L1 Identification: A Clustering and Classification Analysis

Alan Zhu

13 December, 2021

Abstract

L1 transfer is a linguistic phenomenon where some linguistic features from the L1 are applied to the L2, and its effect is specific to different L1s. Identifying the L1 of L2 learners may be useful to provide specific guidance in L2 learning and writing in L2. In this article, I analyze English texts written by L2 learners with French, German, Cantonese, and Japanese L1s as well as texts written by native English speakers from the BAWE corpus to determine the effects of L1 transfer. I use hierarchical agglomerative clustering and k-means clustering based on the part-of-speech frequencies of each text to examine the similarities and differences in writings produced by writers of different L1s. I also build a multinomial logistic regression model with Lasso regularization and a random forest model to predict the L1, achieving reasonable accuracy.

1 Introduction

English is the most spoken language in the world. However, most people speaking English use it as their second language. In fact, out of approximately 1.5 billion English speakers, only 400 million are native speakers (Breene 2019). For most of us who had to learn English as a second language, our first languages play a role in this L2 acquisition process. L1 transfer is a linguistic phenomenon where certain linguistic features from the native language of a L2 learner is applied to the new language they are learning. In writing specifically, L2 learners may use a variety of L1-based composing strategies (Karim and Nassaji 2013) to help them write in their second language. Since the effects of L1 transfer are specific to different L1s, it may be useful to identify the L1s of writers in order to facilitate the use of L1-based strategies and provide L1-specific guidance in L2 writing.

In this study, I analyze English texts written by native and non-native speakers by using clustering analysis based on the part-of-speech frequencies of each document, and build statistical models to predict the L1 from texts. I seek to address the following questions:

- Are there differences and similarities in English texts produced by writers with different L1s? If so, which L1s are more similar?
- Is it possible to identify the writer's L1 solely based on the written text? If so, what is the best model for predicting the L1?

2 Data & Methods

The corpus used for this experiment is the BAWE (British Academic Written English) corpus, which consists of proficient university-level student writing in the U.K. from 2004 to 2007, with writing from a variety of disciplinary areas, ranging from Engineering and Biological Sciences to Business and Law. The majority of texts are written by native English speakers, but there are also texts written by non-native speakers. **Table 1** provides a summary of the corpus, as well as the average number of tokens per file. **Table 2** shows the top 10 most common L1 and their frequencies in the corpus. In this analysis, I chose to analyze specifically the non-native L1s of French, German, Chinese Cantonese, and Japanese. I chose these four languages because I want to examine a diverse mix of languages, with French and German as European languages and Cantonese and Japanese as Asian languages. The choice of Cantonese as opposed to Mandarin or unspecified Chinese is to ensure enough documents in this category while keeping a smaller deviation within the same group.

Table 1: BAWE Corpus Summary

Files	Tokens	Tokens per File
2761	6613671	2395.39

Table 2: Counts and Proportions of Texts in Each L1

L1	Count	Proportion (%)
English	1953	70.74
Chinese unspecified	153	5.54
Chinese Cantonese	66	2.39
French	60	2.17
German	57	2.06
Greek	46	1.67
Japanese	40	1.45
Polish	34	1.23
Hindi	33	1.20
Chinese Mandarin	26	0.94

To answer the first research question, I used clustering to analyze similarities and differences in texts with different L1, particularly hierarchical agglomerative clustering and k-means clustering. In hierarchical agglomerative clustering, individual data points (in this case, documents) are joined step-by-step with their closest ones until all data points are joined in a single cluster. In k-means clustering, k random point are set as initial centers of their clusters, and each point is assigned to their closest centers, and the centers of each cluster are updated as the mean of the points in the cluster, and the points and clusters are iteratively updated until an optimal clustering is found. The purpose of clustering is to understand the differences and similarities between texts, and if these differences and similarities are related to the difference in L1. Furthermore, the results of the clustering can be visualized to help me interpret if any non-English L1s are similar to English or to another language. I used clustering rather than token-based frequency and keyness analysis since the corpus contains writings on varying disciplines, and the distribution of single tokens could be more related to the topic of the writing rather than the L1 of the writer.

