Appendix

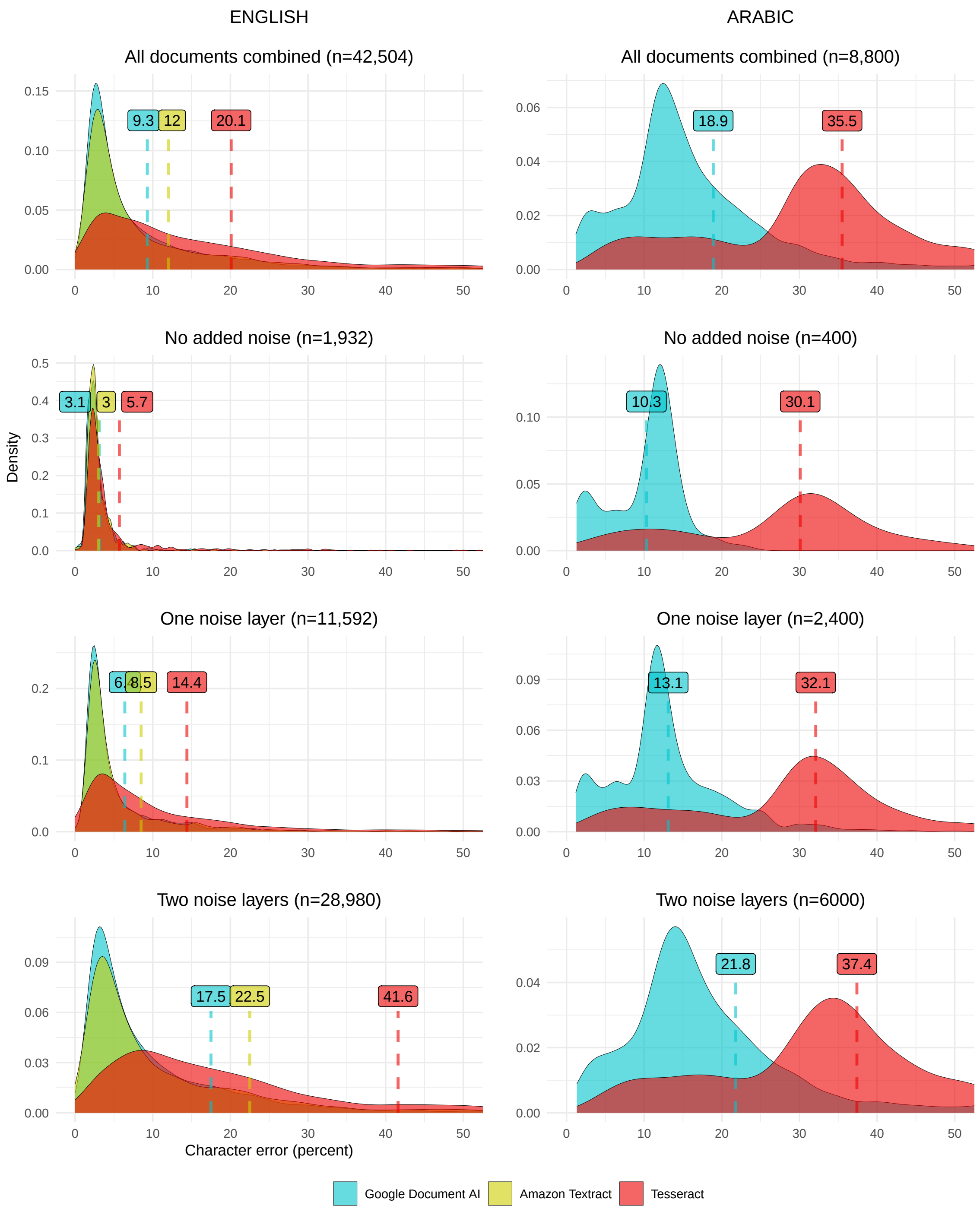
To the manuscript 'OCR with Tesseract, Amazon Textract, and Google Document AI: A Benchmarking Experiment'

