Integrated ensemble

2024-11-20

Integrated ensemble

Function CM to compute F1 scores for both corpora

```
suppressMessages(suppressWarnings(library(caret)))
#install.packages("kableExtra")
suppressMessages(suppressWarnings(library(kableExtra)))

col.lab<-c("Recall_mean","Recall_sd","Preci_mean","Preci_sd","F1_mean","F1_s
d")
B.lab<-c("T","A","AW","De","S")

cm2<-function(X){
RES<-matrix(0,10,10)
X<-as.matrix(X)
for(i in 1:100)
{
res<-sort.list(X[i,-1],decreasing = TRUE)[1]
RES[X[i,1]+1,res]<-RES[X[i,1]+1,res]+1
}
a<-apply(confusionMatrix(RES,mode="prec_recall")$byClass[,c(6,5,7)],2,mean)
b<-apply(confusionMatrix(RES,mode="prec_recall")$byClass[,c(6,5,7)],2,sd)
res<-data.frame(rbind(a,b))
#res
res_vector<-c(res[1,1],res[2,1],res[1,2],res[2,2],res[1,3,],res[2,3])
names(res_vector)<-col.lab
list(vec=res_vector,matri=res)
}
```

Read BERTs test results (probability vectors)

```
corpusA<-list()
corpusB<-list()

suppressMessages(suppressWarnings(library(readxl)))
path1<-"C:/ensemble/CorpusA_BERT_Test_Results.xlsx"
BERT.A<-list()
for(i in 1:5)
  BERT.A[[i]] <- suppressMessages(suppressWarnings(
    read_excel(path1, sheet = i,col_names = FALSE)))

path2<-"C:/ensemble/CorpusB_BERT_Test_Results.xlsx"
BERT.B<-list()
for(i in 1:5)
  BERT.B[[i]] <- suppressMessages(suppressWarnings(
    read_excel(path2, sheet = i,col_names = FALSE)))
```

Compute BERTs scores

```
suppressMessages(suppressWarnings(library(knitr)))
# Calculating BERT scores for Corpus A
singBERT.A<-numeric()
for(i in 1:5)
  singBERT.A<-rbind(singBERT.A,cm2(BERT.A[[i]])[[1]])

singBERT.A<-data.frame(lab=B.lab,singBERT.A)

BERT.lab<-c("TohokuB","AozoraB","AozoraWikiB","DeBERTa","StockMarkB")
rownames(singBERT.A)<-BERT.lab
colnames(singBERT.A)<-c("lab",col.lab)
write.csv(singBERT.A,"C:/ensemble/singBERT.A.csv")
corpusA[[1]]<-singBERT.A[,6]

table2A<-singBERT.A;
table2A[,-1]<-round(table2A[,-1],3)
kable(table2A, format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

	lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1
TohokuB	T	0.653	0.209	0.64	0.201	0.642	0

	lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1
AozoraB	A	0.973	0.044	0.97	0.067	0.969	0
AozoraWikiB	AW	0.973	0.044	0.97	0.048	0.970	0
DeBERTa	De	0.752	0.215	0.68	0.162	0.691	0
StockMarkB	S	0.619	0.201	0.60	0.211	0.600	0

```
# Calculating BERT scores for Corpus B
path2<-"C:/ensemble/CorpusB_BERT_Test_Results.xlsx"
BERT.B<-list()
for(i in 1:5)
  BERT.B[[i]] <- suppressMessages(read_excel(path2, sheet = i,col_names = FA
LSE))

singBERT.B<-numeric()
for(i in 1:5)
  singBERT.B<-rbind(singBERT.B,cm2(BERT.B[[i]])[[1]])

singBERT.B<-as.matrix(singBERT.B)
colnames(singBERT.B)<-col.lab
rownames(singBERT.B)<-BERT.lab
singBERT.B<-data.frame(lab=B.lab,singBERT.B)
write.csv(singBERT.B,"C:/ensemble/singBERT.B.csv")

table2B<-singBERT.B;
table2B[,-1]<-round(table2B[,-1],3)
kable(table2B, format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

	lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1
TohokuB	T	0.762	0.151	0.74	0.117	0.744	0
AozoraB	A	0.813	0.210	0.77	0.195	0.773	0
AozoraWikiB	AW	0.838	0.116	0.82	0.114	0.820	0

	lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1
DeBERTa	De	0.834	0.118	0.82	0.063	0.823	0
StockMarkB	S	0.706	0.126	0.69	0.110	0.692	0

```
corpusB[[1]]<-singBERT.B[,6]
table2<-data.frame(rbind(table2A,table2B))
write.csv(table2,"C:/ensemble/table2.csv")

#corpusB[[1]]
```

Ensemble BERTs functions for both corpora

Ensemble of two BERTs

```
col.lab<-c("Recall_mean","Recall_sd","Preci_mean","Preci_sd","F1_mean","F1_s
d")
B.lab<-c("T","A","AW","De","S")

ensemb2<-function(BB,w){
#BB is the test result list data.
#w is the weight vector .
#ensemble of any two BERTs

B.lab<-c("T","A","AW","De","S")
comb2<-combn(1:5, 2)
string=c()
resu=numeric()
for(i in 1:ncol(comb2)){
J<-comb2[1,i]
K<-comb2[2,i]
X<-BB[[J]]; X[,-1]<-X[,-1]*w[1]+w[2]*BB[[K]][,-1]

#Calculate F1 values, etc. from contingency table
string=c(string,paste("{",B.lab[J],",",B.lab[K],"}"))
resu<-rbind(resu,cm2(X)[[1]])
  }
resu<-data.frame(resu)
colnames(resu)<-col.lab
resu<-data.frame(comb=string,resu)
resu
}
```

Ensemble of three BERTs

```
# Ensemble of three BERT results
ensemb3<-function(BB,w){
comb3<-combn(1:5, 3)
string=c()
resu=numeric()

for(i in 1:ncol(comb3)){
J<-comb3[1,i]
K<-comb3[2,i]
L<-comb3[3,i]
X<-BB[[J]]
X[,-1]<-X[,-1]*w[J]+w[K]*BB[[K]][,-1]+w[L]*BB[[L]][,-1]

string=c(string,paste("{",B.lab[J],",",B.lab[K],",",B.lab[L],"}"))
resu<-rbind(resu,cm2(X)[[1]])
}
resu<-data.frame(resu)
colnames(resu)<-col.lab
resu<-data.frame(comb=string,resu)
resu
}
```