To prepare for clustering analysis, I tagged and combined the universal and EWT (Universal Dependencies English Web Treebank) part-of-speech of each token using the `R` package `udpipe`. Then I created a document-feature matrix using the relative frequencies of each part-of-speech in each document. In doing so, each document is represented by a row in the matrix. Instead of representing each document as a vector of token counts, this representation has a lot fewer dimensions and can be better generalized across texts. I then scaled and standardized the matrix. I also removed features with zero variance in the matrix, as they would not provide useful information in distinguishing between different documents and L1s.

For hierarchical clustering, I randomly subsetted the matrix with 10 documents from each L1 (including English). I chose 10 documents each to have enough of each L1 while not overcrowding the dendrogram. I then carried out hierarchical agglomerative clustering using Ward's Minimum Variance Clustering method, and plotted the dendrogram of the clustering along with the original labels (L1s). For k-means clustering, I subsetted the matrix with 40 documents from each L1, and chose k=4. I plotted the result of the clustering using the first two dimensions from the principle component analysis on the document-feature matrix. To investigate which L1s are the most similar to English, I aggregated the data into 5 points by averaging the frequency distribution across all texts with the same L1, and carried out hierarchical clustering with Ward's method, and plotted the dendrogram (Granger 2017).

Based on the results of the clustering approaches, I attempted to answer the second question by building two statistical models. The first model I built is a multinomial logistic regression with Lasso regularization. There are 58 features (part-of-speech tags), and Lasso regularization penalizes the model for using too many features and results in fewer features being selected in the final model. With fewer features, the variance of the model is smaller and it is less likely that the model will overfit to the training data. The second model is a random forest classification model. Random forest is a non-parametric model that uses an ensemble of decision trees, with a random sample of data and features in each decision tree. The randomness makes the model more robust and reduces the chance of overfitting.

I trained the two models on 30 documents for each non-native L1s, and evaluated the models on another 10 documents for the four L1s. While there are not the same number of documents for each L1, I chose to keep the number of instances for each class the same as to avoid using imbalanced data for training the random forest model, which could result in the classifier favoring particular classes over the others. I then evaluated and compared the two models by producing their confusion matrices, and their accuracy and F-1 scores.

3 Cluster Analysis

In the Hierarchical Agglomerative Clustering, 10 documents are randomly sampled from writing with English, French, German, Cantonese and Japanese L1s. **Figure 1** shows the dendrogram produced from the clustering. The x-axis shows the L1 for each text, with `E` and black representing native English, `F` and blue representing French, `G` and gray representing German, `C` and orange representing Cantonese, and `J` representing Japanese. The y-axis shows the distance between data points and clusters. In the lower left corner of the graph, there are several texts written by native French speakers clustered next to each other. In the middle left, three writings with German L1 are clustered together. Towards the right, four writings with Japanese L1 are clustered together. Points clustered close together means that the distance between these points are smaller compared to the distance to other points. Therefore, text in the French, German, and Japanese clusters follow a similar pattern in terms of part-of-speech frequencies, respectively. While some noticeable structures are found, namely these three clusters, the writings by native English speakers and by native Cantonese speakers are not clustered closely together.

To examine the structure and clustering further, I used 40 samples for each L1 in k-means clustering. To decide the value of k (the number of clusters), I plotted the value of k against the total within cluster sums of squares, and the "elbow" in the plot showed `k = 4`. I then carried out k-means clustering with `k = 4`. The result of the clustering is summarized in **Table 3**. To visualize the clustering result, I plotted the clusters using the first two principal components from PCA. The variance explained by the first principal component is 15.4%, and the second principal component 8.6%, in total 24%. **Figure 2** visualizes the clustering. Cluster 1 contains mostly texts written by native French speakers, with some English L1 texts, and a small number of texts with other L1s. Cluster 2 contains the highest number of texts (73), with a mix of native Cantonese, Japanese, and German speakers with a small number of texts with other L1s. Cluster 3 are mainly texts written by native Cantonese speakers, with some English texts. Cluster 4 are mostly texts written by native German and Japanese speakers, with some English texts. French L1 texts are most well-clustered, with most texts lying in Cluster 1. This suggests perhaps native French speakers have a more distinct writing tendency and pattern compared to writers with other L1s. The other L1s are mainly split in two clusters each, with Cantonese split in Cluster 2 and 3, German and Japanese split in Cluster 2 and 4. Although both German and Japanese L1 texts are split in Cluster 2 and 4, from the plot there are still some differences in the location of these points, and it does not necessarily mean that German and Japanese L1 texts are similar. Native speaker texts are split across all 4 clusters, which may be explained by the higher variance in writing style due to the large number of texts in the corpus.