Contents

1 Character accuracy results

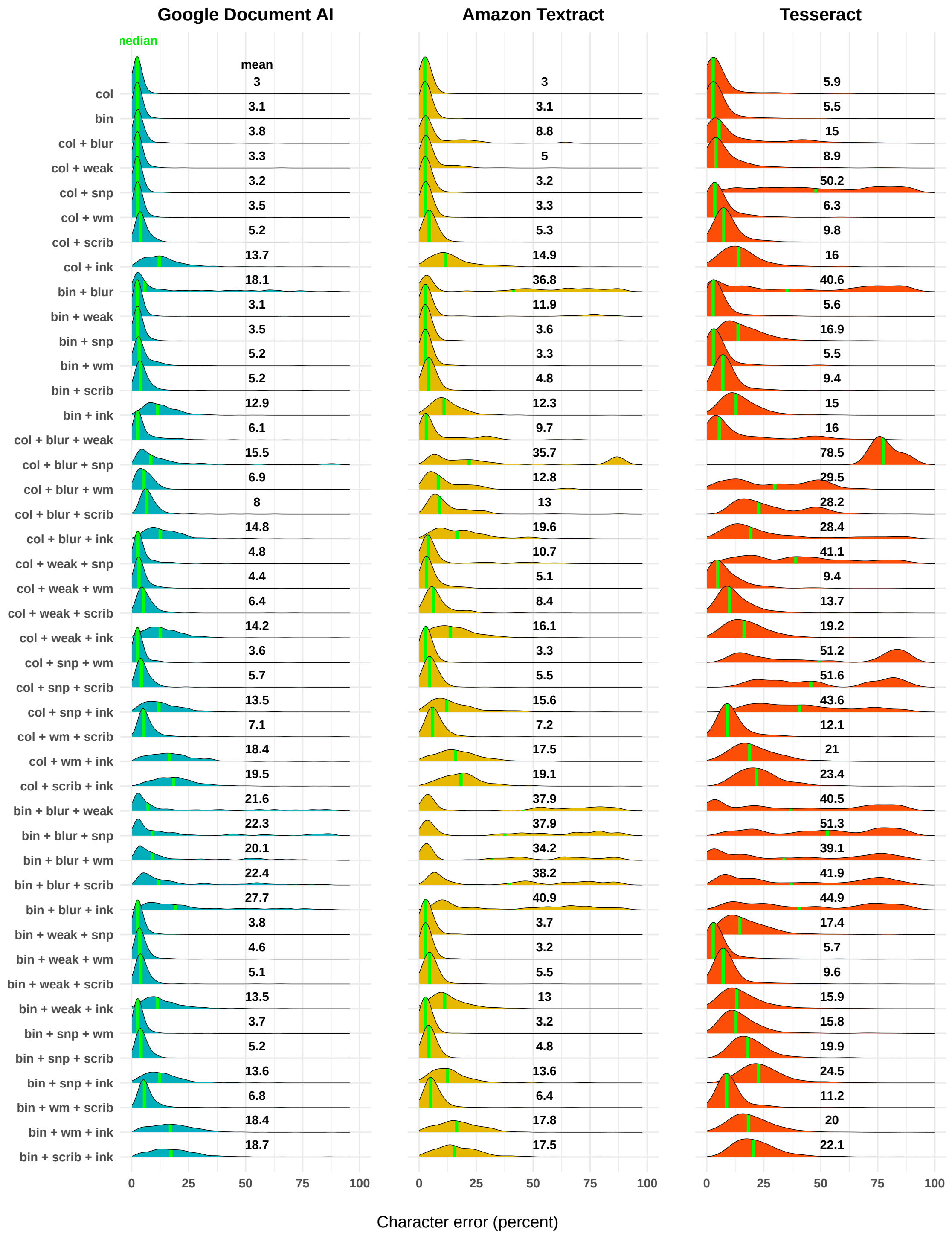
1.1 Character error rates by engine and noise level for English and Arabic documents

Mean error rates in coloured boxes. X axes cropped for visibility, leaving out the tails of the distributions.