Ensemble of four annd five BERTs

```
# Ensemble of 4 and 5 BERT results
ensemb4<-function(BB,w){
string=c()
resu=numeric()

comb4<-combn(1:5, 4)
for(i in 1:ncol(comb4)){
J<-comb4[1,i]
K<-comb4[2,i]
L<-comb4[3,i]
G<-comb4[4,i]
X<-BB[[J]];
X[,-1]<-X[,-1]*w[J]+w[K]*BB[[K]][,-1]+BB[[L]][,-1]*w[L]+w[G]*BB[[G]][,-1]
string=c(string,paste("{",B.lab[J],",",B.lab[K],",",B.lab[L],",",B.lab
[G],"}"))
resu<-rbind(resu,cm2(X)[[1]])
}

#Ensemble of five BERT results
X<-BB[[1]];X[,-1]<-X[,-1]*w[1]+w[2]*BB[[2]][,-1]+BB[[3]][,-1]*w[3]+w[4]*BB
[[4]][,-1]+w[5]*BB[[5]][,-1]

resu<-rbind(resu,cm2(X)[[1]])
resu<-data.frame(resu)
colnames(resu)<-col.lab
resu<-data.frame(comb=c(string,"{T,A,AW,De,S}"),resu)
resu
}
```

Compute ensemble results for corpus A

```
# Calculate the results of the corpus A ensemble
w<-c(1,1,1,1,1)
resu.A2<-ensemb2(BERT.A, w)
resu.A3<-ensemb3(BERT.A, w)
resu.A4<-ensemb4(BERT.A, w)
ensemBERT.A<-rbind(resu.A2,resu.A3,resu.A4)
temp<-sort.list(ensemBERT.A[,6],decreasing=TRUE)
ensemBERT.A<-data.frame(ensemBERT.A[temp,])
rownames(ensemBERT.A)<-1:dim(ensemBERT.A)[1]
write.csv(ensemBERT.A,"C:/ensemble/ensemBERT.A.csv")
corpusA[[2]] <- ensemBERT.A[,6]
kable(head(ensemBERT.A),format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ A , S }	0.9909091	0.0287480	0.99	0.0316228	0.9899749	0.0211677
{ T , A }	0.9833333	0.0527046	0.98	0.0421637	0.9803828	0.0332604
{ A , AW }	0.9818182	0.0383306	0.98	0.0421637	0.9799499	0.0259385
{ A , De }	0.9818182	0.0383306	0.98	0.0421637	0.9799499	0.0259385
{ AW , S }	0.9818182	0.0383306	0.98	0.0421637	0.9799499	0.0259385
{ A , AW , De }	0.9818182	0.0383306	0.98	0.0421637	0.9799499	0.0259385

```
# Calculate the results of the corpus A weighted ensemble
w<-singBERT.A[,6]
resu.A2<-ensemb2(BERT.A, w)
resu.A3<-ensemb3(BERT.A, w)
resu.A4<-ensemb4(BERT.A, w)
ensemBERT.A.W<-rbind(resu.A2,resu.A3,resu.A4)
temp<-sort.list(ensemBERT.A.W[,6],decreasing=TRUE)
ensemBERT.A.W<-data.frame(ensemBERT.A.W[temp,])
rownames(ensemBERT.A.W)<-1:dim(ensemBERT.A.W)[1]
write.csv(ensemBERT.A.W,"C:/ensemble/ensemBERT.A.W.csv")

corpusA[[3]] <- ensemBERT.A.W[,6]

ensemBERT.A[,6]
```

```
##  [1] 0.9899749 0.9803828 0.9799499 0.9799499 0.9799499 0.9799499 0.969924
8
##  [8] 0.9699248 0.9699248 0.9699248 0.9599499 0.9497479 0.9411331 0.940656
9
## [15] 0.9399749 0.9313534 0.9304261 0.9097744 0.9013603 0.8913390 0.879849
6
## [22] 0.8723099 0.7444652 0.7186510 0.7183238 0.6818886
```

Compute ensemble results for corpus B

```
# Calculate the results of the corpus A ensemble
w<-c(1,1,1,1,1)
resu.B2<-ensemb2(BERT.B,w)
resu.B3<-ensemb3(BERT.B,w)
resu.B4<-ensemb4(BERT.B,w)
ensemBERT.B<-rbind(resu.B2,resu.B3,resu.B4)

temp<-sort.list(ensemBERT.B[,6],decreasing=TRUE)
ensemBERT.B<-data.frame(ensemBERT.B[temp,])
rownames(ensemBERT.B)<-1:dim(ensemBERT.B)[1]
write.csv(ensemBERT.B,"C:/ensemble/ensemBERT.B.csv")
corpusB[[2]]<-ensemBERT.B[,6]
kable(head(ensemBERT.B),format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

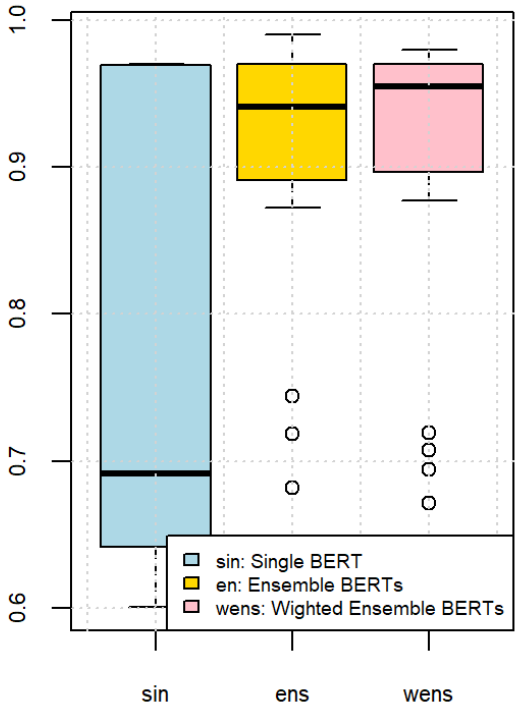
comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	I
{ T , A , AW , De }	0.9116162	0.1127632	0.90	0.0816497	0.9018987	0.07
{ A , AW , De , S }	0.9083333	0.1034139	0.90	0.0816497	0.9010898	0.07
{ T , A , De , S }	0.9048718	0.1354880	0.89	0.0875595	0.8935737	0.09
{T,A,AW,De,S}	0.9005051	0.1084794	0.89	0.0875595	0.8913724	0.07
{ T , AW , De , S }	0.8985897	0.0881609	0.89	0.0875595	0.8902952	0.06
{ T , A , AW }	0.8940948	0.1260121	0.88	0.0918937	0.8821566	0.08