Table 3: K-means Clustering Summary

	Chinese Cantonese	English	French	German	Japanese	Total	Within Sum of Squares
Cluster 1	4	10	18	4	1	37	1302.161
Cluster 2	17	6	8	20	22	73	2301.787

	Chinese Cantonese	English	French	German	Japanese	Total	Within Sum of Squares
Cluster 3	15	11	6	0	3	35	1648.037
Cluster 4	4	13	8	16	14	55	2847.990

The results from both clustering approaches show that there exist some structures and differences in patterns in the distributions of part-of-speech for texts in each L1. This motivated me to build statistical models to predict the L1 of writings based on these distributions.

Figure 3 shows the dendrogram of hierarchical clustering on the five aggregated data points. On average, writings by native German speakers are the most similar to writings by native English speakers. A possible explanation is that German and English are more closely related than other languages, since both are Germanic languages. In the other three languages, French L1 writings are more similar to German L1 and native writings, while Cantonese L1 writings are less similar, and Japanese L1 writings are the most different. An explanation is that French is in the same Indo-European language family as English and German, while Cantonese and Japanese are in completely different language families of Sino-Tibetan and Japonic respectively.

4 Classification Modeling

To predict the L1 of a text given its part-of-speech frequency distribution, I trained two statistical models. First I trained a multinomial logistic regression model with Lasso regularization. I ran cross-validation to determine the penalization term λ for Lasso, and then I chose $\lambda = 0.04464$, which is the largest λ value such that the cross validation error is within 1 standard error of the minimum. With the penalization, only 24 features out of the 58 features remained and are used in the model. I also trained a random forest model to predict the L1 using the same training data. I evaluated both models using the testing data which included 10 texts from each non-native L1s.

Table 4 and **Table 5** show the confusion matrix of the two models evaluated on the testing data, with the column corresponding to the predicted L1 and the row corresponding to the actual L1. **Table 6** shows the accuracies of the two models, along with the class and average F1 scores. The accuracies of both models are above 25%, which means that both models perform much better than random chance. This confirms that the L1 of the author of a text can be determined based on its part-of-speech distribution. The random forest model has a higher F1 score on predicting Cantonese, French, and Japanese than the Lasso model, which means that the random forest model is better at predicting these L1s, and the Lasso model only outperformed the random forest model on predicting German. The random forest model has an average F1 score of 72.78% and accuracy of 72.5%, which is higher than the 63.62% average F1 score and 65% accuracy of the Lasso model. This suggests that a random forest model may be better at predicting L1 from texts than multinomial logistic regression with Lasso.

Table 4: Confusion Matrix for Multinomial Lasso Regression

	Chinese Cantonese	French	German	Japanese
Chinese Cantonese	5	0	1	0
French	1	4	0	0
German	2	3	8	1
Japanese	2	3	1	9

Table 5: Confusion Matrix for Random Forest

	Chinese Cantonese	French	German	Japanese
Chinese Cantonese	7	1	0	0
French	0	8	3	0

	Chinese Cantonese	French	German	Japanese
German	2	1	6	2
Japanese	1	0	1	8

Table 6: Class and Average F1 score and Accuracy of the Models

	Multinomial Lasso	Random Forest
Chinese Cantonese	62.50	77.78
French	53.33	76.19
German	66.67	57.14
Japanese	72.00	80.00
Average F1	63.62	72.78
Accuracy	65.00	72.50

5 Conclusion

The effects of L1 transfer can be observed in the writings of L2 English learners. In my analysis, I used the frequency distribution of part-of-speech tags of English texts to represent each document, and used hierarchical agglomerative clustering and k-means clustering to examine the differences and similarities between texts with different L1s. The results from both clustering approaches show that there are observable differences in the frequency distributions from texts with different L1s. Hierarchical clustering on aggregated data for each L1 shows that the writing of native German speakers are most similar to native English speakers, followed by French, Cantonese, and Japanese.