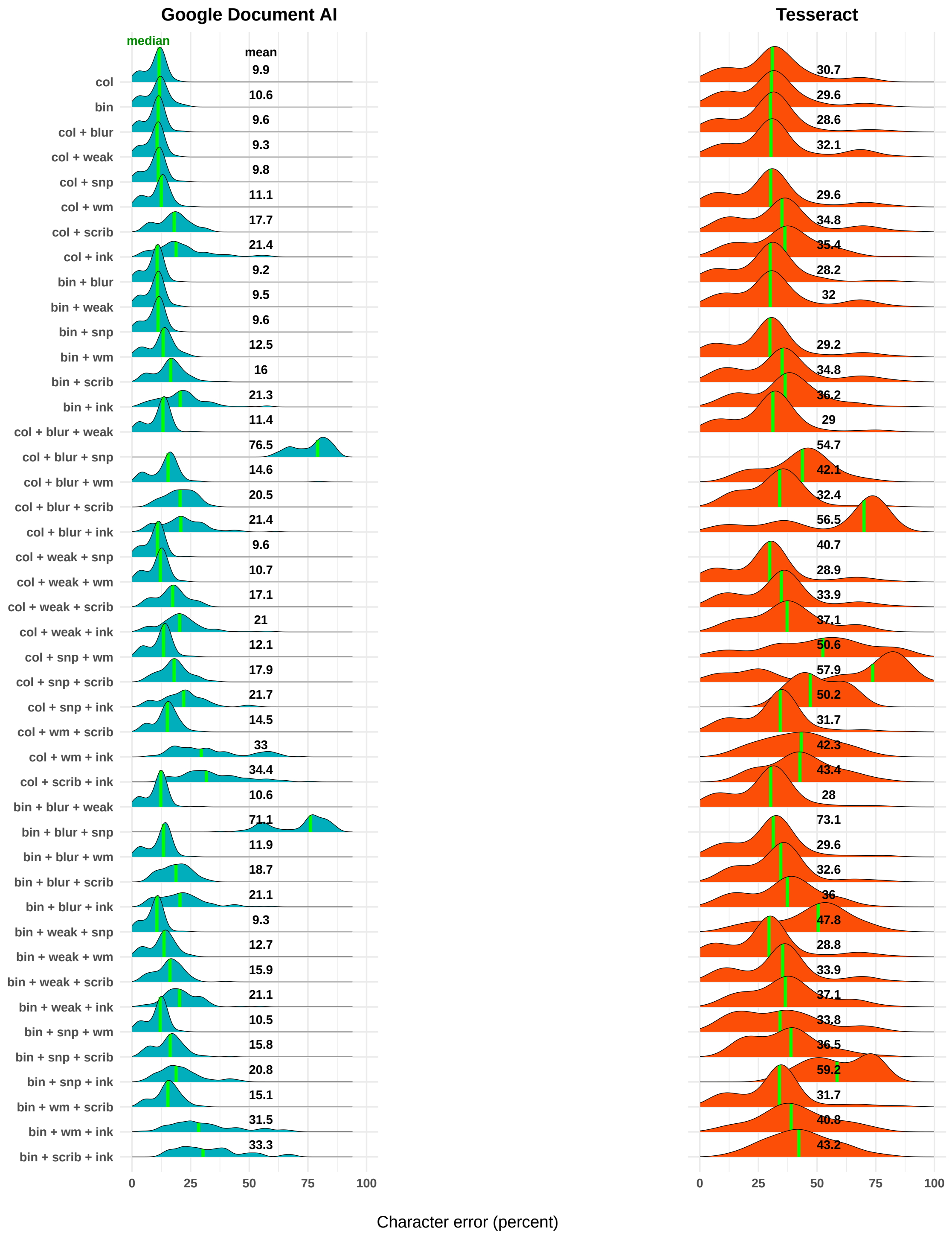
1.2 Character error rates by engine and noise type for English-language documents

Data: Single−column text in historical book scans with noise added articifially (n=42,504; 322 per engine and noise type).
Noise codes: 'col'='colour, 'bin'=binary, 'blur'=blur, 'weak'=weak ink, 'snp'=salt&pepper, 'wm'=watermark, 'scrib'=scribbles, 'ink'=ink stains.



1.3 Character error rates by engine and noise type for Arabic-language documents

Data: Single−column text in image scans of Arabic Wikipedia pages with noise added articifially (n = 8800; 100 per engine and noise type).
Noise codes: 'col'=colour, 'bin'=binary, 'blur'=blur, 'weak'=weak ink, 'snp'=salt&pepper, 'wm'=watermark, 'scrib'=scribbles, 'ink'=ink stains.