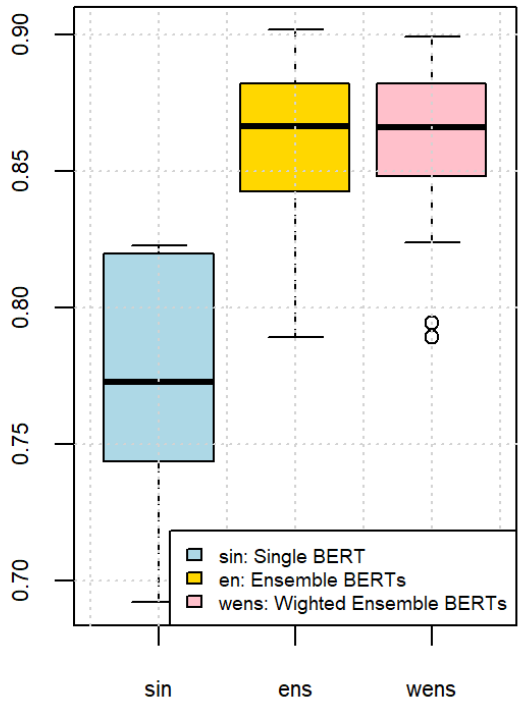
```
# Calculate the results of the corpus B weighted ensemble
w<-singBERT.B[,6]
resu.B2<-ensemb2(BERT.B,w)
resu.B3<-ensemb3(BERT.B,w)
resu.B4<-ensemb4(BERT.B,w)
ensemBERT.B.W<-rbind(resu.B2,resu.B3,resu.B4)
temp<-sort.list(ensemBERT.B.W[,6],decreasing=TRUE)
ensemBERT.B.W<-data.frame(ensemBERT.B.W[temp,])
rownames(ensemBERT.B.W)<-1:dim(ensemBERT.B)[1]
write.csv(ensemBERT.B.W,"C:/ensemble/ensemBERT.B.W.csv")

corpusB[[3]]<-ensemBERT.B.W[,6]
ensemBERT.B.W[,6]
```

```
##  [1] 0.8994274 0.8935737 0.8927645 0.8913724 0.8902952 0.8889011 0.882082
9
##  [8] 0.8807876 0.8802864 0.8730224 0.8704983 0.8703797 0.8700513 0.862399
9
## [15] 0.8619462 0.8582819 0.8535494 0.8533235 0.8522264 0.8483875 0.839776
3
## [22] 0.8350068 0.8337710 0.8241006 0.7944480 0.7889839
```

Draw the box plot of BERT's and ensemble F1 scores

```
# Drawing the resulting box plot
par(mfrow=c(1,2), cex.lab = 0.7, cex.axis = 0.7)
lab=factor(c(rep("sin",each=5),rep("ens",each=26),rep("wens",each=26)),level
s = c("sin", "ens","wens"))
col<-c("lightblue","gold1","pink")
#Boxplot of F1 values for Corpus A
F1<-c(singBERT.A[,6],ensemBERT.A[,6],ensemBERT.A.W[,6])
BERT.A.resu<-data.frame(lab=lab,F1=F1)
boxplot(F1 ~ lab, data = BERT.A.resu,col=col,ylab="",xlab="");grid()
legend("bottomright", legend = c("sin: Single BERT", "en: Ensemble BERTs","w
ens: Wighted Ensemble BERTs"), fill = col, cex = 0.6)
#Boxplot of F1 values for Corpus B
F1<-c(singBERT.B[,6],ensemBERT.B[,6],ensemBERT.B.W[,6])
BERT.B.resu<-data.frame(lab=lab,F1=F1)
boxplot(F1 ~ lab, data = BERT.B.resu,col=col,ylab="",xlab="");grid()
legend("bottomright", legend = c("sin: Single BERT", "en: Ensemble BERTs","w
ens: Wighted Ensemble BERTs"), fill = col, cex = 0.6)
```





Load Features & Classifiers test results

```
library(readxl)
path3<-"C:/ensemble/CorpusA_Feature&Classifiers_Test_Results.xlsx"
AdaRF.A<-list()
#Ada.Char,Ada.Token, Ada.Pattern,RF.Char,RF.Token, RF.Pattern
for(i in 1:6)
AdaRF.A[[i]] <- suppressMessages(read_excel(path3, sheet = i,col_names = FAL
SE))

path4<-"C:/ensemble/CorpusB_Feature&Classifiers_Test_Results.xlsx"
AdaRF.B<-list()
#Ada.Char,Ada.Token, Ada.Pattern,RF.Char,RF.Token, RF.Pattern
for(i in 1:6)
  AdaRF.B[[i]] <- suppressMessages(read_excel(path4, sheet = i,col_names = F
ALSE))
```

Compute single F & C scores for Corpus A

```
resu<-numeric()
for(i in 1:6) resu<-rbind(resu,cm2(AdaRF.A[[i]])[[1]])

##Table type the results of corpus A
resu<-data.frame(resu)
colnames(resu)<-col.lab
F.lab<-c("Ada.Char","Ada.Token","Ada.Pattern","RF.Char","RF.Token","RF.Patte
rn")
FeaClas.A<-data.frame(lab=F.lab,resu)
corpusA[[4]]<-FeaClas.A[,6]
write.csv(FeaClas.A,"C:/ensemble/FeaClas.A.csv")
table3A<-FeaClas.A
table3A[,-1]<-round(FeaClas.A[,-1],3)
kable(table3A,format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
Ada.Char	0.786	0.188	0.76	0.143	0.766	0.152
Ada.Token	0.767	0.149	0.75	0.097	0.754	0.109
Ada.Pattern	0.762	0.116	0.75	0.165	0.747	0.125
RF.Char	0.792	0.130	0.79	0.179	0.784	0.134
RF.Token	0.823	0.124	0.81	0.120	0.810	0.094
RF.Pattern	0.714	0.120	0.71	0.185	0.704	0.135

Compute single F & C scores for Corpus B

```
##Corpus B
resu<-numeric()
for(i in 1:6) resu<-rbind(resu,cm2(AdaRF.B[[i]])[[1]])

resu<-data.frame(resu)
colnames(resu)<-col.lab
FeaClas.B<-data.frame(lab=F.lab,resu)
write.csv(FeaClas.B,"C:/ensemble/FeaClas.B.csv")
corpusB[[4]]<-FeaClas.B[,6]

table3B<-FeaClas.B
table3B[,-1]<-round(FeaClas.B[,-1],3)
kable(table3B,format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
Ada.Char	0.779	0.125	0.76	0.117	0.761	0.083
Ada.Token	0.772	0.122	0.76	0.097	0.762	0.091
Ada.Pattern	0.654	0.131	0.65	0.158	0.647	0.131
RF.Char	0.780	0.106	0.78	0.215	0.767	0.155
RF.Token	0.810	0.109	0.80	0.105	0.800	0.090
RF.Pattern	0.668	0.150	0.65	0.178	0.643	0.142

```
table3<-data.frame(rbind(table3A,table3B))
write.csv(table3,"C:/ensemble/table3.csv")
```

Ensemble of F & C

Function of ensemble of two F & C

```
#Two ensemble functions
FEnsemb2<-function(FE,w){
comb2<-combn(1:6,2 )
string=c()
resu=numeric()

for(i in 1:ncol(comb2)){
J<-comb2[1,i]
K<-comb2[2,i]
X<-FE[[J]];X[,-1]<-w[J]*X[,-1]+w[K]*FE[[K]][,-1]
string=c(string,paste("{",J,",",K,"}"))
resu<-rbind(resu,cm2(X)[[1]])
  }
resu<-data.frame(resu)
colnames(resu)<-col.lab
resu<-data.frame(comb=string,resu)
resu
}
```