The L1 of the non-native English writers can be predicted with reasonable accuracy. Using a training set of 120 samples and a testing set of 40 samples, the multinomial logistic regression with Lasso regularization achieved 65% accuracy, and the random forest model had 72.5% accuracy. The random forest model appears to be the better model to predict the L1 of writers.

It is important to note the limitations of this study. The BAWE corpus used in this study contains a relatively small number of non-native English writing. The four L1s in this study have less than 100 texts each. Across these texts, the writings have a wide range of topics and the writers have varying degrees of proficiency in English. Therefore, the variation in the documents should also be attributed to the difference in discipline and proficiency in addition to the difference in L1. Furthermore, the models trained and evaluated on a smaller sample size may not be as reliable as models trained and tested on a larger sample.

Despite these limitations, this study presents promising directions for future research. For instance, comparing texts by writers native in other widely-spoken languages such as Spanish, Portuguese, Arabic, in addition to the four languages in this analysis. Or, investigating the differences between bilinguals that speak English and another language and L1 speakers in that language. In addition to the part-of-speech frequencies, the frequencies of particular words or phrases can also be used to compare texts and predict L1. More statistical models can be tested, and the important features in these models can be investigated using traditional linguistic methods and case studies to determine the specific effects of L1 transfers.

6 References

Breene, Keith. 2019. "Which Countries Are Best at English as a Second Language?" *World Economic Forum.* https://www.weforum.org/agenda/2019/11/countries-that-speak-english-as-a-second-language/.

Granger, Sylviane. 2017. "Academic Phraseology: A Key Ingredient in Successful L2 Academic Literacy." *Oslo Studies in Language* 9 (December): 9–27. https://doi.org/10.5617/osla.5844.

