Performance visualisations

Anita Kurm

5/21/2020

Set-up, data import

```
pacman::p_load(tidyverse, rjson, extrafont)
font_import(prompt = FALSE, pattern = "Raleway") #you need to download the font for that
```

```
## Scanning ttf files in /Library/Fonts/, /System/Library/Fonts, ~/Library/Fonts/ ...
```

```
## Extracting .afm files from .ttf files...
```

```
## /Users/anitakurm/Library/Fonts/Raleway-Regular.ttf : Raleway-Regular already registered in fonts database
. Skipping.
## /Users/anitakurm/Library/Fonts/Raleway-Thin.ttf : Raleway-Thin already registered in fonts database. Skip
ping.
## Found FontName for 0 fonts.
## Scanning afm files in /Library/Frameworks/R.framework/Versions/3.6/Resources/library/extrafontdb/metrics
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FontName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FontName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FontName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FontName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FontName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FontName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
## Warning in grepl("^FamilyName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FontName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^FullName", text): input string 4 is invalid in this
## locale
```

```
## Warning in grepl("^Weight", text): input string 4 is invalid in this locale
```

```
df <- read_csv("twitterQA_berts.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Question = col_character(),
##   Answer = col_character(),
##   Tweet = col_character(),
##   qid = col_character(),
##   L_BERT_answer = col_character(),
##   L_BERT_time = col_double(),
##   DistilBERT_answer = col_character(),
##   DistilBERT_time = col_double()
## )
```

```
tok_times <- fromJSON(file = "tokenizer_loading.txt")
mod_times <- fromJSON(file = "model_loading.txt")

# Define color palette
cp <- c("aquamarine3", "grey19")

# See available fonts
#fonts()
```

Initial dataset stats

```
init <- read_csv("initial11778.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Answer = col_character(),
##   Gold_1 = col_character(),
##   Gold_2 = col_character(),
##   Question = col_character(),
##   Tweet = col_character(),
##   qid = col_character()
## )
```

```
init %>%
  mutate(Gold1_length = sapply(strsplit(Gold_1, " "), length),
         Gold2_length = ifelse(is.na(Gold_2), 0, sapply(strsplit(Gold_2, " "), length)),
         Gold_max_length = ifelse(Gold1_length>Gold2_length, Gold1_length, Gold2_length),
         Question_length = sapply(strsplit(Question, " "), length)) %>%
  summarise(n(), mean(Gold_max_length), mean(Question_length))
```

```
## # A tibble: 1 x 3
##   `n()` `mean(Gold_max_length)` `mean(Question_length)`
##   <int>                   <dbl>                   <dbl>
## 1 11778                    2.50                    6.97
```

Data pre-processing

```
# Reshape data by creating two separate dfs and binding together
l_bert <- df %>%
  select(qid,
         X1,
         'Answer_pred' = L_BERT_answer,
         'Time' = L_BERT_time) %>%
  mutate(A_len = str_length(Answer_pred),
         Model = "BERT",
         Tok_time = tok_times$L_BERT,
         Load_time = mod_times$L_BERT)

d_bert <- df %>%
  select(qid,
         X1,
         'Answer_pred' = DistilBERT_answer,
         'Time' = DistilBERT_time) %>%
  mutate(A_len = str_length(Answer_pred),
         Model = "DistilBERT",
         Tok_time = tok_times$DistilBERT,
         Load_time = mod_times$DistilBERT)

data <- rbind(l_bert, d_bert)
```

Get a dataframe with both answers present:

```
df_present <- df %>%
  filter(!is.na(L_BERT_answer) & !is.na(DistilBERT_answer))
#write_csv(df_present, "tweetQA_bothpresent.csv")

d_present <- data %>%
  filter(!is.na(data$Answer_pred))
```

Evaluate data loss and processing time

Processing time summary (full dataset):

```
# Summarise
time_summary <- data %>%
  mutate(Tok_time = as.numeric(Tok_time),
         Load_time = as.numeric(Load_time)) %>%
  group_by(Model) %>%
  summarise(Missing = sum(is.na(Answer_pred)),
            Answered = sum(!is.na(Answer_pred)),
            Mean_time = mean(Time),
            'Max time' = max(Time),
            'Min time' = min(Time),
            'Total time' = sum(Time),
            'Tokenizer loading time' = max(Tok_time),
            'Model loading time' = max(Load_time),
            'Total time with loading' = sum(Time) + max(Tok_time)+ max(Load_time)) %>%
  mutate_if(is.numeric, round, 3)

time_summary
```

```
## # A tibble: 2 x 10
##   Model Missing Answered Mean_time `Max time` `Min time` `Total time`
##   <chr>   <dbl>    <dbl>     <dbl>      <dbl>      <dbl>        <dbl>
## 1 BERT      303    11475     0.429     115.        0.182        5047.
## 2 Dist…     446    11332     0.08        6.06      0.027         943.
## # … with 3 more variables: `Tokenizer loading time` <dbl>, `Model loading
## #   time` <dbl>, `Total time with loading` <dbl>
```