Function of ensemble of three F & C

```
FEnsemb3<-function(FE,w){
res<-list()
string=c()
resu=numeric()
comb3<-combn(1:6, 3)
for(i in 1:ncol(comb3)){
J<-comb3[1,i]
K<-comb3[2,i]
L<-comb3[3,i]
X<-FE[[J]]
X[,-1]<-w[J]*X[,-1]+w[K]*FE[[K]][,-1]+w[L]*FE[[L]][,-1]

string=c(string,paste("{",J,",",K,",",L,"}"))
resu<-rbind(resu,cm2(X)[[1]])
  }
resu<-data.frame(resu)
colnames(resu)<-col.lab
resu<-data.frame(comb=string,resu)
resu
}
```

Function of ensemble of four F & C

```
FEnsemb4<-function(FE,w){
res<-list()
string=c()
resu=numeric()
comb4<-combn(1:6, 4)
res<-list()
for(i in 1:ncol(comb4)){
  J<-comb4[1,i]
  K<-comb4[2,i]
  L<-comb4[3,i]
  G<-comb4[4,i]
  X<-FE[[J]]
  X[,-1]<-w[J]*X[,-1]+w[K]*FE[[K]][,-1]+w[L]*FE[[L]][,-1]+w[G]*FE[[G]][,-1]

string=c(string,paste("{",J,",",K,",",L,",",G,"}"))
resu<-rbind(resu,cm2(X)[[1]])
  }
resu<-data.frame(resu)
colnames(resu)<-col.lab
resu<-data.frame(comb=string,resu)
resu
}
```

Function of ensemble of five and six F & C

```
FEnsemb5<-function(FE,w){
string=c()
resu=numeric()
comb5<-combn(1:6, 5)

for(i in 1:ncol(comb5)){
J<-comb5[1,i]; K<-comb5[2,i];L<-comb5[3,i]
G<-comb5[4,i]; F<-comb5[5,i];
X<-FE[[J]]
X[,-1]<-w[J]*X[,-1]+w[K]*FE[[K]][,-1]+w[L]*FE[[L]][,-1]+w[G]*FE[[G]][,-1]+w
[F]*FE[[F]][,-1]

string=c(string,paste("{",J,",",K,",",L,",",G,",",F,"}"))
resu<-rbind(resu,cm2(X)[[1]])
}
## ensemble six F & C
X<-FE[[1]]
X[,-1]<-w[1]*X[,-1]+w[2]*FE[[2]][,-1]+w[3]*FE[[3]][,-1]+w[4]*FE[[4]][,-1]+
        w[5]*FE[[5]][,-1]+w[6]*FE[[6]][,-1]

resu<-rbind(resu,cm2(X)[[1]])

resu<-data.frame(resu)
colnames(resu)<-col.lab
resu<-data.frame(comb=c(string,paste
("{",1,",",2,",",3,",",4,",",5,",",6,"}")),resu)
resu
}
```

Compute the ensemble scores for corpus A

```
## Without weights
w<-c(1,1,1,1,1,1)
FEnsemb2(AdaRF.A, w)->res2
FEnsemb3(AdaRF.A, w)->res3
FEnsemb4(AdaRF.A, w)->res4
FEnsemb5(AdaRF.A, w)->res5
ensembFeature.A<-rbind(res2,res3,res4,res5)
temp<-sort.list(ensembFeature.A[,6],decreasing=TRUE)
ensembFeature.A<-data.frame(ensembFeature.A[temp,])
rownames(ensembFeature.A)<-1:dim(ensembFeature.A)[1]
write.csv(ensembFeature.A,"c:/ensemble/ensembFeature.A.csv")
corpusA[[5]]<-ensembFeature.A[1:50,6]

## Weighted ensemble for Corpus A
w<-FeaClas.A[,6]
FEnsemb2(AdaRF.A, w)->res2
FEnsemb3(AdaRF.A, w)->res3
FEnsemb4(AdaRF.A, w)->res4
FEnsemb5(AdaRF.A, w)->res5
ensembFeature.A.W<-rbind(res2,res3,res4,res5)
temp<-sort.list(ensembFeature.A.W[,6],decreasing=TRUE)
ensembFeature.A.W<-data.frame(ensembFeature.A.W[temp,])
rownames(ensembFeature.A.W)<-1:dim(ensembFeature.A.W)[1]

write.csv(ensembFeature.A.W,"c:/ensemble/ensembFeature.A.W.csv")
corpusA[[6]]<-ensembFeature.A.W[1:50,6]
kable(head(ensembFeature.A),format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ 1 , 3 , 5 }	0.9223621	0.1015360	0.91	0.0737865	0.9117180	0.0627522
{ 1 , 3 , 4 , 5 }	0.9223621	0.1015360	0.91	0.0737865	0.9117180	0.0627522
{ 1 , 3 , 4 , 5 , 6 }	0.9223621	0.1015360	0.91	0.0737865	0.9117180	0.0627522

comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ 1 , 3 , 5 , 6 }	0.9118823	0.0982960	0.90	0.0942809	0.9005233	0.0676110
{ 1 , 3 , 4 }	0.9032828	0.1219002	0.89	0.0737865	0.8928078	0.0821913
{ 1 , 2 , 3 , 6 }	0.8926651	0.1151089	0.88	0.0421637	0.8833235	0.0701059

Compute the ensemble scores for corpus B

```
## Without weights
w<-c(1,1,1,1,1,1)
FEnsemb2(AdaRF.B, w)->res2
FEnsemb3(AdaRF.B, w)->res3
FEnsemb4(AdaRF.B, w)->res4
FEnsemb5(AdaRF.B, w)->res5
ensembFeature.B<-rbind(res2,res3,res4,res5)

temp<-sort.list(ensembFeature.B[,6],decreasing=TRUE)
ensembFeature.B<-data.frame(ensembFeature.B[temp,])
rownames(ensembFeature.B)<-1:dim(ensembFeature.B)[1]
write.csv(ensembFeature.B,"c:/ensemble/ensembFeature.B.csv")
corpusB[[5]]<-ensembFeature.B[1:50,6]
kable(head(ensembFeature.B),format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ 1 , 2 , 6 }	0.8970202	0.0724072	0.89	0.0994429	0.8887394	0.0585035
{ 1 , 2 , 4 , 6 }	0.8970202	0.0724072	0.89	0.0994429	0.8887394	0.0585035

comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ 1 , 2 , 5 , 6 }	0.8970202	0.0724072	0.89	0.0994429	0.8887394	0.0585035
{ 1 , 2 , 4 , 5 , 6 }	0.8970202	0.0724072	0.89	0.0994429	0.8887394	0.0585035
{ 1 , 2 , 4 }	0.8722475	0.0726725	0.87	0.1059350	0.8685365	0.0772004
{ 4 , 5 , 6 }	0.8821900	0.0999370	0.87	0.1567021	0.8657890	0.1016219

```
## Weighted ensemble for Corpus B
w<-FeaClas.B[,6]
FEnsemb2(AdaRF.B, w)->res2
FEnsemb3(AdaRF.B, w)->res3
FEnsemb4(AdaRF.B, w)->res4
FEnsemb5(AdaRF.B, w)->res5
ensembFeature.B.W<-rbind(res2,res3,res4,res5)
temp<-sort.list(ensembFeature.B.W[,6],decreasing=TRUE)
ensembFeature.B.W<-data.frame(ensembFeature.B.W[temp,])
rownames(ensembFeature.B.W)<-1:dim(ensembFeature.B.W)[1]

write.csv(ensembFeature.B.W,"c:/ensemble/ensembFeature.B.W.csv")
kable(head(ensembFeature.B.W), format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

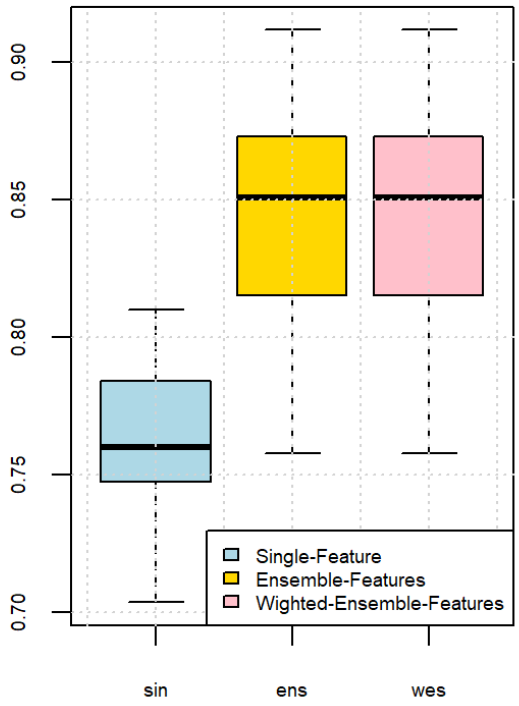
comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ 1 , 2 , 6 }	0.8970202	0.0724072	0.89	0.0994429	0.8887394	0.0585035
{ 1 , 2 , 4 , 6 }	0.8970202	0.0724072	0.89	0.0994429	0.8887394	0.0585035

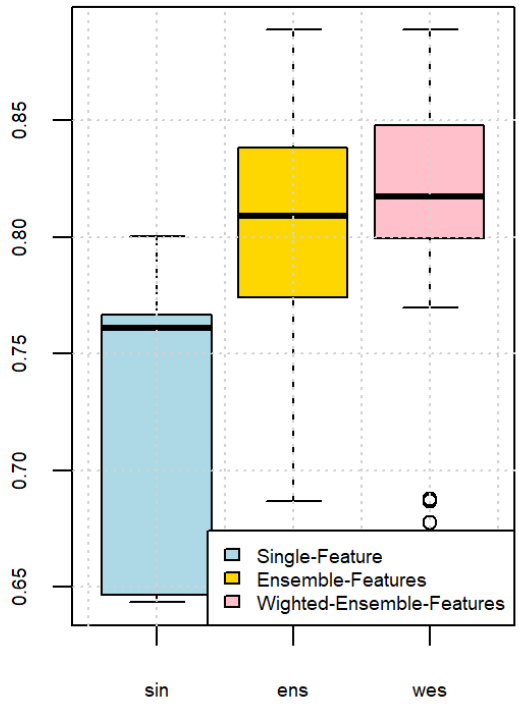
comb	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ 1 , 2 , 4 , 5 , 6 }	0.8970202	0.0724072	0.89	0.0994429	0.8887394	0.0585035
{ 1 , 2 , 5 , 6 }	0.8856566	0.0688267	0.88	0.1032796	0.8783498	0.0600967
{ 4 , 5 , 6 }	0.8836619	0.0987133	0.87	0.1251666	0.8686392	0.0775492
{ 1 , 2 , 4 }	0.8722475	0.0726725	0.87	0.1059350	0.8685365	0.0772004

```
corpusB[[6]]<-ensembFeature.B.W[1:50,6]
```

Draw the boxplot of F & C's and ensemble F1 scores

```
#Boxplot
par(mfrow=c(1,2),cex.lab = 0.6, cex.axis = 0.6)
col=c("lightblue","gold1","pink")
lab=factor(c(rep("sin",each=6),rep("ens",each=57),rep("wes",each=57)), level
s = c("sin", "ens","wes"))
#Draw boxplot of Corpus A F1 scores
F1<-c(FeaClas.A[,6],ensembFeature.A[,6],ensembFeature.A.W[,6])
FeaClas.AF1<-data.frame(lab=lab,F1=F1)
boxplot(F1 ~ lab, data = FeaClas.AF1, col=col,ylab="",xlab="");grid()
legend("bottomright", legend = c("Single-Feature", "Ensemble-Features","Wigh
ted-Ensemble-Features"), fill = col, cex = 0.6)
#Draw boxplot of Corpus B F1 scores
F1<-c(FeaClas.B[,6],ensembFeature.B[,6],ensembFeature.B.W[,6])
FeaClas.BF1<-data.frame(lab=lab,F1=F1)
boxplot(F1 ~ lab, data = FeaClas.BF1,col=col,ylab="",xlab="");grid()
legend("bottomright", legend = c("Single-Feature", "Ensemble-Features","Wigh
ted-Ensemble-Features"), fill = col, cex = 0.6)
```





Integrated ensemble

Compute method of Strom40 (ensemble%20for%20one%20BERT%20and%20one%

```
one2one<-function(XX,YY,xw,yw)
{
resu<-numeric()
string<-c()
k=0
for(i in 1:5)
  for(j in 1:6){
    k=k+1
    temp<-XX[[i]]
    temp[,-1]<-xw[i]*XX[[i]][,-1]+yw[j]*YY[[j]][,-1]
    resu<-rbind(resu,cm2(temp)[[1]])
    string=c(string,paste("{",B.lab[i],"|",j,"}"))
  }
 colnames(resu)<-col.lab
 resu<-data.frame(lab=string,resu)
 temp<-sort.list(resu[,6],decreasing=TRUE)
one2one.res<-resu[temp,]
one2one.res
}

#compute method of Strom40
xw<-c(1,1,1,1,1);yw<-c(1,1,1,1,1,1)
strom40A<-one2one(BERT.A,AdaRF.A,xw,yw)
corpusA[[8]]<-strom40A[,6]
head(strom40A[,6])
```

```
## [1] 0.9899749 0.9799499 0.9799499 0.9799499 0.9799499 0.9699499
```

```
xw<-c(1,1,1,1,1);yw<-c(1,1,1,1,1,1)
strom40B<-one2one(BERT.B,AdaRF.B,xw,yw)
corpusB[[8]]<-strom40B[,6]
head(strom40B[,6])
```