2 R code for noise generation

```r
### Prerequisites ###########
#
# a) Install the following R packages: dplyr, glue, magick, yarr, showtext, shape
#
# b) For functions 4-6, set the image dimensions (in pixels) as follows:
# img_width <- #<An integer>
# img_height <- #<An integer>
#
# c) For function 5, install the proprietary font "Shopping Script" (https://www.dafont.com/shopping-sc
#
#############################


### Noise-generating functions

# All functions take two inputs:
# 1) source_ims: a vector of filepaths for the source images
# 2) dest_folder: a folder path for destination directory

# 1. BLUR
blur <- function(source_ims, dest_folder) {
  for (i in source_ims){
    magick::image_read(i) %>%
    magick::image_blur(5, 4) %>%
    magick::image_write(glue::glue("{dest_folder}/{basename(i)}"))
  }
}

# 2. WEAK INK
weaken <- function(source_ims, dest_folder) {
  for (i in source_ims){
    magick::image_read(i) %>%
    magick::image_oilpaint() %>%
    magick::image_write(glue::glue("{dest_folder}/{basename(i)}"))
  }
}

# 3. SALT & PEPPER
snp <- function(source_ims, dest_folder) {
  for (i in source_ims){
    magick::image_read(i) %>%
    magick::image_noise(noisetype = "poisson") %>%
    magick::image_write(glue::glue("{dest_folder}/{basename(i)}"))
  }
}

# 4. WATERMARK
watermark <- function(source_ims, dest_folder) {
  transp_grey <- yarrr::transparent(orig.col = "gray80",
                                    trans.val = 0.4,
                                    maxColorValue = 255)
  for (i in source_ims){
    img <- magick::image_read(i)
    tiff(glue::glue("{dest_folder}/{basename(i)}"),
         width=img_width,
         height=img_height,
         units="px",
         res=300)
    plot(img)
    text(1200, 2000, # NB adapt coordinates to image dimensions
         "Watermark",
         cex=20,
         srt=50,
         col=transp_grey)
    dev.off()
  }
}

# 5. SCRIBBLES
scribble <- function(source_ims, dest_folder) {
  showtext::font_add(family = "Shopping Script",
                     regular = "~/.fonts/ShoppingScript-Regular.otf")
  showtext::showtext_auto()
  for (i in source_ims){
    img <- magick::image_read(i)
    tiff(glue::glue("{dest_folder}/{basename(i)}"),
         width=img_width,
         height=img_height,
         units="px",
         res=300)
    plot(img)
    # NB adapt coordinates below to image dimensions
    text(1500, 2000, "fascinating", family = "Shopping Script", cex = 10, srt = 15, col = "gray30")
    text(600, 1000, "__", family = "Shopping Script", cex = 12, col = "gray10", srt = 4)
    text(300, 700, "_ _", family = "Shopping Script", cex = 12, col = "gray20", srt = -4)
    text(1600, 800, "NB!", family = "Shopping Script", cex = 10, srt = -10, col = "gray30")
    text(1650, 1400, "V", family = "Shopping Script", cex =8, srt = 10, col = "gray35")
    text(1700, 500, "V", family = "Shopping Script", cex =12, srt = 30, col = "gray25")
    text(400, 1500, "mmm", family = "Shopping Script", cex=10)
    text(1100, 500, "mnnm", family = "Shopping Script", cex=10, col = "gray35")
    text(1000, 1200, "mmmmmmm", family = "Shopping Script", cex=10, col = "gray25")
    text(50, 1300, "|", family = "Shopping Script", cex=10)
    text(100, 1100, "Z", family = "Shopping Script", cex=10, col = "gray25")
    text(400, 1800, "O", family = "Shopping Script", cex=12, col = "gray35")
    text(1200, 700, "O", family = "Shopping Script", cex=10, col = "gray25")
    text(450, 400, "X", family = "Shopping Script", cex=10, col = "gray15")
    dev.off()
  }
}

# 6. INK STAINS
ink <- function(source_ims, dest_folder) {
  for (i in source_ims){
    tiff(file="ink.tiff",
          width=img_width,
          height=img_height,
          units="px",
          res=300)
    par(bg=NA)
    shape::emptyplot()
    shape::filledshape(matrix(nc = 4,
                              nr = 4,
                              runif(8)),
                       col = shadepalette(50, "black", "grey20"))
    dev.off()
    main <- magick::image_read(i)
    inset <- magick::image_read("ink.tiff")
    final <- magick::image_composite(main, inset, operator = "atop")
    image_write(final, glue::glue("{dest_folder}/{basename(i)}"))
  }
}
```