2022. *Proc Ling Soc Amer* 7(1). 5247. https://doi.org/10.3765/plsa.v7i1.5247.

Sociolinguistically-aware computational models of Mandarin-English codeswitching

Irene Yi *

Abstract. Current research on computational modeling of codeswitching has focused on the use of syntactic constraints as model predictors (Li & Fung 2014; Li & Vu 2019). However, proposed syntactic constraints (Poplack 1978; Poplack 1980; Myers-Scotton 1993; Belazi et al. 1994) are largely based around Spanish-English codeswitching, and are violated repeatedly (and potentially systematically) by codeswitching involving other languages. Thus, a computational model trained on these syntactic constraints, when applied to codeswitching involving languages that are not Spanish-English, may not capture the naturalistic patterns of those languages in codeswitching contexts. This paper demonstrates the value of sociolinguistic factors as predictors in training a Classification and Regression Tree (CART) model on novel Mandarin-English codeswitch data, which come from 12 bilingual speakers of two different generations from Grand Rapids, Michigan. Participants also answered metalinguistic questions about their own language practices and attitudes and completed a written Language History Questionnaire (LHQ) (Li et al. 2020), which asked for self-evaluations of language habits (proficiency, immersion, and dominance in the two languages). LHQ responses were then quantified into numerical scores serving as sociolinguistic predictors in the CART model. The model, which highlighted that age, L2 Dominance, and L1 Immersion were among the top predictors, achieved an accuracy of 0.804 with the area under its ROC curve being 0.692. This is comparable to, if not more powerful than, previous computational studies (e.g. Li & Fung 2014) that trained models using only proposed syntactic constraints as predictors. This paper shows the importance of sociolinguistic factors in computational research previously focused on syntactic constraints; the intersection of these methodologies could improve a cross-linguistic *and* computational understanding of codeswitching patterns.

Keywords. CART; codeswitching; computational modeling; Mandarin-English bilinguals; sociolinguistics

1. Introduction. As language and technology become more and more intertwined, it is imperative that any techno-logical advances address the complex sociolinguistic reality of many communities. One example of this sociolinguistic complexity is codeswitching. The sociolinguistic and syntactic factors that influence codeswitching help to deepen our understanding of not only linguistic identity, but also language habits—and by extension, language technology.

* I would like to thank Professor Gašper Beguš and Professor Isaac Bleaman (both UC Berkeley), my advisors and supervisors for this project. This work would not have been possible without both of your guidance and help. I would additionally like to thank the participants of my study, as well as the pilot study participants who helped me hone my methodology. While she was not directly involved in this project during the time it was actively going on, I would also like to thank Professor Claire Bowern (Yale University) for everything in the past year; I would not be who or where I am without you, Claire, so thank you. Finally, I would like to thank my lovely family and friends for their support in every way, through every step of the process. To everyone mentioned here: I thank you all so much, with my whole heart.
Author: Irene Yi, Yale University (ireneyi@berkeley.edu).

The goal of this paper is to explore the language technologies of multilingual speech and train a Classification and Regression Tree (CART) model on codeswitched Mandarin-English speech, using features that are sociolinguistic in nature during model training to show the importance of considering sociolinguistic factors in computational linguistic research. A companion paper (Yi n.d.) explores two connected questions that were part of the larger research project this current paper is contextualized within: 1) investigating nuanced differences in linguistic identities and language attitudes of Chinese American bilingual speakers based on sociolinguistic factors (e.g. age, education level, language immersion, etc.), and 2) comparing newly collected codeswitched data against previously proposed syntactic constraints on codeswitching.

Previous literature on codeswitching language technologies, and multilingual language technologies at large, is still in its pioneering stages. The research that does exist on codeswitching language technologies are more limited in scope to the syntactic constraints proposed by linguists who have sought to understand codeswitching from a syntactic typology perspective. However, contemporary sociolinguists have found that processes of codeswitching are not only difficult to categorize syntactically, but also dependent on sociolinguistic nuances of identity, presentation, audience design, and topic of conversation. Prior studies in computational linguistics often trained linguistic models with proposed syntactic constraints in mind to generate what would be deemed a "grammatical" codeswitched sentence, whereas this paper trains the model on what has been uttered by Mandarin-English bilinguals in conversation. This kind of model with naturalistic data, viewed under the lens of sociolinguistic influences and not trained using the syntactic constraints that have been proposed in the literature, is a novel contribution to a more holistic and nuanced understanding of language habits, identity, and technology.

2. Background. Many different types of computational models have been developed and used for language processing—specifically for intra-sentential and even intra-word codeswitching. For this paper, I will be looking primarily at intra-sentential CS, rather than intra-word CS, though there are a handful of instances of the latter in my data as well. I will briefly describe a portion of tasks that have been completed by computational models already, and locate the gap of sociolinguistically-aware technology that the current paper attempts to fill.