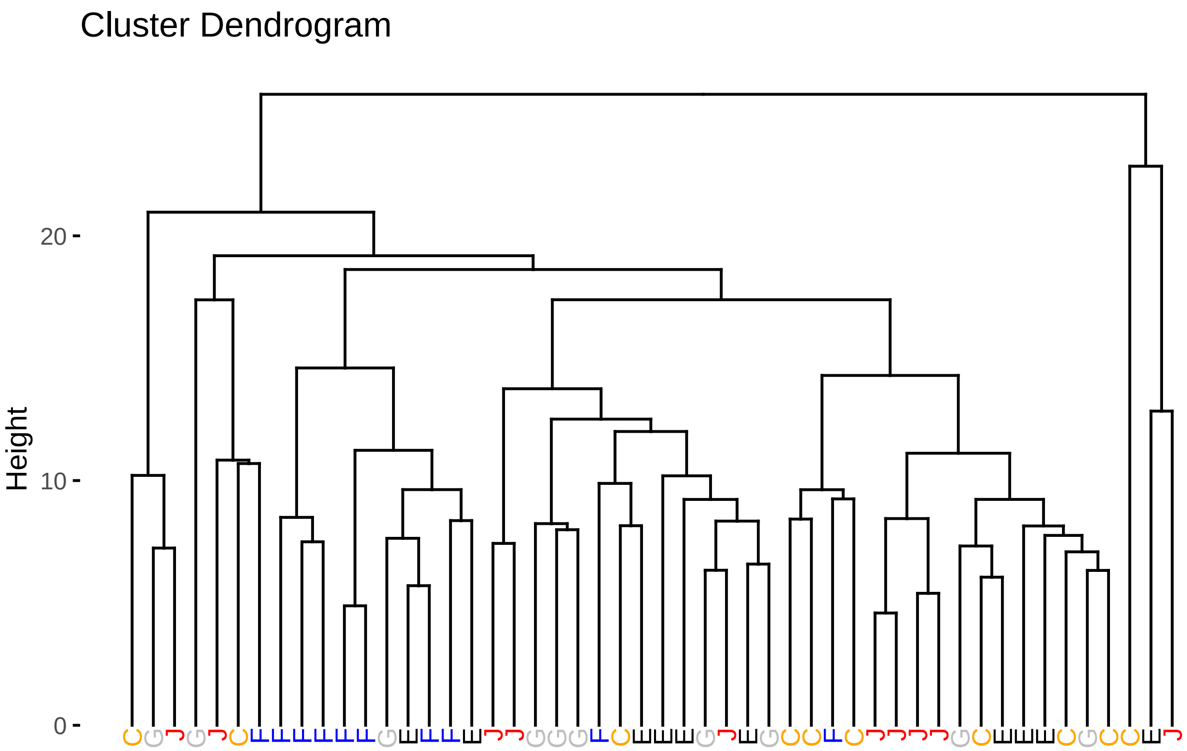

Figure 1: Cluster Dendrogram of Hierarchical Clustering

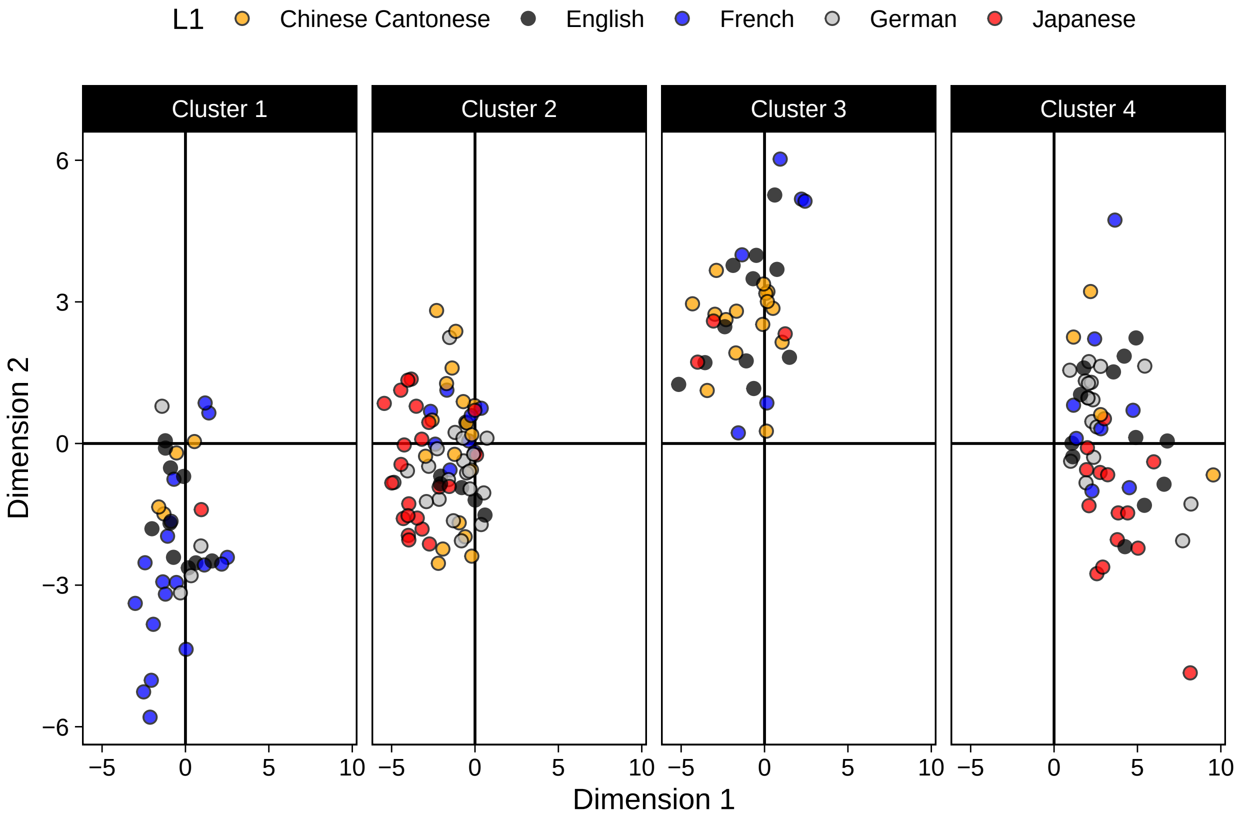

Figure 2: K-means Clustering Plotted on the First Two Principal Components

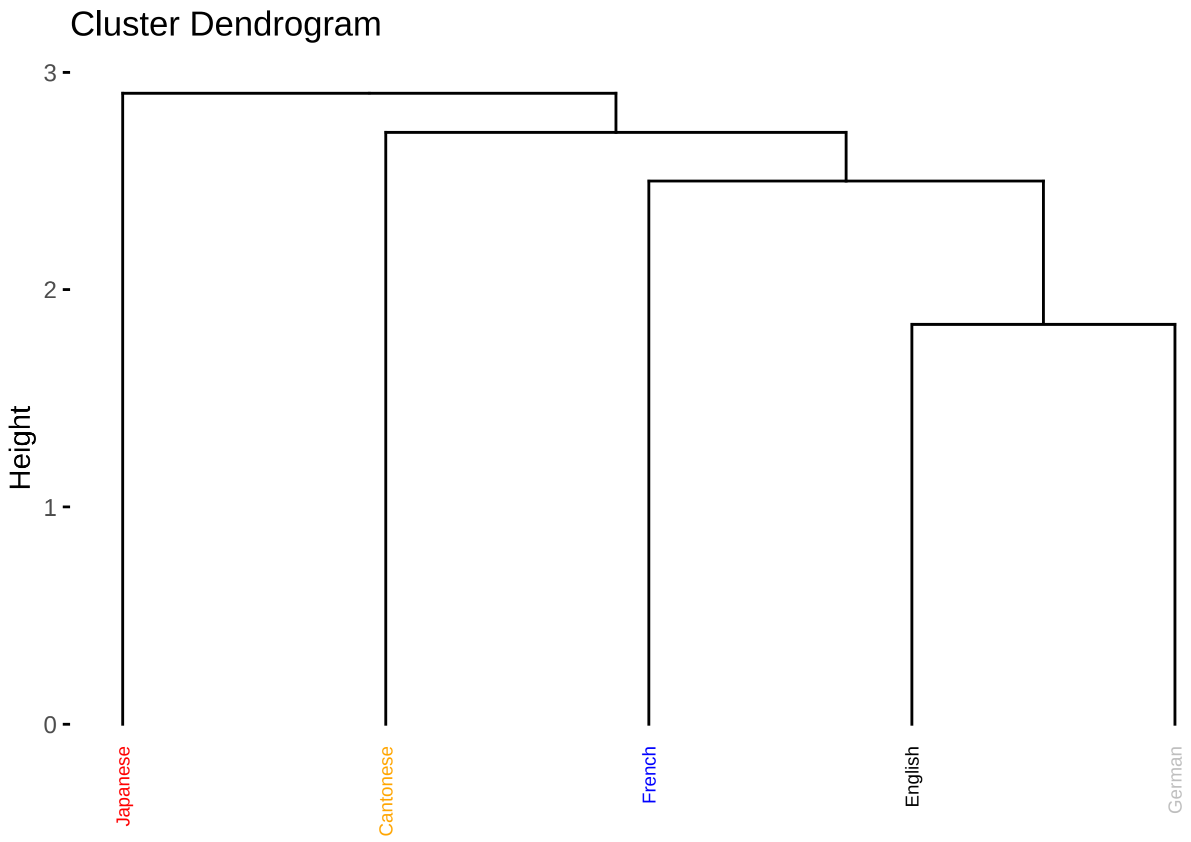

Figure 3: Cluster Dendrogram of Hierarchical Clustering Using Aggregated Data for each L1

Karim, Khaled, and Hossein Nassaji. 2013. “First Language Transfer in Second Language Writing: An Examination of Current Research.” *Iranian Journal of Language Teaching Research* 1 (January): 117–34.