Processing time summary (all present dataset):

```
# Summarise
time_summary <- d_present %>%
  filter(Time<10) %>%
  mutate(Tok_time = as.numeric(Tok_time),
         Load_time = as.numeric(Load_time)) %>%
  group_by(Model) %>%
  summarise(Missing = sum(is.na(Answer_pred)),
            Answered = sum(!is.na(Answer_pred)),
            Mean_time = mean(Time),
            "sd time" = sd(Time),
            'Max time' = max(Time),
            'Min time' = min(Time),
            'Total time' = sum(Time),
            'Tokenizer loading time' = max(Tok_time),
            'Model loading time' = max(Load_time),
            'Total time with loading' = sum(Time) + max(Tok_time)+ max(Load_time)) %>%
  mutate_if(is.numeric, round, 3)

time_summary
```

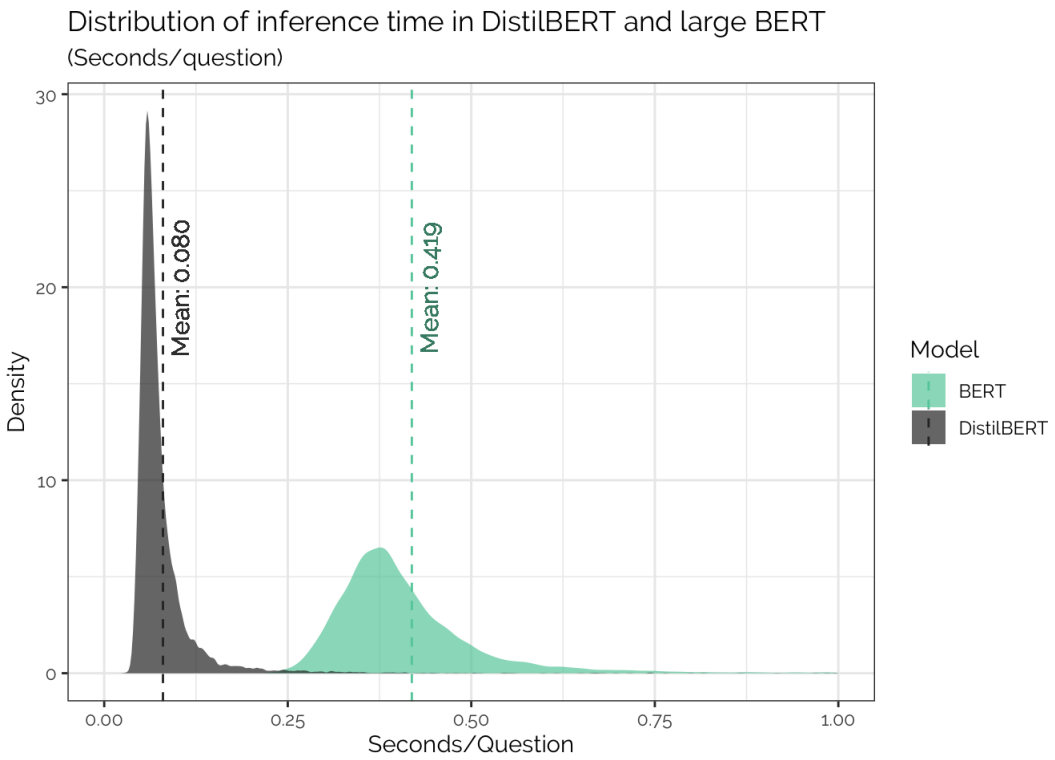
```
## # A tibble: 2 x 11
##   Model Missing Answered Mean_time `sd time` `Max time` `Min time`
##   <chr>   <dbl>    <dbl>     <dbl>     <dbl>      <dbl>      <dbl>
## 1 BERT        0    11474     0.419     0.169       4.13      0.182
## 2 Dist…       0    11332     0.08      0.101       6.06      0.027
## # … with 4 more variables: `Total time` <dbl>, `Tokenizer loading
## #   time` <dbl>, `Model loading time` <dbl>, `Total time with
## #   loading` <dbl>
```

Visualisations

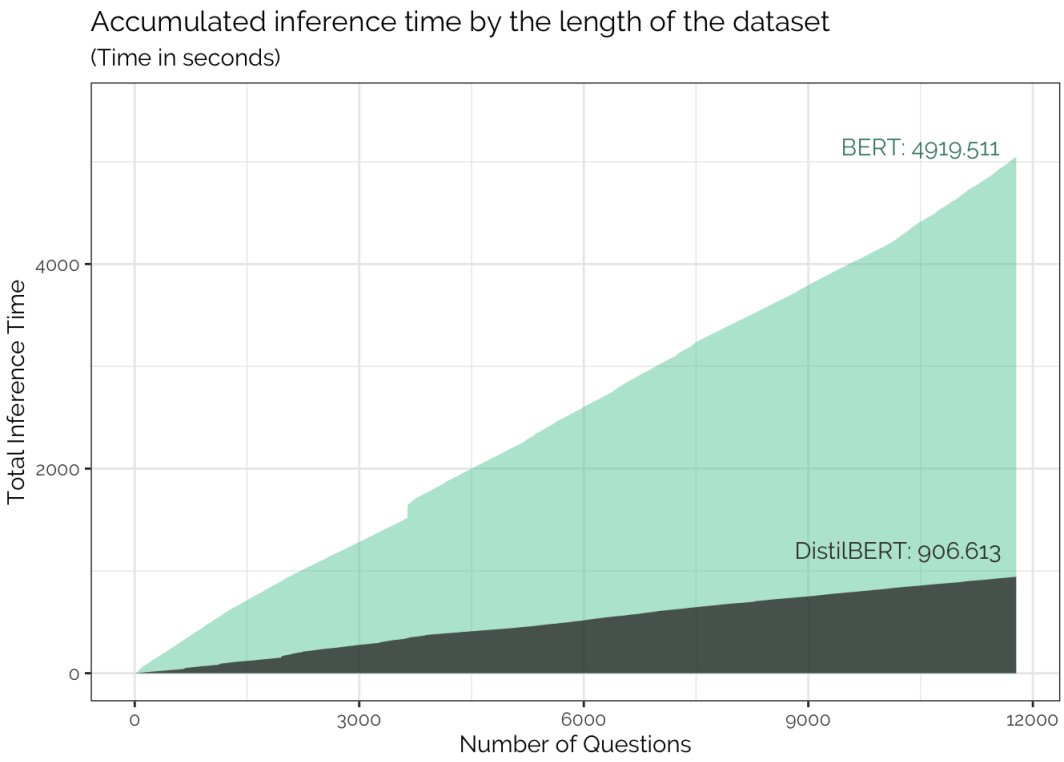
Time by model

```
d_present <- d_present %>%
  filter(Time<10)
# Density plot
density <- ggplot(d_present, aes(Time, fill = Model))+
  geom_vline(data=time_summary, aes(xintercept=Mean_time, color=Model),
             linetype="dashed")+
  geom_text(aes(x=0.080, label="\nMean: 0.080", y=20), colour="grey23", angle=90, size=4, family = "Raleway"
) +
  geom_text(aes(x=0.419, label="\nMean: 0.419", y=20), colour="aquamarine4", angle=90, size=4, family = "Ral
eway")+
  geom_density(col = NA, alpha = 0.7)+
  theme_bw()+
  scale_colour_manual(values=cp)+
  scale_fill_manual(values=cp)+
  labs( x = "Seconds/Question",
        y = "Density",
       title = "Distribution of inference time in DistilBERT and large BERT",
       subtitle = "(Seconds/question)")+
  theme(text = element_text(family ="Raleway"))+
  xlim(0,1)

density
```

```
## Warning: Removed 117 rows containing non-finite values (stat_density).
```

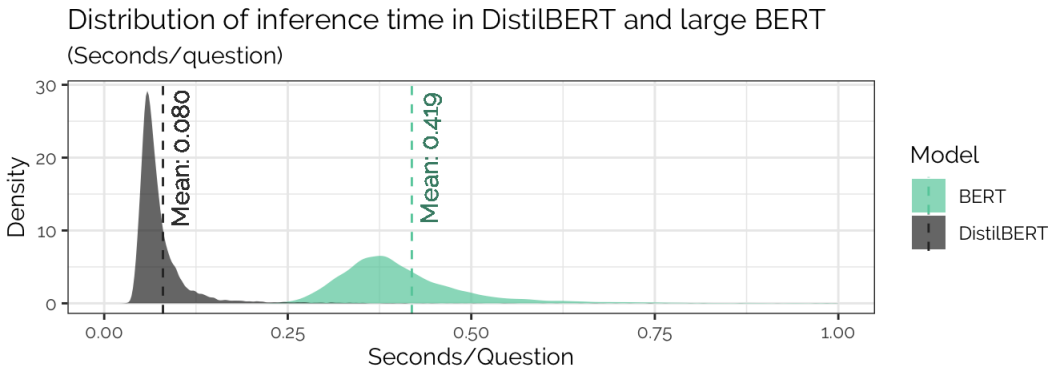


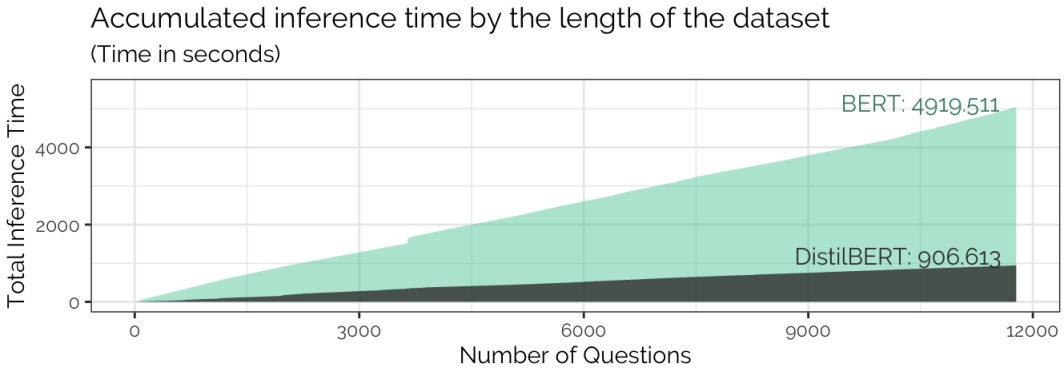

```
#Cumulative processing time plot
cumsum_l <- cumsum(l_bert$Time)
cumsum_d <- cumsum(d_bert$Time)

cumulative <-  ggplot()+
  geom_area(aes(1:length(cumsum_l), cumsum_l), fill = "aquamarine3", alpha = 0.5)+
  geom_text(aes(x=10500, label="BERT: 4919.511", y=5150), colour="aquamarine4", family = "Raleway", hjust="c
enter") +
  geom_text(aes(x=10200, label="DistilBERT: 906.613", y=1200), colour="grey23", family = "Raleway", hjust="c
enter") +
  geom_area(aes(1:length(cumsum_d), cumsum_d), fill = "grey23", alpha = 0.8)+
  theme_bw()+
  labs(x = "Number of Questions",
       y = "Total Inference Time",
       title = "Accumulated inference time by the length of the dataset",
       subtitle = "(Time in seconds)")+
  theme(text = element_text(family = "Raleway"), legend.title=element_blank())+
  ylim(0,5500)

cumulative
```




```
time_plots <- gridExtra::grid.arrange(density,cumulative, nrow=2)
```

```
## Warning: Removed 117 rows containing non-finite values (stat_density).
```






```
ggsave("timeplots.png", time_plots, width = 8, height = 6)
```

Linear models for stats

```
# Predict time by model
time_by_model <- lm(Time ~ Model, d_present)
summary(time_by_model)
```

```
##
## Call:
## lm(formula = Time ~ Model, data = d_present)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2365 -0.0354 -0.0177  0.0079  5.9836
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.418769   0.001302   321.6   <2e-16 ***
## ModelDistilBERT -0.338765   0.001847  -183.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1395 on 22804 degrees of freedom
## Multiple R-squared:  0.5959, Adjusted R-squared:  0.5958
## F-statistic: 3.362e+04 on 1 and 22804 DF,  p-value: < 2.2e-16
```

```
t.test(Time~Model, d_present)
```

```
##
##  Welch Two Sample t-test
##
## data:  Time by Model
## t = 183.91, df = 18776, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3351542 0.3423754
## sample estimates:
##       mean in group BERT mean in group DistilBERT
##               0.41876947               0.08000467
```

Manual evaliation of performance metrics GLEU and METEOR

```
d100 <- read_csv2("/Users/anitakurm/Downloads/df_samples_scores100_judged.csv")
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Warning: Missing column names filled in: 'X1' [1], 'X11' [11]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   BERT = col_character(),
##   DistilBERT = col_character(),
##   Gold_1 = col_character(),
##   Gold_2 = col_character(),
##   BERT_METEOR = col_double(),
##   DistilBERT_METEOR = col_double(),
##   BERT_GLEU = col_double(),
##   DistilBERT_GLEU = col_double(),
##   JUDGE = col_double(),
##   X11 = col_character()
## )
```

```
d100 <- d100 %>%
  mutate(Gold1_length = sapply(strsplit(Gold_1, " "), length),
         Gold2_length = ifelse(is.na(Gold_2), 0, sapply(strsplit(Gold_2, " "), length)),
         Gold_max_length = ifelse(Gold1_length>Gold2_length, Gold1_length, Gold2_length),
         Meteor_correct = ifelse(JUDGE == 0, 1, 0),
         length_cat = ifelse(Gold_max_length <= 2, " up to 2 (incl)", "greater than 2"),
         length_cat2 = ifelse(Gold_max_length <= 3, " up to 3 (incl)", "greater than 3"))

d100 %>%
  group_by(X11) %>%
  summarise(n())
```

```
## # A tibble: 3 x 2
##   X11   `n()`
##   <chr> <int>
## 1 C         6
## 2 F        11
## 3 <NA>     83
```

```
d100_compare <- d100 %>%
  filter(!is.na(JUDGE))

d100_compare %>%
  summarise('GLEU correct' = sum(JUDGE),
            'METEOR correct' = sum(Meteor_correct))
```

```
## # A tibble: 1 x 2
##   `GLEU correct` `METEOR correct`
##            <dbl>            <dbl>
## 1             48               34
```

```
d100_compare %>%
  group_by('Gold Standard Length' = length_cat) %>%
  summarise('GLEU correct' = sum(JUDGE),
            'METEOR correct' = sum(Meteor_correct))
```

```
## # A tibble: 2 x 3
##   `Gold Standard Length` `GLEU correct` `METEOR correct`
##   <chr>                           <dbl>            <dbl>
## 1 " up to 2 (incl)"                  39               21
## 2 greater than 2                      9               13
```

```
d100_compare %>%
  group_by('Gold Standard Length' = length_cat2) %>%
  summarise('GLEU correct' = sum(JUDGE),
            'METEOR correct' = sum(Meteor_correct))
```

```
## # A tibble: 2 x 3
##   `Gold Standard Length` `GLEU correct` `METEOR correct`
##   <chr>                           <dbl>            <dbl>
## 1 " up to 3 (incl)"                  42               26
## 2 greater than 3                      6                8
```

```
d100_compare %>%
  group_by('Gold Standard Length' = Gold_max_length) %>%
  summarise('GLEU correct' = sum(JUDGE),
            'METEOR correct' = sum(Meteor_correct))
```

```
## # A tibble: 7 x 3
##   `Gold Standard Length` `GLEU correct` `METEOR correct`
##                    <dbl>          <dbl>            <dbl>
## 1                      1             19                6
## 2                      2             20               15
## 3                      3              3                5
## 4                      4              4                2
## 5                      5              1                2
## 6                      6              0                3
## 7                      7              1                1
```

Applying METEOR for long answers dataset and GLEU for short answers dataset

```
# Read in the data
long = read.csv("df_long_answers.csv")
short = read.csv("df_short_answers.csv")
# Take only necessary columns
long = select(long,7:10)
short = select(short,7:10)

# re-define color palette so it's consistent across plots
cp <- c("aquamarine3", "grey19")
```

Long

```
bert = select(long,1,3)
bert['Model'] = 'BERT'
names(bert)[1] <- "METEOR"
names(bert)[2] <- "GLEU"
distil = select(long, 2,4)
distil['Model'] = 'DistilBERT'
names(distil)[1] <- "METEOR"
names(distil)[2] <- "GLEU"
data = rbind(bert, distil)
mu <- plyr::ddply(data, "Model", summarise, grp.mean=mean(METEOR))
mu
```

```
##        Model  grp.mean
## 1       BERT 0.4334346
## 2 DistilBERT 0.3874651
```

```
meteor <- ggplot(data, aes(x=METEOR, fill=Model)) +
  geom_density(col = NA, alpha=0.6, position="identity") +
  geom_vline(data=mu, aes(xintercept=grp.mean, color=Model),
             linetype="dashed") +
  geom_text(aes(x=0.387, label="\nMean: 0.387", y=2.5), colour="grey23", angle=90, size=4, family = "Raleway
") +
  geom_text(aes(x=0.433 , label="\nMean: 0.433  ", y=2.5), colour="aquamarine4", angle=90, size=4, family =
"Raleway")+
  theme_bw() +
  ylim(0,3.5)+
  theme(text = element_text(family = "Raleway"))+
  scale_colour_manual(values=cp)+
  scale_fill_manual(values=cp)+
  labs(x = "METEOR score",
       y = "Density")

meteor_by_model <- lm(METEOR ~ Model, data)
summary(meteor_by_model)
```

```
##
## Call:
## lm(formula = METEOR ~ Model, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43343 -0.32549 -0.01586  0.27816  0.61253
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.433435   0.005289  81.945  < 2e-16 ***
## ModelDistilBERT -0.045970   0.007480  -6.145 8.37e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3269 on 7638 degrees of freedom
## Multiple R-squared:  0.00492,    Adjusted R-squared:  0.00479
## F-statistic: 37.77 on 1 and 7638 DF,  p-value: 8.374e-10
```

```
t.test(METEOR ~ Model, data)
```

```
##
##  Welch Two Sample t-test
##
## data:  METEOR by Model
## t = 6.1454, df = 7637.8, p-value = 8.374e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03130618 0.06063291
## sample estimates:
##       mean in group BERT mean in group DistilBERT
##                0.4334346                0.3874651
```

Short

```
bert = select(short,1,3)
bert['Model'] = 'BERT'
names(bert)[1] <- "METEOR"
names(bert)[2] <- "GLEU"
distil = select(short, 2,4)
distil['Model'] = 'DistilBERT'
names(distil)[1] <- "METEOR"
names(distil)[2] <- "GLEU"
data = rbind(bert, distil)
mu <- plyr::ddply(data, "Model", summarise, grp.mean=mean(GLEU))
mu
```

```
##        Model  grp.mean
## 1       BERT 0.6143202
## 2 DistilBERT 0.5349806
```

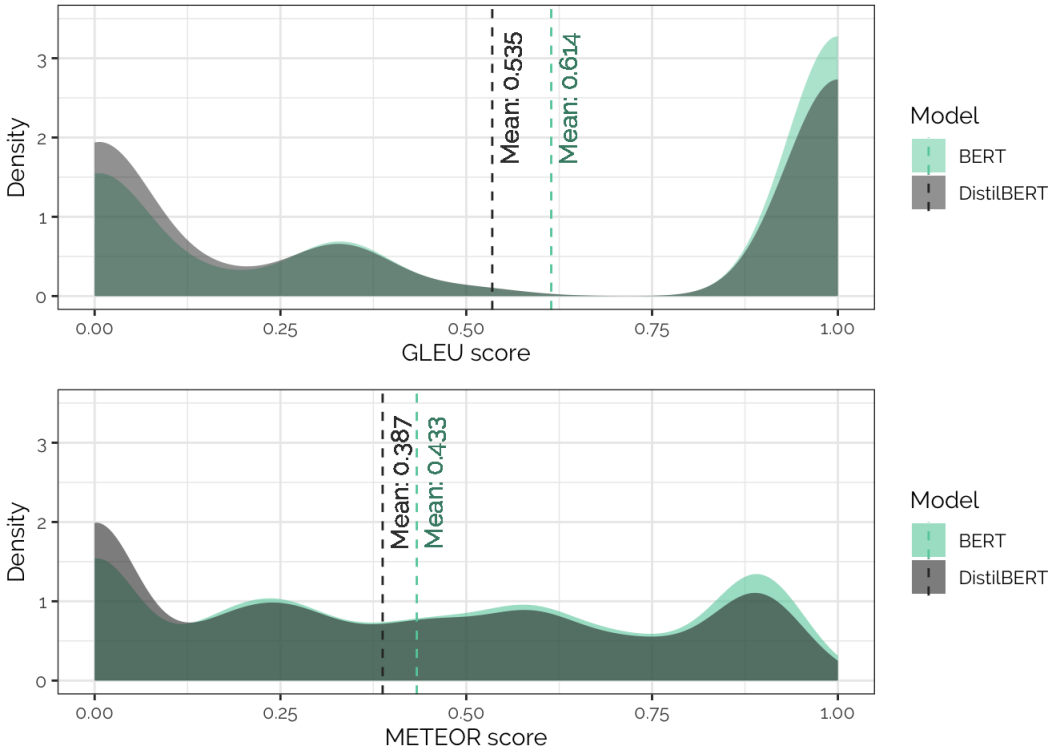
```
gleu <- ggplot(data, aes(x=GLEU, fill=Model)) +
  geom_density(col = NA, alpha=0.5, position="identity") +
  geom_vline(data=mu, aes(xintercept=grp.mean, color=Model),
             linetype="dashed") +
  geom_text(aes(x=0.535, label="\nMean: 0.535", y=2.5), colour="grey23", angle=90, size=4, family = "Raleway
") +
  geom_text(aes(x=0.614 , label="\nMean: 0.614  ", y=2.5), colour="aquamarine4", angle=90, size=4, family =
"Raleway")+
  theme_bw() +
  xlim(0,1) +
  ylim(0,3.5)+
  theme(text = element_text(family = "Raleway"))+
  scale_colour_manual(values=cp)+
  scale_fill_manual(values=cp)+
  labs(x = "GLEU score",
       y = "Density")

gleu_by_model <- lm(GLEU ~ Model, data)
summary(gleu_by_model)
```

```
##
## Call:
## lm(formula = GLEU ~ Model, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6143 -0.5350  0.3857  0.3857  0.4650
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.614320   0.005330  115.27   <2e-16 ***
## ModelDistilBERT -0.079340   0.007537  -10.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4536 on 14488 degrees of freedom
## Multiple R-squared:  0.00759,    Adjusted R-squared:  0.007522
## F-statistic: 110.8 on 1 and 14488 DF,  p-value: < 2.2e-16
```

```
t.test(GLEU ~ Model, data)
```

```
##
##  Welch Two Sample t-test
##
## data:  GLEU by Model
## t = 10.526, df = 14477, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.06456574 0.09411336
## sample estimates:
##       mean in group BERT mean in group DistilBERT
##                0.6143202                0.5349806
```

```
met_gl <- gridExtra::grid.arrange(gleu, meteor, nrow = 2)
```




```
ggsave("meteor_gleu.png", met_gl,width = 9, height = 8)
```

Gather citations

```
citation()
```

```
##
## To cite R in publications use:
##
##   R Core Team (2019). R: A language and environment for
##   statistical computing. R Foundation for Statistical Computing,
##   Vienna, Austria. URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2019},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please
## cite it when using it for data analysis. See also
## 'citation("pkgname")' for citing R packages.
```

```
citation("tidyverse")
```

```
##
## To cite package 'tidyverse' in publications use:
##
##   Hadley Wickham (2017). tidyverse: Easily Install and Load the
##   'Tidyverse'. R package version 1.2.1.
##   https://CRAN.R-project.org/package=tidyverse
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {tidyverse: Easily Install and Load the 'Tidyverse'},
##     author = {Hadley Wickham},
##     year = {2017},
##     note = {R package version 1.2.1},
##     url = {https://CRAN.R-project.org/package=tidyverse},
##   }
```

```
citation("rjson")
```

```
##
## To cite package 'rjson' in publications use:
##
##   Alex Couture-Beil (2018). rjson: JSON for R. R package version
##   0.2.20. https://CRAN.R-project.org/package=rjson
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {rjson: JSON for R},
##     author = {Alex Couture-Beil},
##     year = {2018},
##     note = {R package version 0.2.20},
##     url = {https://CRAN.R-project.org/package=rjson},
##   }
##
## ATTENTION: This citation information has been auto-generated from
## the package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

```
citation("extrafont")
```

```
##
## To cite package 'extrafont' in publications use:
##
##   Winston Chang, (2014). extrafont: Tools for using fonts. R
##   package version 0.17.
##   https://CRAN.R-project.org/package=extrafont
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {extrafont: Tools for using fonts},
##     author = {Winston Chang,},
##     year = {2014},
##     note = {R package version 0.17},
##     url = {https://CRAN.R-project.org/package=extrafont},
##   }
##
## ATTENTION: This citation information has been auto-generated from
## the package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```