```
## [1] 0.9008898 0.8900725 0.8811990 0.8775273 0.8713562 0.8585129
```

Computing method of Wu[39]

```
Wu39<-function(XX,YY){
 #XX<-BERT.A; YY<-AdaRF.B
  Char<-(YY[[1]][,-1]+YY[[4]][,-1])/2
  Token<-(YY[[2]][,-1]+YY[[5]][,-1])/2
  Phra<-(YY[[3]][,-1]+YY[[6]][,-1])/2
  temp<-XX[[1]]
  temp[,-1]<-(XX[[1]][,-1]+XX[[2]][,-1]+XX[[3]][,-1]+
                XX[[4]][,-1]+XX[[5]][,-1])/5

temp1<-temp; temp1[,-1]<-temp1[,-1]+Char
res1<-cm2(temp1)[[1]]
temp2<-temp; temp2[,-1]<-temp2[,-1]+Token
res2<-cm2(temp2)[[1]]
temp3<-temp; temp3[,-1]<-temp3[,-1]+Phra
res3<-cm2(temp3)[[1]]
res<-rbind(res1,res2,res3)
res<-data.frame(lab=c("{5BERTs,1,4}","{5BERTs,2,5}","{5BERTs,3,6}"),res)
res
}

Wu39.A<-Wu39(BERT.A,AdaRF.A)
corpusA[[7]]<-Wu39.A[,6]
summary(Wu39.A[,6])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9393  0.9396  0.9399  0.9531  0.9599  0.9800
```

```
Wu39.B<-Wu39(BERT.B,AdaRF.B)
corpusB[[7]]<-Wu39.B[,6]
summary(Wu39.B[,6])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8884  0.9095  0.9305  0.9193  0.9347  0.9389
```

Integrated ensemble

```
interEnsemb<-function(BB,FF,wb,wf){
eBA.lab<-list()
comb2<-combn(1:5, 2);L2<-dim(comb2)[2]
comb3<-combn(1:5, 3);L3<-dim(comb3)[2]
comb4<-combn(1:5, 4);L4<-dim(comb4)[2]
comb5<-combn(1:5, 5)

# Set the label list for the BERTs combination
for(i in 1:L2) eBA.lab[[i]]<-comb2[,i]
for(i in 1:L3) {(L2+i)->k; eBA.lab[[k]]<-comb3[,i]}
for(i in 1:L4) {(L2+L3+i)->k; eBA.lab[[k]]<-comb4[,i]}
eBA.lab[[L2+L3+L4+1]]<-comb5[,1]

# Set the label list for the F & C combination
eFA.lab<-list()
comb2<-combn(1:6, 2);L2<-dim(comb2)[2]
comb3<-combn(1:6, 3);L3<-dim(comb3)[2]
comb4<-combn(1:6, 4);L4<-dim(comb4)[2]
comb5<-combn(1:6, 5);L5<-dim(comb5)[2]

for(i in 1:L2) eFA.lab[[i]]<-comb2[,i]
for(i in 1:L3) {(L2+i)->k; eFA.lab[[k]]<-comb3[,i]}
for(i in 1:L4) {(L2+L3+i)->k; eFA.lab[[k]]<-comb4[,i]}
for(i in 1:L5) {(L2+L3+L4+i)->k; eFA.lab[[k]]<-comb5[,i]}
eFA.lab[[L2+L3+L4+L5+1]]<-c(1,2,3,4,5,6)

#Compute for all combinations
resu<-c()
resLab<-c()
for(i in 1:length(eBA.lab)){
  for(j in 1:length(eFA.lab)){
    #Data creation for BERTs

    tempBi<-eBA.lab[[i]]
    LBi<-length(tempBi)
    BX<-BB[[tempBi[1]]];
    for(k in 2:LBi)
      BX[,-1]<-BX[,-1]*wb[tempBi[1]]+wb[tempBi[k]]*BB[[tempBi[k]]][,-1]
    BX[,-1]<-BX[,-1]/5

    #Data creation for F & C
    tempFj<-eFA.lab[[j]]
```

```
    LFj<-length(tempFj)
    FX<-FF[[tempFj[1]]]
    for(k in 2:LFj)
      FX[,-1]<-FX[,-1]*wf[tempFj[1]]+wf[tempFj[k]]*FF[[tempFj[k]]][,-1]
    FX[,-1]<-FX[,-1]/6

     #Integrative Ensemble
    BFX<-BX; BFX[,-1]<-BX[,-1]+FX[,-1]

    labi<- paste(B.lab[eBA.lab[[i]]], collapse = ",")
    labj<- paste(eFA.lab[[j]], collapse = ",")
    labij<-paste("{",labi,"|",labj,"}")
    resLab<-rbind(resLab,labij)
    resu<-rbind(resu,cm2(BFX)[[1]])
  }
}
resu<-data.frame(resLab, resu)
temp<-sort.list(resu[,6],decreasing = TRUE)
resu<-resu[temp,]
rownames(resu)<-1:dim(resu)[1]
colnames(resu)<-c("lab",col.lab)
resu
}
```

Compute the integrative ensemble for corpus A

```
#without weight
wb<-c(1,1,1,1,1); wf<-c(1,1,1,1,1,1)
interEnsemb(BERT.A,AdaRF.A, wb,wf)->InterEnsemble.A
temp<-sort.list(InterEnsemble.A[,6],decreasing = TRUE)
InterEnsemble.A<-InterEnsemble.A[temp,]
write.csv(InterEnsemble.A,"c:/ensemble/InterEnsemble.A.csv")
corpusA[[9]]<-InterEnsemble.A[1:50,6]
kable(head(InterEnsemble.A,10),format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ A,S | 3,5 }	1.0000000	0.000000	1.00	0.0000000	1.0000000	0.0000000
{ A,S | 3,6 }	1.0000000	0.000000	1.00	0.0000000	1.0000000	0.0000000
{ A,S | 4,6 }	1.0000000	0.000000	1.00	0.0000000	1.0000000	0.0000000
{ A,S | 5,6 }	1.0000000	0.000000	1.00	0.0000000	1.0000000	0.0000000
{ A,S | 3,5,6 }	1.0000000	0.000000	1.00	0.0000000	1.0000000	0.0000000
{ A,S | 4,5,6 }	1.0000000	0.000000	1.00	0.0000000	1.0000000	0.0000000
{ T,A | 3,6 }	0.9909091	0.028748	0.99	0.0316228	0.9899749	0.0211677
{ T,A | 1,3,6 }	0.9909091	0.028748	0.99	0.0316228	0.9899749	0.0211677

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ T,A | 3,4,6 }	0.9909091	0.028748	0.99	0.0316228	0.9899749	0.0211677
{ T,A | 3,5,6 }	0.9909091	0.028748	0.99	0.0316228	0.9899749	0.0211677

```
#Weighted ensemble
wb<-singBERT.A[,6]
wf<-FeaClas.A[,6]
interEnsemb(BERT.A,AdaRF.A, wb,wf)->InterEnsemble.A.W
temp<-sort.list(InterEnsemble.A.W[,6],decreasing = TRUE)
InterEnsemble.A.W<-InterEnsemble.A.W[temp,]
write.csv(InterEnsemble.A.W,"c:/ensemble/InterEnsemble.A.W.csv")
corpusA[[10]]<-InterEnsemble.A.W[1:50,6]

kable(head(InterEnsemble.A.W),format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ A,S | 1,3 }	1	0	1	0	1	0
{ A,S | 2,3 }	1	0	1	0	1	0
{ A,S | 3,4 }	1	0	1	0	1	0
{ A,S | 3,5 }	1	0	1	0	1	0

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ A,S | 3,6 }	1	0	1	0	1	0
{ A,S | 1,2,3 }	1	0	1	0	1	0

Compute the integrative ensemble for corpus B

```
#Without weight
wb<-c(1,1,1,1,1);wf<-c(1,1,1,1,1,1)
interEnsemb(BERT.B,AdaRF.B, wb,wf)->InterEnsemb.B
temp<-sort.list(InterEnsemb.B[,6],decreasing = TRUE)
InterEnsemble.B<-InterEnsemb.B[temp,]
corpusB[[9]]<-InterEnsemb.B[1:50,6]
write.csv(InterEnsemb.B,"C:/ensemble/InterEnsemb.B.csv")
kable(head(InterEnsemb.B,10),format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ T,De | 1,2 }	0.9642424	0.0607876	0.96	0.0516398	0.9603577	0.0381913
{ T,AW | 1,2 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083
{ T,AW | 1,2,4 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083
{ T,AW,De | 1,2,4,6 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083

lab	Recall_mean	Recall_sd	Preci_mean	Preci_sd	F1_mean	F1_sd
{ T,AW,De | 1,2,4,5,6 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083
{ AW,De,S | 1,2 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083
{ AW,De,S | 1,6 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083
{ AW,De,S | 1,2,4 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083
{ AW,De,S | 1,2,5 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083
{ AW,De,S | 1,2,6 }	0.9651515	0.0597797	0.96	0.0516398	0.9603326	0.030083

```
#Weighted ensemble
wb<-singBERT.B[,6]; wf<-FeaClas.B[,6]
interEnsemb(BERT.B,AdaRF.B, wb,wf)->InterEnsemb.B.W
temp<-sort.list(InterEnsemb.B.W[,6],decreasing = TRUE)
InterEnsemble.A.W<-InterEnsemb.B.W[temp,]
corpusB[[10]]<-InterEnsemb.B.W[1:50,6]
write.csv(InterEnsemb.B.W,"c:/ensemble/InterEnsemb.B.W.csv")
```

Create table 4

```
max_vector1<-numeric()
mean_vector1<-numeric()
max_vector2<-numeric()
mean_vector2<-numeric()
sd_vector1<-numeric()
sd_vector2<-numeric()
for(i in 1:10){
  max_vector1<-c(max_vector1,max(corpusA[[i]]))
  max_vector2<-c(max_vector2,max(corpusB[[i]]))
  mean_vector1<-c(mean_vector1,mean(corpusA[[i]]))
  mean_vector2<-c(mean_vector2,mean(corpusB[[i]]))
  sd_vector1<-c(sd_vector1,sd(corpusA[[i]]))
  sd_vector2<-c(sd_vector2,sd(corpusB[[i]]))
}
# Set data name
labs<-c("BERTs",
        "Ensemble BERTs",
        "Weighted Ensemble BERTs",
        "Feature",
        "Ensemble Features & Classifiers",
        "Weighted Ensemble Features & Classifiers",
        "Ensemble One Feature & Classifiers and BERTs [39]",
        "Ensemble One BETR and Features & Classifiers [40]",
        "Intergreted Ensemble",
        "Intergreted Weighted Ensemble"
    )

table4<-data.frame(method=labs,
mean=mean_vector1,sd=sd_vector1,Max=max_vector1,
mean=mean_vector2,sd=sd_vector2,Max=max_vector2
)

table4[,-1]<-round(table4[,-1],3)
write.csv(table4,"C:/ensemble/table4.csv")
kable(table4,format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

method	mean	sd	Max	mean.1	sd.1	Max.1
BERTs	0.775	0.181	0.970	0.770	0.055	0.823

method	mean	sd	Max	mean.1	sd.1	Max.1
Ensemble BERTs	0.911	0.091	0.990	0.861	0.030	0.902
Weighted Ensemble BERTs	0.910	0.096	0.980	0.861	0.029	0.899
Feature	0.761	0.036	0.810	0.730	0.067	0.800
Ensemble Features & Classifiers	0.852	0.033	0.912	0.817	0.039	0.889
Weighted Ensemble Features & Classifiers	0.851	0.034	0.912	0.828	0.033	0.889
Ensemble One Feature & Classifiers and BERTs [39]	0.953	0.023	0.980	0.919	0.027	0.939
Ensemble One BETR and Features & Classifiers [40]	0.839	0.127	0.990	0.814	0.052	0.901
Intergreted Ensemble	0.991	0.003	1.000	0.957	0.005	0.960
Intergreted Weighted Ensemble	1.000	0.000	1.000	0.954	0.005	0.960

Create table 5

```
table5A<-data.frame(
  comBERT=ensemBERT.A[1:10,1],cB_F1=round(ensemBERT.A[1:10,6],3),
  comFea=ensembFeature.A[1:10,1],cF_F1=round(ensembFeature.A[1:10,6],3),
  intEns=InterEnsemble.A[1:10,1],cF_F1=round(InterEnsemble.A[1:10,6],3)

                    )
kable(table5A,format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

comBERT	cB_F1	comFea	cF_F1	intEns	cF_F1.1
{ A , S }	0.99	{ 1 , 3 , 5 }	0.912	{ A,S | 3,5 }	1.00
{ T , A }	0.98	{ 1 , 3 , 4 , 5 }	0.912	{ A,S | 3,6 }	1.00

comBERT	cB_F1	comFea	cF_F1	intEns	cF_F1.1
{ A , AW }	0.98	{ 1 , 3 , 4 , 5 , 6 }	0.912	{ A,S | 4,6 }	1.00
{ A , De }	0.98	{ 1 , 3 , 5 , 6 }	0.901	{ A,S | 5,6 }	1.00
{ AW , S }	0.98	{ 1 , 3 , 4 }	0.893	{ A,S | 3,5,6 }	1.00
{ A , AW , De }	0.98	{ 1 , 2 , 3 , 6 }	0.883	{ A,S | 4,5,6 }	1.00
{ T , A , AW }	0.97	{ 1 , 2 , 3 , 4 , 6 }	0.883	{ T,A | 3,6 }	0.99
{ A , AW , S }	0.97	{ 1 , 2 , 3 , 5 , 6 }	0.883	{ T,A | 1,3,6 }	0.99
{ T , A , AW , De }	0.97	{ 1 , 2 , 3 , 4 , 5 , 6 }	0.883	{ T,A | 3,4,6 }	0.99
{ T , A , AW , S }	0.97	{ 1 , 3 , 6 }	0.882	{ T,A | 3,5,6 }	0.99

```
table5B<-data.frame(
  comBERT=ensemBERT.B[1:10,1],cB_F1=round(ensemBERT.B[1:10,6],3),
  comFea=ensembFeature.B[1:10,1],cF_F1=round(ensembFeature.B[1:10,6],3),
  intEns=InterEnsemble.B[1:10,1],cF_F1=round(InterEnsemble.B[1:10,6],3)
  )
kable(table5B,format = "html") %>%
  kable_styling(full_width = FALSE, position = "left")
```

comBERT	cB_F1	comFea	cF_F1	intEns	cF_F1.1
{ T , A , AW , De }	0.902	{ 1 , 2 , 6 }	0.889	{ T,De | 1,2 }	0.96
{ A , AW , De , S }	0.901	{ 1 , 2 , 4 , 6 }	0.889	{ T,AW | 1,2 }	0.96
{ T , A , De , S }	0.894	{ 1 , 2 , 5 , 6 }	0.889	{ T,AW | 1,2,4 }	0.96
{T,A,AW,De,S}	0.891	{ 1 , 2 , 4 , 5 , 6 }	0.889	{ T,AW,De | 1,2,4,6 }	0.96

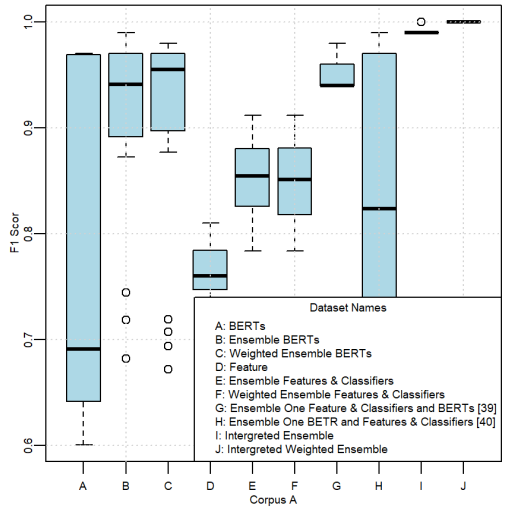
comBERT	cB_F1	comFea	cF_F1	intEns	cF_F1.1
{ T , AW , De , S }	0.890	{ 1 , 2 , 4 }	0.869	{ T,AW,De | 1,2,4,5,6 }	0.96
{ T , A , AW }	0.882	{ 4 , 5 , 6 }	0.866	{ AW,De,S | 1,2 }	0.96
{ T , AW , De }	0.882	{ 1 , 4 , 5 , 6 }	0.859	{ AW,De,S | 1,6 }	0.96
{ T , A , AW , S }	0.881	{ 1 , 2 , 5 }	0.858	{ AW,De,S | 1,2,4 }	0.96
{ A , AW , De }	0.880	{ 1 , 2 , 4 , 5 }	0.858	{ AW,De,S | 1,2,5 }	0.96
{ AW , De }	0.880	{ 1 , 2 , 3 , 4 }	0.855	{ AW,De,S | 1,2,6 }	0.96

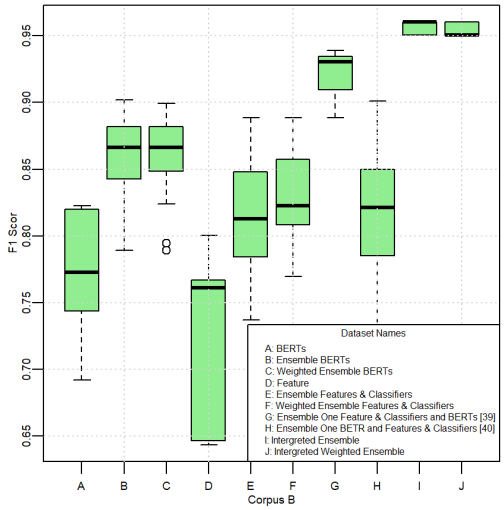
```
table5<-rbind(table5A,table5B)
#kable(table5, format = "html") %>%
#  kable_styling(full_width = FALSE, position = "left")
write.csv(table5,"c:/ensemble/table5.csv")
```

Draw Figure 2

For Corpus A

```
names(corpusA)<-labs
# Set horizontal axis labels
x_labels <- LETTERS[1:10]

# Set legend label
legend_labels <- paste(x_labels, names(corpusA), sep = ": ")

# Draw Box-plot of Corpus A
par(mfrow=c(1,2),cex.lab = 0.6, cex.axis = 0.6,
  mgp = c(1, 0.4, 0),mai = c(1, 1, 1, 1))
  boxplot(corpusA, col = c("lightblue"),names = x_labels, main = "", xlab =
"Corpus A", ylab = "F1 Scor", cex.axis = 0.6, cex.lab = 0.6)

# Add grid lines and legend
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted")
legend("bottomright", legend = legend_labels, title = "Dataset Names", cex =
0.6)

# Draw Box-plot of Corpus B
# Set data name
names(corpusB)<-labs
# Set horizontal axis labels
x_labels <- LETTERS[1:10]

# Set legend label
legend_labels <- paste(x_labels, names(corpusB), sep = ": ")

# Draw Box-plot of Corpus B
par(mgp = c(1, 0.4, 0),mai = c(1, 1, 1, 1))
boxplot(corpusB, col = "lightgreen",names = x_labels, main = "", xlab = "Cor
pus B", ylab = "F1 Scor", cex.axis = 0.6, cex.lab = 0.6)

# Add grid lines and legend
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted")
legend("bottomright", legend = legend_labels, title = "Dataset Names", cex =
0.5)
```


F1 Scor
1.0
0.9
0.8
0.7
0.6
A
B
C
D
E
F
G
H
I
J
Corpus A
Dataset Names
A: BERTs
B: Ensemble BERTs
C: Weighted Ensemble BERTs
D: Feature
E: Ensemble Features & Classifiers
F: Weighted Ensemble Features & Classifiers
G: Ensemble One Feature & Classifiers and BERTs [39]
H: Ensemble One BETR and Features & Classifiers [40]
I: Intergreted Ensemble
J: Intergreted Weighted Ensemble


F1 Scor
0.95
0.90
0.85
0.80
0.75
0.70
0.65
A
B
C
D
E
F
G
H
I
J
Corpus B
Dataset Names
A: BERTs
B: Ensemble BERTs
C: Weighted Ensemble BERTs
D: Feature
E: Ensemble Features & Classifiers
F: Weighted Ensemble Features & Classifiers
G: Ensemble One Feature & Classifiers and BERTs [39]
H: Ensemble One BETR and Features & Classifiers [40]
I: Intergreted Ensemble
J: Intergreted Weighted Ensemble