Computational models have ventured into syntactic and semantic representations of CS, though, to the author's knowledge, there are no publicly available tools for multilingual POS tagging or multilingual vector spaces (for semantic analysis). Such models look at the syntactic constraints proposed by linguists and model or generate CS data based on the constraints. Bullock et al. (2018) look into the discourse surrounding the validity of the previously discussed Matrix Language Framework (MLF) (Myers-Scotton 1993), specifically as it relates to NLP tasks. Bullock et al. (2018) use several of the indexes mentioned in Khanuja et al. (2020) (e.g. M-index) against several codeswitched corpora, running a logistic regression model to identify the presence of a Matrix Language (ML). While the model was able to accurately predict the presence of a ML over half the time, Bullock et al. (2018) wrote that there was not much significant information regarding the patterns of the ML versus the Embedded Language (EL) that could be extracted from the results of the model.

Li and Fung (2014) use a weighted finite state transducer (WFST) framework to handle codeswitching recognition and parsing from a Functional Head Constraint (FHC) structure. By training a model with respect to a proposed systematic syntactic constraint, Li and Fung (2014) circumvent the problem of not having enough CS data to effectively train models on. They pose this approach as a way of combining bilingual data with a given syntactic (and thus, more

predictable) structure. To do this, Li and Fung (2014) first expand the WFST's search network using a translation model; then, they restrict the parsing to only the permissible paths under FHC. Lattice parsing (enabling sequential coupling) and partial parsing (for tight coupling between parsing and filtering) were both tested and compared. While Li and Fung's (2014) WFST under FHC is more robust because it successfully avoids making early, erroneous decisions on CS boundaries, it is only able to do so because of the restriction of FHC. As mentioned, FHC does not allow CS to happen between a functional head and its complement, which is a constraint that often gets violated in my data. Using FHC on their model, Li and Fung (2014) were able to achieve a precision, recall, and F-score of 0.68, 0.71, and 0.70, respectively. My model (described in a later section) yields comparable results with sociolinguistic factors as predictors, rather than FHC.

The gap here, then, is that there are no non-syntactic and non-constraint-based models of CS. There is no research into how computationally modeling sociolinguistic features (like age, education level, language attitudes, L1/L2 proficiency, language habits, level of balanced bilingualism, etc.) affect CS. Thus, this paper will take these sociolinguistic factors into account, rather than syntactic ones like POS tags, and make them predictors for my model.

3. Methods and data. Novel data were collected through sociolinguistic interviews that elicited Mandarin-English codeswitched speech. These interviews were conducted remotely over Zoom due to the COVID-19 pandemic, and the procedure was approved by the Institutional Review Board (IRB) before any of the data collection process began. In order to elicit codeswitched speech, each interview included the same list of questions that were themselves asked using codeswitching between Mandarin and English. Zoom calls were recorded with participants' consent, as well as IRB approval.

Data were collected from a total of 12 participants, six of whom were in the older age category (>= 45 years old), and six of whom were in the younger one (20-30 years old). This line dividing the age group was drawn to correlate with their immigration generation. The six participants in the older age group all immigrated to the US as adults, while the six participants in the younger age group were born in the US as children of immigrants. All twelve participants are bilingual in Mandarin and English, and they are all familiar with the practice of codeswitching. Most participants did not know the name "codeswitching" for this process; rather, they called it "Chinglish." All participants were part of the Chinese Association of West Michigan (CAWM), located in Grand Rapids, Michigan. CAWM is a cultural organization that provides a community to the Chinese population of Grand Rapids and West Michigan as a whole. CAWM provides many cultural services, including food festivals, holiday celebrations, and the Grand Rapids Chinese Language School (a weekend language school).

Each person participated in their own Zoom call interview, making a total of 12 recorded interviews of codeswitched speech. The total speech time transcribed was 4 hours, 39 minutes, and 12 seconds.

3.1. DATA PREPROCESSING AND FEATURE SELECTION. Because the raw data (i.e. the first stage of manual transcriptions where everything said was transcribed) is messy and hard to work with for analysis, all transcriptions went through data cleaning and preprocessing in Python before being analyzed with models. All hesitation words (e.g. "like"/"um" in English or "那个"/"什么的" in Mandarin) were removed. Additionally, all English text was converted to lowercase, and all punctuation was removed. In the case of an English contraction, the apostrophe was removed and the contraction was converted to its full form (i.e. "I'm" would be "i am" in the clean data, with

the contraction expanded and letters in lowercase). Finally, text was tokenized into sentences separated by line breaks.

In addition to the content and transcriptions from the Zoom calls, each participant filled out a copy of the Language History Questionnaire 3.0 (LHQ 3.0) from Li et al. (2020). The LHQ is a tool used by linguists in generating a self-reported record of language proficiencies and habits of participants. This survey is built based on the most common questions historically asked by researchers to participants in studies that have to do with language and linguistics, and the quantified scores from LHQ survey responses are described below.

3.2. PROFICIENCY. The LHQ prompts participants to rate their proficiency levels for reading, listening, writing, and speaking on a scale from 1-7. For language *i*, Proficiency is calculated with (1) (Li et al. 2020):

(1) $$\textbf{\textit{Proficiency}}_{\,i} = \frac{1}{7}\sum_{j=\{R,L,W,S\}} \omega_j P_{i,j}$$

{R, L, W, S} are the scores for the components of reading, listening, writing, and speaking, respectively. P_{ij} represents the self-rated proficiency for language *i* for the j^{th} component, and w_j are the weights given to each component in the calculation. For my data, I set all components of the weights to be equal. LHQ uses the scaling factor of 1/7 to normalize the sum to be a value between 0 and 1 (since the scores are rated on a 7-point Likert scale). 0 represents the lowest level of proficiency, and 1 represents native level proficiency. Proficiency, then, is the sum of the weights multiplied by the proficiency ratings for each component, scaled with a factor of 1/7. While proficiency level could be argued to reflect a cognitive influence rather than a sociolinguistic one, I included self-rated proficiency as a feature that could potentially influence the presence or absence of codeswitching in my model because one's self perception of their fluency can reflect the comfort level they have with this language, as well as a speaker's attitudes towards their own language habits. Redinger (2010) found differences in the language and codeswitching choices of students who were trilingual in French, German, and Luxembourgish based on their self-perceived proficiencies as well as their schools' testing analyses of their proficiencies. Students who had higher self-rated proficiencies or school-tested proficiencies in a language tended to feel more comfortable codeswitching using that language (usually with another language they had high self-rated proficiencies in) (Redinger 2010). Additionally, self-rated proficiency is a reflection of one's own linguistic identity (i.e. how fluent they think they are in a language can affect how they construct their identity through code choices (Dweik & Qawar 2015)). With my data, I expect a higher proficiency to correlate with a higher level of usage in the respective language. Thus, if both languages are rated at similar high proficiencies, I would expect a higher likelihood of codeswitching.

3.3. IMMERSION. Immersion plays a large role in the language use and habits of speakers, so it is a natural sociolinguistic factor to include. Immersion can lead to higher self-rated proficiency in a language (Li et al. 2020), and thus more potential codeswitching between two languages that they are fluent in. LHQ uses the survey answers of age, age of acquisition (AoA), and years of use (YoU) of language *i* to calculate the Immersion score with (2) (Li et al. 2020):

(2) $$\textbf{\textit{Immersion}}_{\,i} = \frac{1}{2}\left[\sum_{j=\{R,L,W,S\}} \omega_j\left(\frac{\textbf{\textit{Age}} - \textbf{\textit{AOA}}_{i,j}}{\textbf{\textit{Age}}}\right) + \left(\frac{\textbf{\textit{YoU}}_i}{\textbf{\textit{Age}}}\right)\right]$$

Age is the speaker's current age in years, age of acquisition is the speaker's age when they started using language *i* in component *j* (i.e. speaking, reading, writing, or listening), and years of use represents the total number of years using language *i* in any component. Again, w_j is the weight of the given component, and a scaling factor of 1/2 is used to normalize the value to be between 0 and 1, as well as to give AoA and YoU equal weight. As with proficiency, I would expect that languages rated at similar immersion scores would correspond to an increased likelihood of codeswitching in my data.

3.4. DOMINANCE. Li et al. (2020) show that language dominance is related to a speaker's proficiency and daily use of a language, making it a reasonable sociolinguistic factor to include as a potential predictor in a speaker's language patterns. LHQ asks speakers for the number of hours per day they spend on different components of each language (e.g. how many hours one uses Mandarin for reading the news). Dominance for language *i* is calculated using (3) (Li et al. 2020):

(3)
$$Dominance_i = \sum_{j=\{R,L,W,S\}} \omega_j \left[\frac{1}{2}\left(\frac{P_{i,j}}{7}\right) + \frac{1}{2}\left(\frac{H_{i,j}}{K}\right)\right]$$

H represents the hours per day a speaker spends using language *i* on component *j*, and K is a constant scaling factor that LHQ writers have set to 16. The other variables shown have been used in the previous equations, and they represent the same scores. The additional scaling factor of 1/2 is also included here to normalize the dominance value to be between 0 and 1, as well as give proficiency and usage hours equal weight. Because each individual speaker can assess their own hours of usage differently from other speakers, LHQ uses another score, the L2 to L1 ratio, to give each speaker an individualized and contextualized dominance score against their other language, rather than comparing across speakers. I will give my predictions of the effect of dominance scores on codeswitching below, after introducing the L2 to L1 ratio.

3.5. L2 TO L1 DOMINANCE RATIO. LHQ provides a simple dominance ratio of L2 to L1 (or, in cases where more than two languages are measured by LHQ, the ratio of language *i* to L1). Li et al. (2020) compare this ratio to z-scores used in statistics in that it provides a more standardized comparison of language dominance across speakers. That is, by contextualizing the dominance of each language by individual speakers within their own set of languages, it is easier and more reliable to compare across multiple speakers.

This ratio is also used to determine if a speaker possesses "balanced" bilingualism (Treffers-Daller 2017), or if one language is clearly dominant over the other. The ratio is calculated using the straightforward (4) (Li et al. 2020), where L1 and L2 dominance scores are shown by $Dominance_{L1}$ and $Dominance_{L2}$, respectively:

(4)
$$Ratio_{Dominance\,L2} = \frac{Dominance_{L2}}{Dominance_{L1}} = \frac{0.4609}{0.671875} = 0.686$$

Because a ratio of 1.0 would indicate similar dominance levels between L1 and L2, I would expect that speakers whose L2 to L1 dominance ratio is closer to 1 would codeswitch more frequently.

3.6. MULTILINGUAL DIVERSITY SCORE. The latest version of LHQ also included the calculation of a Multilingual Diversity Score (MLD), which takes into account all the languages that are represented in each speaker's survey results (i.e. not just the answers for a speaker's L1 and L2). The MLD is the name that LHQ uses for a measure called the Shannon Entropy (here marked by

the variable *H*, distinct from $H_{i,j}$ used in the Dominance Score calculation), as proposed by Gullifer and Titone (2019). The Shannon Entropy (*H*) measures the social diversity of language use—that is, whether a speaker's language habits (i.e. hours of daily usage, domains of usage, etc.) reflect a Compartmentalized Fashion (CF) or Integrated Fashion (IF) of bilingualism. Gullifer and Titione (2019) define CF as bilingualism in which speakers codeswitch at the most minimal amount possible because they view their languages as distinctly separate and reserved for different domains of use. IF is bilingualism that manifests in an abundance of codeswitching due to the speaker viewing the languages as integrated within the same domains of use—or viewing both languages (often showing up in codeswitches) as acceptable for use with certain audiences. A higher Shannon Entropy *H* or MLD (as LHQ calls it) shows that a speaker's language habits lean more towards IF than CF. *H* is calculated using (5) and (6) (Li et al. 2020), with (5) being the calculation of a temporary variable that is used in the final calculation of *H* (or MLD):

(5)
$$PD_i = \frac{Dominance_i}{\sum_{i=1}^{n} Dominance_i}$$

PD_i represents the Proportion of Dominance of language *i*, where *n* is the total number of languages a participant has learned or uses. PD_i is used in (6):

(6)
$$H = -\sum_{i=1}^{n} PD_i \log_2(PD_i)$$

H will result in a value between 0 and 2, where a score of 1 shows maximum balanced bilingualism as defined by Gullifer and Titone (2019). A score closer to 0 or 2 will represent a speaker's habits leaning towards one language or the other. For example, for *H* to be a score of exactly 1 (a "balanced bilingual"), the equation would have to look like (7) (Gullifer & Titone 2019) as follows, where PD_i is 0.5:

(7)
$$\begin{aligned} H &= -(0.5 * \log_2 0.5 + 0.5 * \log_2 0.5) \\ &= -[0.5 * (-1) + 0.5 * (-1)] \\ &= 1 \end{aligned}$$

I included the MLD in my analysis because it quantifies what integrated (i.e. balanced) bilingualism compared to compartmentalized bilingualism looks like based on self reported scores of proficiency, immersion, and dominance. It takes into account many sociolinguistic influences like domain and method of language use (in components *j* of a given language), frequency of use in hours and dominance ratios across languages, and self-rated scores reflecting one's linguistic identity and language attitudes. By tracing each calculation in the MLD equation back to the equation for Proficiency (1), MLD reflects the most aggregate measure of a speaker's language use. Because of this, this score is an exceptionally informative quantification of a speaker's tendency to integrate their languages by using codeswitching. Therefore, I expect MLD scores closer to 1 to correlate with a higher likelihood of codeswitching, while MLD scores close to 0 or 2 to correlate with the opposite.

3.7. AGE AND EDUCATION LEVEL. Age and education level were also collected with the LHQ survey. Age is given straightforwardly as a number, while education level is given with one of the three labels: College (Bachelor's Degree), Graduate School (Master's Degree), or Graduate School (Doctoral Degree). I encoded the three levels of education as a one-hot vector (with

dimensions 1 x 3) as follows: [Bachelor's Degree, Master's Degree, Doctoral Degree]. Someone with just an undergraduate college education would be encoded as [1, 0, 0], whereas someone with a master's degree would be [0, 1, 0] and someone with a doctorate would be [0, 0, 1].

In the portion of the Zoom calls where participants were asked metalinguistic questions, many expressed that codeswitching, to them, is unacademic and unprofessional. This could be because of the misconception discussed in Poplack (1980) that codeswitching is a sign that someone is not proficient or fully bilingual in one or more of their languages (i.e. CS is used to fill gaps rather than be a language habit of someone who has strong cognitive access to both languages). Thus, I include the factor of education level to see if this affects codeswitching. I hypothesize participants who hold a Bachelor's Degree to codeswitch more frequently than participants who hold a Master's Degree or Doctoral Degree. Everyone in the older age group holds some Graduate School degree, whereas only one participant in the younger age group holds a Graduate School degree (Master's). Because participant age and education level is split almost on the same line, there could potentially be confounding between the two variables.

3.8. SENTENCE LENGTH. While sentence length is not a sociolinguistic variable, it can influence the presence or absence of a codeswitch in a sentence simply by the virtue of longer sentences having more possible instances of a CS occurrence. For the task of measuring sentence length, I used Stanza (Qi et al. 2020), a collection of natural language processing (NLP) tools created and made publicly available by Stanford University's NLP researchers. Stanza contains Python tools for Mandarin tokenization of words and English tokenization of words. Using a series of small preprocessing tasks that allowed me to count the number of Mandarin words, count the number of English words, and then add them together, I found the sentence length for each sentence in my data. I predict that a longer sentence will be correlated with a higher frequency of codeswitches.

3.9. FINDING THE PRESENCE OF A CODESWITCH IN A SENTENCE-LEVEL TOKEN. Finally, the variable being predicted is the presence or absence of a codeswitch in a sentence. For this, I made a simple tool (described below) in Python to go through each sentence and see if the sentence contained both an instance of Chinese characters and an instance of English letters. If at least one instance of each was found, that sentence would be assigned the binary value of '1.' If a codeswitch was not found, the sentence would be assigned a value of '0.'

The predictors I included in the models in this paper were the following: sentence length, age, education level, L1 proficiency score, L2 proficiency score, L1 immersion score, L2 immersion score, L1 dominance score, L2 dominance score, L2 to L1 dominance ratio, and multilingual diversity score. These predictors were all used in my final model as features to predict the presence or absence of a codeswitch in a sentence. Education level was the only categorical variable, and I used one-hot encoding to capture education level in a vector of three indices ([Bachelor's Degree, Master's Degree, Doctoral Degree]).

4. Results. A total of 12 participants' Zoom interviews were transcribed, which amounted to slightly more than four hours of speech. There were 1340 sentences transcribed in the final (cleaned) dataset. Out of the 1340 sentences, 309 contained at least one instance of CS. 640 sentences were spoken by the older age group and 700 were from the younger age group. There were a total of 16,285 words, 7064 of which were English words and 9221 of which were in Chinese. Of the 9221 Chinese words, 7082 were spoken by the older age group and 2139 were spoken by the younger generation. Of the 7064 English words, 1045 were uttered by the older generation and 6019 came from the younger generation.

4.1. LANGUAGE IDENTIFICATION. I created my own simple deterministic LID tool using UTF-8 from Unicode and the Regular Expressions Python package to detect whether or not a string contains Chinese characters, English orthography, or both (i.e. a codeswitch). Because English and Mandarin use different orthography, this tool performed with 100% accuracy on my data (evaluated by manually checking the tool's LID on my text). While we should, in theory, expect that there is 100% accuracy all the time, potential roadblocks include Chinese characters that are not yet in UTF-8. In my data, punctuation was removed during data preprocessing, but if this tool is run on data that does not have punctuation removed with languages that use different punctuation symbols (as is the case with Mandarin using different commas and a different period from English), the accuracy could decrease as well.

LID is an important task needed for other tasks such as identifying a sentence that contains a codeswitch or counting the number of codeswitches in a sentence, and thus was included in analyzing the data that was inputted into the final model.

5. Model. CA CART models predict a dependent variable based on a handful of input variables. In my case, these input variables are the 11 predictors I outlined in my methodology: age, education level, L1 proficiency score, L2 proficiency score, L1 immersion score, L2 immersion score, L1 dominance score, L2 dominance score, L2 to L1 dominance ratio, and multilingual diversity score. Each token in this model is a sentence, so there are a total of 1340 tokens. I performed this analysis using R, drawing inspiration from the models and theory in Fedorova (2021). The algorithm behind a CART model is a series of questions, where the answers of one sequence of questions will determine what, if any, should be the next question(s). After processing, a tree is generated where the nodes are the answers to the questions that were significant. The higher up an answer or question shows up on the tree, the more important the question or answer was.

The evaluation metrics used in this section are accuracy and the receiver operating characteristic (ROC) curve. Because 309 of the 1340 sentences in the dataset contain at least one instance of a codeswitch, there is an inherent class imbalance in the data. Specifically, the smaller class (sentences containing CS) makes up .23 of the data, while the larger, non-CS class makes up .77 of the data. Because of this, the evaluation metric of accuracy can be misleading as it can be a mere reflection of how well the model learns class imbalances. If a model predicted that a sentence did not contain CS 100% of the time, it would still result in an accuracy of .77. Thus, when looking at the accuracy score of the models described in this section, the value of the accuracy is compared against the value of the larger class, which is .77. This means that many seemingly high accuracy scores are actually not reflective of the class imbalances of the data, and therefore not the most ideal representation of a model's predictive power.

Fortunately, the ROC curve and its corresponding area under the curve (AUC) is a much better evaluation metric for datasets with class imbalances. The AUC of the ROC measures how well a model can distinguish true positives as true positives and true negatives as true negatives, thus helping test against false positive values, even despite inherent class imbalances. When looking at this evaluation metric, the value of the ROC curve (i.e. the AUC of the ROC) is compared against the value of random chance, or .50. A high ROC curve value, then, is actually representative of a model's high predictive power. Because of this, the ROC curve is a much better evaluation metric for my models and dataset. In the models discussed in this section, both accuracy and the ROC curve will be mentioned.

5.1. LOGISTIC REGRESSION. I first performed a logistic regression on my data. Each token was a sentence, and I partitioned 1070 of 1340 (about 80%) total sentences to be in the training set. The

rest of the data was my test set. The accuracy of the simple logistic regression model was .652, and the AUC of the ROC curve was .689, which shows that my regression model performed better than if the model had learned only the class imbalances and predicted based on those (which meant the model performed better than the equivalent of random chance based on false positives). It is important to note that the accuracy of this model is lower than even the class imbalance value of .77, implying that the patterns in my data may not be linear. Since a simple logistic regression model is linear, it did not capture the trends in the dataset as well as the other models described below.

5.2. CART MODEL. Because of the large number of sociolinguistic predictors in my data, the algorithm of a CART model (explained above) provides a clear visual representation of which predictors are the most important. In the process of determining which predictors were most important to include in the CART model, I tested the inclusion and exclusion of predictors until the final model was developed and run. Of the 11 predictors, sentence length is the only non-sociolinguistic predictor (i.e. it is more of a syntactic predictor), so each of the other individual (sociolinguistic) predictors were isolated from each other to be tested alongside the predictor of sentence length. Sentence length was the constant predictor throughout all of these test runs. The results (accuracy and ROC) are shown in Table 1 below. As mentioned above, the ROC is more representative of a model's true predictive power than the accuracy. The seemingly high accuracy scores of some predictors is misleading for this reason. In other words, the values in the column "AUC of the ROC" were helpful in finding the predictors that ended up appearing as important nodes (with predictive power) in the final CART model.

Predictor	Accuracy	AUC of the ROC
Age	**0.841**	**0.732**
Education (encoded as one-hot vector)	0.792	0.574
L1 Proficiency	0.833	0.706
L2 Proficiency	0.852	0.755
L1 Immersion	**0.830**	**0.746**
L2 Immersion	0.829	0.704
L1 Dominance	0.826	0.691
L2 Dominance	**0.844**	**0.756**
L2 to L1 Dominance Ratio	0.796	0.608
Multilingual Diversity Score	0.837	0.693

Table 1. Accuracy and ROC of individual predictors

The predictors that ultimately showed up as visible nodes in the CART model are bolded in Table 1, though it is worth noting that L2 Proficiency also showed a high AUC of the ROC when tested individually as a predictor. The reason L2 Proficiency did not show up in the final visualization of the CART model (discussed below) may be due to predictor interaction when all the predictors were included together in the final model. However, L2 Proficiency being a strong predictor in CS habits is in line with the findings of Redinger (2010) (as mentioned in an earlier section).

When performing the final CART analysis on the data, the model used the same 11 predictors as in the logistic regression. The same partitioning of training and testing data was also used. The final CART model performed with an accuracy of 0.804 and an AUC for the ROC of .692. Both of these measures were improvements from the logistic regression model, indicating that the nonlinear modeling captured the data better (as seen with the accuracy score being higher than the class imbalance value of .77), and that the CART model had a higher true predictive power with a greater area under the ROC curve.

The four features that the CART model highlights as important nodes on the tree were age, sentence length, L1 immersion, and L2 dominance. L2 dominance shows up at the very top of the tree, followed by sentence length, L1 immersion, age, and then sentence length again. The tree generated by my model is show in Figure 1 below:

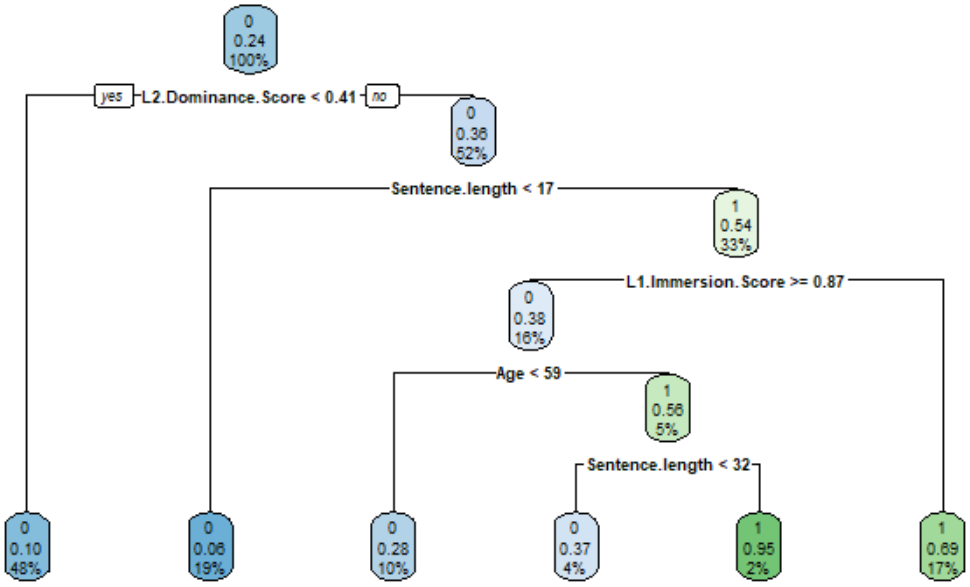

Figure 1. CART tree

At each node, the model shows what the values of each predictor were when analyzing which direction the tree would fall after each step. While the 4 predictors shown in Figure 1 were the most important predictors, this final model still included all 11 predictors; the rest merely did not show up on the tree visualization.

To check for potential noise with interaction of the remaining 7 predictors, the model was tested using only the 4 predictors (1 syntactic one, 3 sociolinguistic ones) shown in the visualization of Figure 1. However, when only the 4 predictors were used, the AUC of the ROC decreased from .692 to .655. The accuracy increased from .804 to .807, but as mentioned above, accuracy is not as representative of an evaluation metric for predictive power with these models and data. These results imply that while the visualization only shows the 4 most important predictors, the remaining predictors are also critical to the predictive power of the model and cannot be removed in a truly holistic view of sociolinguistic factors that affect CS.

To further confirm if the predictors my CART model generated as important were actually significant, I used the sjPlot package downloaded into RStudio to look at the effects of

each individual predictor shown on this tree on CS predictions. The four graphs are shown in Figure 2 below:

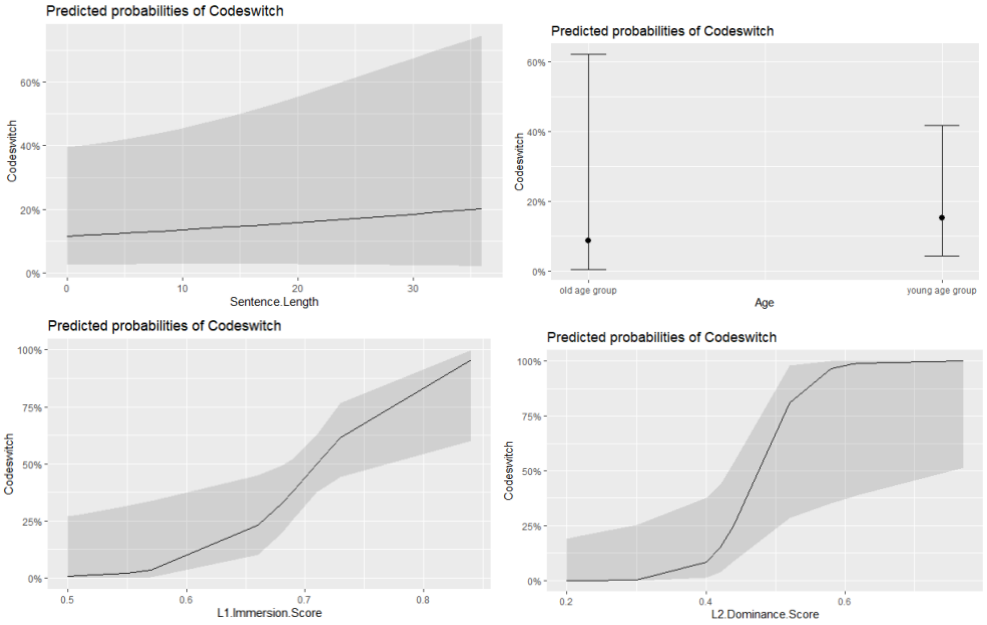

Figure 2. sjPlot generations of predictors

As the graphs show, an increase in sentence length leads to a higher percentage of codeswitching. The trend line in the sentence length graph is gradual. The line in the graph modeling age shows a decrease in codeswitching frequencies with an increase in age. This is in line with the chi-square test performed earlier. The L1 immersion and L2 dominance graphs show an increase in codeswitching with an increase in score, though the L2 dominance score plateaus at around .6 (in L2 dominance score value). These graphs show the logic behind why these predictors are strong predictors of codeswitching. Sentence length gives more space for potential CS instances to occur (in a longer sentence), correlating with higher rates of CS. For age, the previously performed chi-square test showed that younger generation speakers codeswitch at a statistically significantly higher frequency than older generation speakers. The plot for age generated by sjPlot treats age as a binary factor. L1 immersion and L2 dominance both show the high presence of one language or the other in a speaker's day to day life and language use, and it makes sense that they would co-occur as both strong predictors of CS.

5.3. RANDOM FOREST AND UPSAMPLING. While CART's visualization of individual predictors is helpful, I wanted to confirm these trends in as many ways as possible, so I performed an analysis using a Random Forest model as well. Random Forest is a machine learning algorithm that is composed of many smaller decision trees. Each small decision tree, or estimator, makes its own predictions. While CART is only one tree, Random Forest outputs the mode of all the decision trees that make predictions, so the analysis and accuracy is more robust and accounts for some of the variance of a single CART model. The Random Forest model used the same 11 predictors as the CART model and logistic regression model, as the CART model showed that all 11 predictors contributed to predictive power. The Random Forest model performed with an accuracy of .811, with the AUC for the corresponding ROC curve being .742. However, the downfalls of my data include a class imbalance (i.e. .77 of sentences that are non-CS and only .23 of sentences that are CS) and a small size for a machine learning algorithm or model.