7 Code Appendix

```r
# Library Imports
library(readtext)
library(readxl)
library(tidyverse)
library(quanteda)
library(quanteda.extras)
library(udpipe)
library(dendextend)
library(factoextra)
library(glmnet)
library(randomForest)
library(caret)


# Reading Corpus
files_list <- list.files("bawe/CORPUS_TXT",
                         full.names = T, pattern = "*.txt", recursive = T)
bawe_text <- readtext(files_list)
bawe_corpus <- corpus(bawe_text)
bawe_tkns <- tokens(bawe_corpus, what="word", remove_punct=T, remove_symbols=T,
                      remove_numbers=T, remove_url=T)
bawe_dfm <- dfm(bawe_tkns)


# Reading Meta File
bawe.meta <- read_excel("bawe/documentation/BAWE.xls")
names(bawe.meta) <- make.names(names(bawe.meta))


# Corpus Summary
l1.summary <- bawe.meta %>%
  group_by(L1) %>%
  dplyr::summarize(count = n()) %>%
  arrange(desc(count)) %>%
  mutate(proportion = count/sum(count)*100)
bawe.summary <- data.frame(files = ndoc(bawe_corpus),
                           tokens = sum(ntoken(bawe_tkns))) %>%
  mutate(token.p.file = tokens/files)


# POS Tagging and creating DFM
ud_model <- udpipe_load_model("english-ewt-ud-2.5-191206.udpipe")
annotation <- udpipe_annotate(ud_model, x = bawe_text$text,
                              doc_id = bawe_text$doc_id, parser = "none")
anno_edit <- annotation %>%
  as_tibble() %>%
  unite("upos", upos:xpos)
sub_tokens <- split(anno_edit$upos, anno_edit$doc_id)
sub_tokens <- as.tokens(sub_tokens)
sub_tokens <- tokens_remove(sub_tokens, "^punct_\\S+", valuetype = "regex")
sub_tokens <- tokens_remove(sub_tokens, "^sym_\\S+", valuetype = "regex")
sub_tokens <- tokens_remove(sub_tokens, "^x_\\S+", valuetype = "regex")
sub_dfm <- sub_tokens %>%
  dfm() %>%
  dfm_weight(scheme = "prop") %>%
  convert(to = "data.frame")
sub_dfm <- sub_dfm %>%
  column_to_rownames("doc_id") %>%
  dplyr::select(order(colnames(.)))
sub_dfm <- sub_dfm %>%
  scale() %>%
  data.frame()


# Hierarchical Agglomerative Clustering
set.seed(42)
l1s <- c("English", "Chinese Cantonese", "Japanese", "French", "German")
bawe_l1 <- bawe.meta %>%
  filter(L1 %in% l1s) %>%
  group_by(L1) %>%
  sample_n(10) %>%
  arrange(id)
sub_dfm_l1 <- sub_dfm[bawe_l1$id,]
zero_var_cols <- which(apply(sub_dfm_l1, 2, var)==0)
sub_dfm_new <- sub_dfm_l1[,-zero_var_cols]
l1_abbr <- plyr::mapvalues(bawe_l1$L1, l1s, c("E", "C", "J", "F", "G"))
l1_colors <- plyr::mapvalues(bawe_l1$L1, l1s, c("black", "orange", "red", "blue", "grey"))
dist.mat <- dist(sub_dfm_new, method = "euclidean")
hc <- hclust(dist.mat, method = "ward.D2")
dend <- hc %>%
  as.dendrogram() %>%
  set("labels", l1_abbr, order_value = T) %>%
  set("labels_colors", l1_colors, order_value = T)
hc$labels <- l1_abbr
hc.dend <- fviz_dend(hc, cex = 0.7, lwd=0.5, show_labels=T,
                     label_cols = plyr::mapvalues(bawe_l1$L1[hc$order],
                           l1s, c("black", "orange", "red", "blue", "grey")),
                     type="rectangle")


# Hierarchical Clustering with Aggregated Data
bawe_l1_all <- bawe.meta %>%
  filter(L1 %in% l1s) %>%
  arrange(id)
sub_dfm_l1_all <- sub_dfm[bawe_l1_all$id,]
sub_dfm_agg <- sub_dfm_l1_all %>%
  mutate(L1 = bawe_l1_all$L1) %>%
  group_by(L1) %>%
  summarize_all(mean) %>%
  column_to_rownames("L1")
dist.mat <- dist(sub_dfm_agg, method = "euclidean")
hc <- hclust(dist.mat, method = "ward.D2")
```

```r
dend <- hc %>% as.dendrogram()
hc$labels <- c("Cantonese", "English", "French", "German", "Japanese")
hc.dend2 <- fviz_dend(hc, cex = 0.5, lwd=0.5, show_labels=T,
                      label_cols =  plyr::mapvalues(row.names(sub_dfm_agg)[hc$order],
                            l1s, c("black", "orange", "red", "blue", "grey")),
                      type="rectangle")


# K-means Clustering with PCA
set.seed(42)
l1s <- c("English", "Chinese Cantonese", "Japanese", "French", "German")
bawe_l1 <- bawe.meta %>%
  filter(L1 %in% l1s) %>%
  group_by(L1) %>%
  sample_n(40) %>%
  arrange(id)
sub_dfm_l1 <- sub_dfm[bawe_l1$id,]
zero_var_cols <- which(apply(sub_dfm_l1, 2, var)==0)
sub_dfm_new <- sub_dfm_l1[,-zero_var_cols]
l1_abbr <- plyr::mapvalues(bawe_l1$L1, l1s, c("E", "C", "J", "F", "G"))
l1_colors <- plyr::mapvalues(bawe_l1$L1, l1s, c("black", "orange", "red", "blue", "grey"))
rownames(sub_dfm_new) <- paste(l1_abbr, 1:nrow(bawe_l1), sep="_")
km <- kmeans(sub_dfm_new, 4)
factoextra::fviz_cluster(km, data = sub_dfm_new)
km_pca <- prcomp(sub_dfm_new)
round(factoextra::get_eigenvalue(km_pca), 1) %>% head
coord_df <- data.frame(km_pca$x[,1:2]) %>%
  mutate(L1 = bawe_l1$L1) %>%
  mutate(Cluster = as.factor(paste0("Cluster ", km$cluster)))
kmeans.plot <- ggplot(coord_df) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  geom_point(aes(x = PC1, y = PC2, fill = L1), size = 2, shape = 21, alpha = .75) +
  scale_fill_manual(values = c("orange", "black", "blue", "grey", "red")) +
  xlab(paste0("Dimension 1")) +
  ylab("Dimension 2") +
  theme_linedraw() +
  theme(panel.grid.major.x = element_blank()) +
  theme(panel.grid.minor.x = element_blank()) +
  theme(panel.grid.major.y = element_blank()) +
  theme(panel.grid.minor.y = element_blank()) +
  theme(legend.position="top") +
  facet_grid(~Cluster)
df.km <- data.frame(L1 = bawe_l1$L1, cluster = km$cluster)
df.km <- as.data.frame.matrix(table(df.km$cluster, df.km$L1))
df.km$Size <- km$size
df.km$tss <- km$withinss
rownames(df.km) <- paste("Cluster", 1:4)


# Train-Test Split
set.seed(42)
sub_dfm_new <- sub_dfm_l1[,-zero_var_cols] %>% mutate(L1 = bawe_l1$L1)
```

```
rownames(sub_dfm_new) <- bawe_l1$id
train_idx <- bawe_l1 %>%
  group_by(L1) %>%
  sample_n(30)
train_dfm <- sub_dfm_new[train_idx$id,] %>%
  filter(L1 != "English") %>%
  select(L1, everything())
test_dfm <- sub_dfm_new[!(rownames(sub_dfm_new) %in% train_idx$id),] %>%
  filter(L1 != "English") %>%
  select(L1, everything())


# Multinomial Lasso Regression
cv_fit <- cv.glmnet(as.matrix(train_dfm[, -1]), train_dfm[, 1],
                    family = "multinomial",type.multinomial = "grouped")
plot(cv_fit)
lambda_min <- cv_fit$lambda.min
lambda_lse <- cv_fit$lambda.1se
lasso_pred <- predict(cv_fit, newx = as.matrix(test_dfm[,-1]), s = lambda_lse,
                      type="class")


# Random Forest
set.seed(1234)
rf.model <- randomForest(formula = as.factor(L1) ~ ., data = train_dfm, mtry=5)
print(rf.model)
rf.pred <- predict(rf.model, newdata=test_dfm[,-1], type="class")


# Evaluating Both Models
cm.lasso <- confusionMatrix(as.factor(as.character(lasso_pred)),
                                   as.factor(test_dfm$L1), mode="everything")
cm.rf <- confusionMatrix(as.factor(rf.pred), as.factor(test_dfm$L1),
                                mode="everything")
f1.df <- cbind(as.data.frame.matrix(cm.lasso$byClass)$F1,
               as.data.frame.matrix(cm.rf$byClass)$F1)
average.f1 <- colMeans(f1.df)
accuracy <- c(cm.lasso$overall["Accuracy"], cm.rf$overall["Accuracy"])
metrics.df <- rbind(metrics.df,average.f1,accuracy)*100
rownames(metrics.df) <- c("Chinese Cantonese", "French", "German",
                          "Japanese", "Average F1", "Accuracy")
colnames(metrics.df) <- c("Multinomial Lasso", "Random Forest")
```