODO: perhaps vector length can be interpreted as deviation from the "norm", i.e. the mean frequencies in the collection (which corresponds to $z_i = 0$); long vectors thus indicate more idiosyncratic style; normalization removes these differences and focuses on the *profile* of an author's style, i.e. the pattern of more and less frequent words, rather than the absolute deviation

TODO: compute average positive and negative deviation (L_1 or L_2) for each vector, then scatterplot with colour / symbol indicating author

References

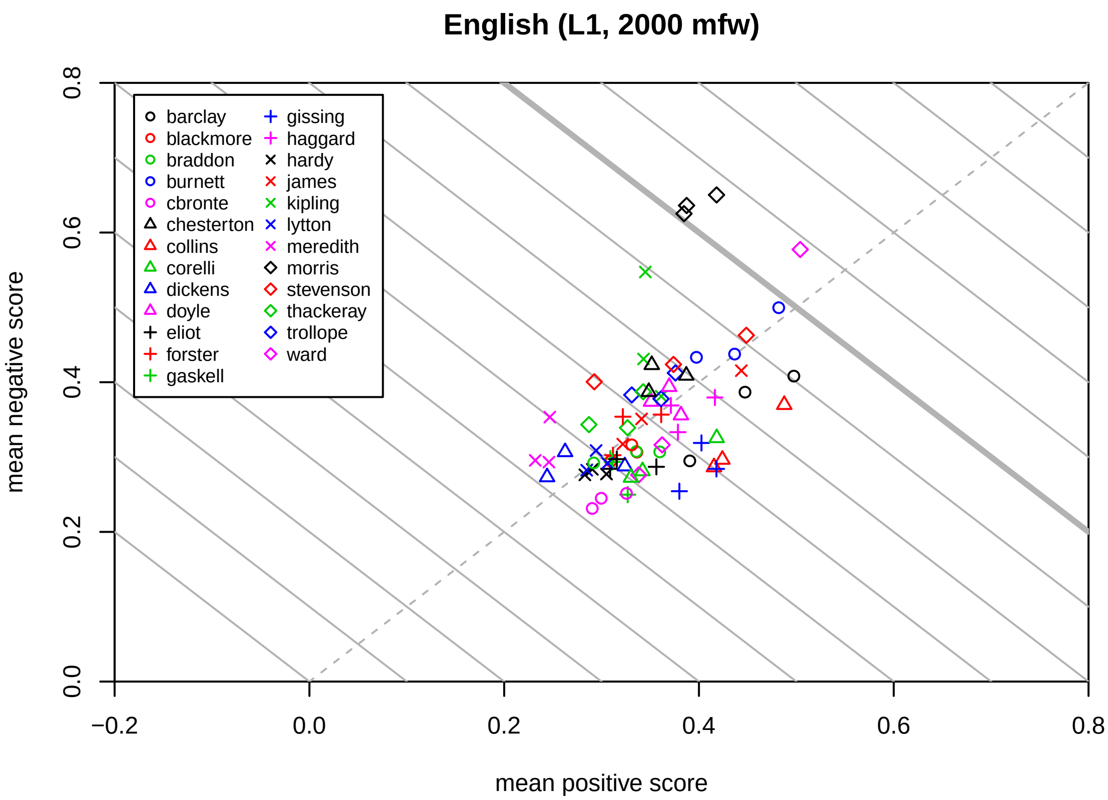

English (L1, 2000 mfw)
barclay
blackmore
braddon
burnett
cbronte
chesterton
collins
corelli
dickens
doyle
eliot
forster
gaskell
gissing
haggard
hardy
james
kipling
lytton
meredith
morris
stevenson
thackeray
trollope
ward
mean negative score
mean positive score
0.0
0.2
0.4
0.6
0.8
−0.2
0.0
0.2
0.4
0.6
0.8

Figure 27: