Legal Comprehension & Summarizing via Annotation I Project

Jessica Macaluso

2025-03-31

First, let's disable scientific notation and then read in `tidyverse`.

```
options(scipen = 999)
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

Next, let's read in our dataframes.

Here is our Cohen's Kappa annotation data.

```
kappa <- read.csv(file='kappa2_F22_F23.csv')
head(kappa)
```

```
##        ID        Group     Test1     Test2  kappa_cwf  kappa_dks  kappa_avg
## 1 P01_F22 Gloss_Second 0.7500000 1.0000000 0.36469849 0.58348294 0.47409072
## 2 P02_F22 Gloss_Second 0.5000000 0.7500000 0.14290211 0.03030303 0.08660257
## 3 P03_F22  Gloss_First 0.5000000 1.0000000 0.08247504 0.11259383 0.09753443
## 4 P04_F22  Gloss_First 0.2500000 0.8333333 0.28462300 0.23380282 0.25921291
## 5 P05_F22 Gloss_Second 0.4166667 0.7500000 0.39439543 0.34954628 0.37197086
## 6 P06_F22 Gloss_Second 0.2500000 0.8333333 0.00000000 0.20239880 0.10119940
##     time kappa_ush
## 1 Fall22        NA
## 2 Fall22        NA
## 3 Fall22        NA
## 4 Fall22        NA
## 5 Fall22        NA
## 6 Fall22        NA
```

Here is our writing test performance dataframe.

```
writing <- read.csv(file='writing2_F22_F23.csv')
head(writing)
```

```
##        ID        Group Pre_Framework Pre_CaseFacts WritingTest1 Post_Framework
## 1 P06_F22 Gloss_Second             2             2         0.23              3
## 2 P01_F22 Gloss_Second             3             2         0.75              2
## 3 P13_F22 Gloss_Second             2             1         0.25              2
## 4 P04_F22  Gloss_First             2             1         0.25              3
## 5 P11_F22 Gloss_Second             2             1         0.17              2
## 6 P03_F22  Gloss_First             2             2         0.50              3
##   Post_CaseFacts WritingTest2
## 1              2         0.83
## 2              3         1.00
## 3              3         0.92
## 4              3         0.83
## 5              3         1.00
## 6              3         1.00
```

Now, let’s merge our dataframes by participant.

```
merge(kappa, writing, by='ID', all = TRUE) -> full_F22_F23
head(full_F22_F23)
```

```
##        ID      Group.x     Test1     Test2  kappa_cwf  kappa_dks  kappa_avg
## 1 P01_F22 Gloss_Second 0.7500000 1.0000000 0.36469849 0.58348294 0.47409072
## 2 P02_F22 Gloss_Second 0.5000000 0.7500000 0.14290211 0.03030303 0.08660257
## 3 P03_F22  Gloss_First 0.5000000 1.0000000 0.08247504 0.11259383 0.09753443
## 4 P04_F22  Gloss_First 0.2500000 0.8333333 0.28462300 0.23380282 0.25921291
## 5 P04_F23  Gloss_First 0.7777778        NA 0.00000000 0.34577603 0.11525868
## 6 P05_F22 Gloss_Second 0.4166667 0.7500000 0.39439543 0.34954628 0.37197086
##     time kappa_ush      Group.y Pre_Framework Pre_CaseFacts WritingTest1
## 1 Fall22        NA Gloss_Second             3             2         0.75
## 2 Fall22        NA Gloss_Second             2             1         0.50
## 3 Fall22        NA  Gloss_First             2             2         0.50
## 4 Fall22        NA  Gloss_First             2             1         0.25
## 5 Fall23         0  Gloss_First             2             1         0.17
## 6 Fall22        NA Gloss_Second             2             1         0.42
##   Post_Framework Post_CaseFacts WritingTest2
## 1              2              3         1.00
## 2              2              2         0.75
## 3              3              3         1.00
## 4              3              3         0.83
## 5              3              2         0.67
## 6              3              3         0.75
```

```
nrow(full_F22_F23)
```

```
## [1] 35
```

We have 35 participants/students from both Fall 2022 and Fall 2023.

Let’s make a new dataframe with just the columns of interest.

```
simple_df <- full_F22_F23 %>% select(ID, Group.x, Test1, Test2,
                                     kappa_avg, WritingTest1, WritingTest2)
simple_df <- rename(simple_df,c("Group" = "Group.x"))
simple_df <- rename(simple_df,c("KappaAverage" = "kappa_avg"))
simple_df
```

```
##         ID        Group     Test1     Test2 KappaAverage WritingTest1
## 1  P01_F22 Gloss_Second 0.7500000 1.0000000  0.474090718         0.75
## 2  P02_F22 Gloss_Second 0.5000000 0.7500000  0.086602570         0.50
## 3  P03_F22  Gloss_First 0.5000000 1.0000000  0.097534433         0.50
## 4  P04_F22  Gloss_First 0.2500000 0.8333333  0.259212909         0.25
## 5  P04_F23  Gloss_First 0.7777778        NA  0.115258677         0.17
## 6  P05_F22 Gloss_Second 0.4166667 0.7500000  0.371970856         0.42
## 7  P05_F23  Gloss_First        NA 0.7777778  0.086182056         0.00
## 8  P06_F22 Gloss_Second 0.2500000 0.8333333  0.101199400         0.23
## 9  P06_F23  Gloss_First 0.6666667 0.7777778  0.290739506         0.58
## 10 P07_F22  Gloss_First 0.5833333 0.6666667  0.009899208         0.58
## 11 P07_F23  Gloss_First 0.4444444 0.5555556  0.329617282         0.42
## 12 P08_F22  Gloss_First 0.7500000 0.4166667  0.330706707         0.75
## 13 P08_F23  Gloss_First 0.6666667 0.7777778  0.233946378         0.58
## 14 P09_F22  Gloss_First 0.5833333 0.3333333  0.124388510         0.58
## 15 P09_F23  Gloss_First 0.8888889 0.6666667           NA         0.58
## 16 P10_F22 Gloss_Second 0.3333333 0.8333333  0.343770562         0.37
## 17 P10_F23  Gloss_First 0.7777778 0.7777778  0.078390254         0.58
## 18 P11_F22 Gloss_Second 0.1666667 1.0000000  0.211465112         0.17
## 19 P11_F23  Gloss_First 0.6666667 0.8888889  0.056361100         0.33
## 20 P12_F22  Gloss_First 0.5000000 0.8333333  0.088761175         0.50
## 21 P12_F23 Gloss_Second 0.6666667 0.6666667  0.000000000         0.25
## 22 P13_F22 Gloss_Second 0.2500000 0.9166667  0.068252866         0.25
## 23 P13_F23 Gloss_Second 0.5555556 0.7777778  0.000000000         0.17
## 24 P14_F22  Gloss_First 0.5000000 1.0000000  0.159643674         0.50
## 25 P14_F23 Gloss_Second 0.3333333 0.5555556  0.000000000         0.67
## 26 P15_F22  Gloss_First 0.1666667 0.5833333  0.045741325         0.17
## 27 P15_F23 Gloss_Second 0.7777778 0.3333333  0.000000000         0.25
## 28 P16_F22  Gloss_First 0.2500000 0.7500000  0.203334372         0.25
## 29 P16_F23 Gloss_Second 0.6666667        NA           NA         0.42
## 30 P17_F22  Gloss_First 0.9166667 0.5000000  0.120724687         0.92
## 31 P17_F23 Gloss_Second 0.5555556 0.5555556  0.000000000         0.00
## 32 P18_F22         <NA>        NA        NA  0.000000000           NA
## 33 P18_F23 Gloss_Second 0.4444444        NA  0.000000000         0.83
## 34 P19_F22         <NA>        NA        NA  0.185704853           NA
## 35 P19_F23 Gloss_Second 0.4444444 0.4444444  0.000000000         0.33
##    WritingTest2
## 1          1.00
## 2          0.75
## 3          1.00
## 4          0.83
## 5          0.67
## 6          0.75
## 7          0.67
## 8          0.83
## 9          0.75
## 10         0.67
## 11         0.75
## 12         0.42
## 13         0.75
## 14         0.33
## 15         0.58
## 16         0.83
## 17         0.58
## 18         1.00
## 19         0.67
## 20         0.83
## 21         0.50
## 22         0.92
## 23         0.58
## 24         1.00
## 25         0.75
## 26         0.58
## 27         0.50
## 28         0.75
## 29         0.67
## 30         0.50
## 31         0.00
## 32           NA
## 33         0.83
## 34           NA
## 35         0.50
```

Let’s visualize our data!

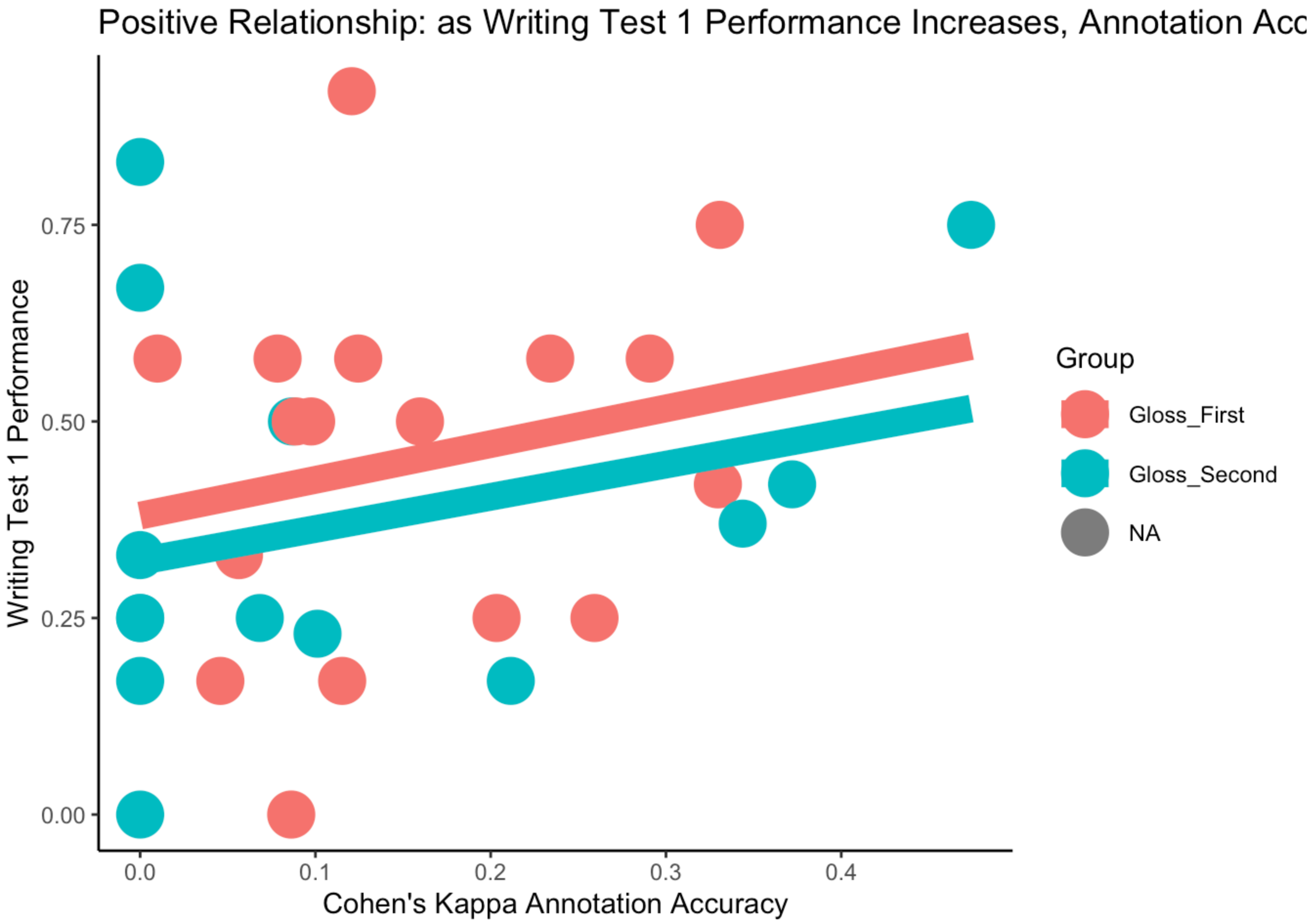
```
ggplot(data = simple_df, aes(x = KappaAverage, y = WritingTest1, color = Group)) +
  geom_point(size = 8) +
  geom_smooth(method = "lm", se = FALSE, fullrange = TRUE, aes(color = Group), size = 5)
+
  labs(x = "Cohen's Kappa Annotation Accuracy",
       y = "Writing Test 1 Performance",
       title = "Positive Relationship: as Writing Test 1 Performance Increases, Annotati
on Accuracy Increases for Both Gloss Groups") +
  theme_classic()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

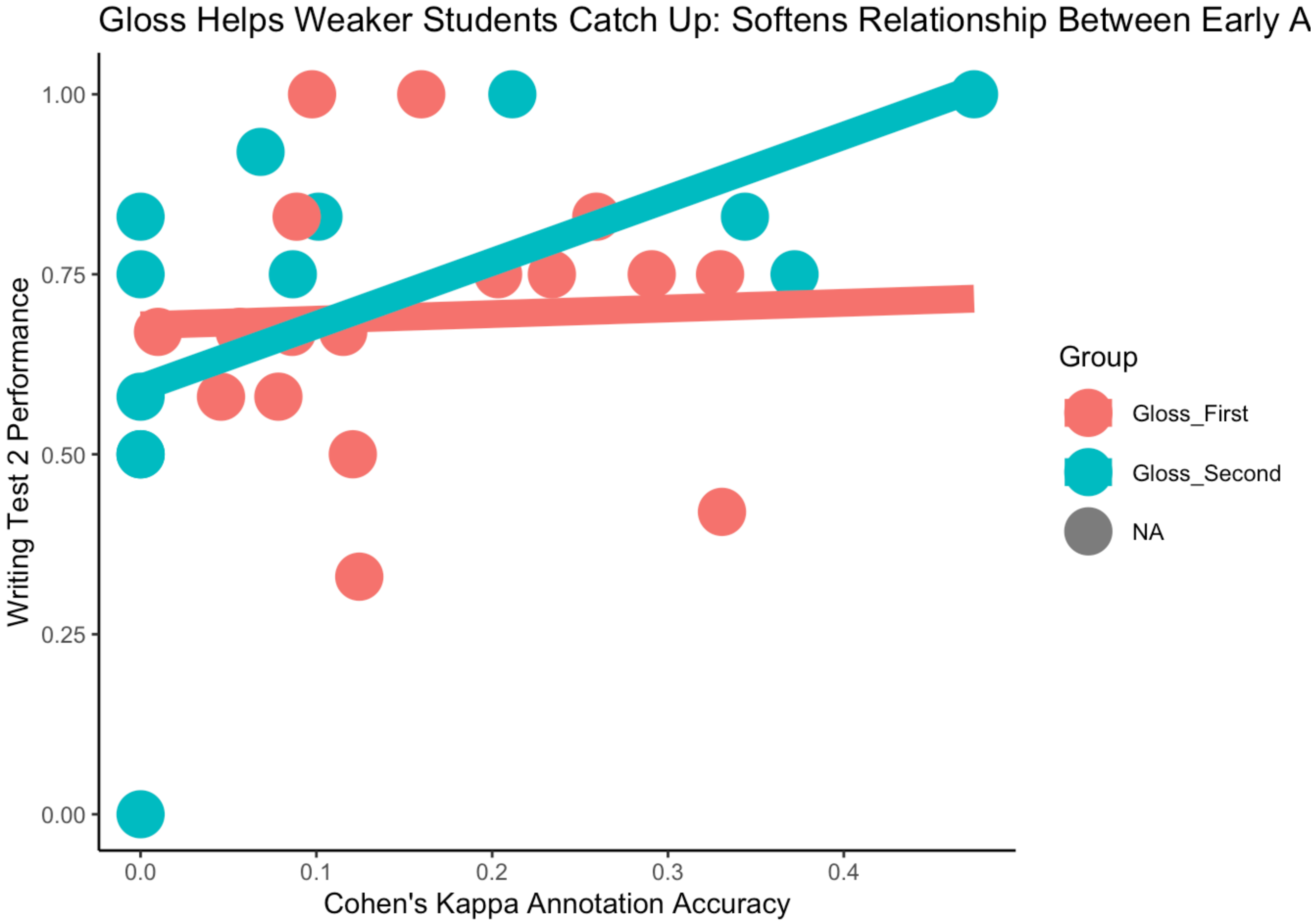


```
ggplot(data = simple_df, aes(x = KappaAverage, y = WritingTest2, color = Group)) +
  geom_point(size = 8) +
  geom_smooth(method = "lm", se = FALSE, fullrange = TRUE, aes(color = Group), size = 5)
+
  labs(x = "Cohen's Kappa Annotation Accuracy",
       y = "Writing Test 2 Performance",
       title = "Gloss Helps Weaker Students Catch Up: Softens Relationship Between Early
Accuracy & Writing Test 2 Performance") +
  theme_classic()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


Gloss Helps Weaker Students Catch Up: Softens Relationship Between Early A

Let’s run a few regression models to understand our findings.

```
model3 <- lm(WritingTest1 ~ 1 + KappaAverage + Group, data = simple_df)
model3 %>% summary()
```

```
## 
## Call:
## lm(formula = WritingTest1 ~ 1 + KappaAverage + Group, data = simple_df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.42154 -0.14199 -0.05817  0.14251  0.50945 
## 
## Coefficients:
##                   Estimate Std. Error t value  Pr(>|t|)    
## (Intercept)        0.38502    0.07536   5.109 0.0000206 ***
## KappaAverage       0.42373    0.32339   1.310     0.201    
## GroupGloss_Second -0.06447    0.08467  -0.761     0.453    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2323 on 28 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.08569,    Adjusted R-squared:  0.02038 
## F-statistic: 1.312 on 2 and 28 DF,  p-value: 0.2853
```

No significant effects.

```
model4 <- lm(WritingTest2 ~ 1 + KappaAverage + Group, data = simple_df)
model4 %>% summary()
```

```
##
## Call:
## lm(formula = WritingTest2 ~ 1 + KappaAverage + Group, data = simple_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62201 -0.08362  0.00951  0.10541  0.34443
##
## Coefficients:
##                   Estimate Std. Error t value      Pr(>|t|)
## (Intercept)        0.59485    0.06773   8.783 0.00000000156 ***
## KappaAverage       0.62257    0.29064   2.142         0.041 *
## GroupGloss_Second  0.02717    0.07610   0.357         0.724
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2088 on 28 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1409, Adjusted R-squared:  0.07953
## F-statistic: 2.296 on 2 and 28 DF,  p-value: 0.1193
```

Yes, significant differences for Cohen’s Kappa when looking at WritingTest2 performance.