Because of this, many researchers use a process called upsampling, which essentially expands the dataset by randomly sampling (with replacement) from the smaller class until both classes are equal, to counter class imbalances and have more data to draw from. The original data is untouched; additional samples are merely added. This helped the class imbalance issue in my data because, as seen above, my data inherently has many more instances of sentences with no CS at all than sentences that have the presence of CS. After upsampling, the Random Forest model performed with .807 accuracy, with the AUC of the upsampled Random Forest ROC curve being .774. Thus, the class imbalance problem was improved in the upsampled Random Forest model, as can be seen by the improvement in the area under the ROC curve.

A summary of the models described in this section is shown in Table 2 below, where the upsampled Random Forest model ultimately performed with the highest AUC of the ROC (.774). Thus, the upsampled Random Forest had the largest predictive power. This makes sense, as Random Forest itself helps to account for variance in CART, and upsampling combats the class imbalance problem even further. However, the CART model provided the best visualization and dissection of each predictor in the model.

Model	Accuracy	AUC of the ROC
Logistic Regression	0.652	0.689
Final CART model with 11 predictors	0.804	0.692
Random Forest	0.811	0.742
Random Forest with Upsampling	0.807	0.774
CART with only 4 predictors	0.807	0.655

Table 2. Summary of models

6. Conclusion and discussion. In exploring language technologies, I ran into issues surrounding the processing of multilingual speech; existing language technologies do not take many sociolinguistic factors into account, as much of the current research focuses on syntactic predictions. However, this allowed for my CART model to fill a gap where sociolinguistics was missing in computational literature. My model looked at 11 predictors, 10 of which were related to sociolinguistic factors. The four predictors that CART modeled as the most important were age, sentence length, L1 immersion, and L2 dominance, where codeswitching frequencies increased with sentence length, L1 immersion scores, and L2 dominance scores. Codeswitching frequencies decreased with increasing age. 3 out of these 4 CART-generated predictors are sociolinguistically-related, which shows that this should be an area of greater future research. The results of my models using sociolinguistic predictors are comparable to the results of previous work using only syntactic constraints (e.g. Li and Fung 2014) as model predictors. In the future, sociolinguistic analyses and predictors should be researched to a much greater extent, especially in conjunction with the already existing syntactic and semantic ways to model language in computational research. Further, by virtue of developing multilingual models, one will have to take sociolinguistic and identity factors into account, as multilingualism is inextricably tied to sociolinguistics and linguistic identity.

References

Belazi, Heidi, Edward J. Rubin & Almeida Jacqueline Toribio. 1994. Code switching and X-Bar Theory: The Functional Head Constraint. *Linguistic Inquiry*, 25(2). 221–237.

Bullock, Barbara E., Wally Guzman, Jacqueline Serigos, Vivek Sharath & Almeida Jacqueline Toribio. 2018. Predicting the presence of a Matrix Language in codeswitching. *3rd Workshop of Computational Approaches to Linguistic Code-switching, ACL*. https://doi.org/10.18653/v1/W18-3208.

Dweik, Bader Sa'id & Hanadi A. Qawar. 2015. Language choice and language attitudes in a multilingual Arab Canadian community: Quebec-Canada: A sociolinguistic study. *British Journal of English Linguistics* 3. 1–12.

Fedorova, Ekaterina. 2021. #ForYouPage: Exploring differences in users' interaction with the TikTok recommendation algorithm and hashtag system using machine learning techniques. Berkeley, CA: University of California Berkeley BA thesis.

Gullifer, Jason & Debra Titone. 2019. Characterizing the social diversity of bilingualism using language entropy. *Bilingualism: Language and Cognition* 23. 283–294.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, & Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 58. 3575–3585. https://doi.org/10.18653/v1/2020.acl-main.329.

Li, Chia-Yu & Ngoc Thang Vu. 2019. Integrating knowledge in end-to-end automatic speech recognition for Mandarin-English code-switching. *Proceedings of the 2019 International Conference on Asian Language Processing (IALP)*. 160–165. https://doi.org/10.48550/arXiv.2112.10202.

Li, Ping, Fan Zhang, Anya Yu & Xiaowei Zhao. 2020. Language History Questionnaire (LHQ3): An enhanced tool for assessing multilingual experience. *Bilingualism: Language and Cognition* 23(5). 938–944.

Li, Ying & Pascale Fung. 2014. Language modeling with Functional Head Constraint for code switching speech recognition. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 907–916. https://doi.org/10.3115/v1/D14-1098.

Myers-Scotton, Carol. 1993. *Duelling languages: Grammatical structure in codeswitching.* Oxford: Clarendon Press.

Poplack, Shana. 1980. Sometimes I'll start a sentence in spanish y termino en espanol: Toward a typology of code-switching. *Linguistics* 18(7–8). 581–618. https://doi.org/10.1515/ling.1980.18.7-8.581

Poplack, Shana. 1978. Syntactic structure and social function of code-switching. In Richard Duran (ed.), *Latino Language and Communicative Behaviour* 6.

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton & Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. *Association for Computational Linguistics (ACL) System Demonstrations.* https://doi.org/10.18653/v1%2F2020.acl-demos.14

Redinger, Daniel. 2010. Language attitudes and code-switching behavior in a multilingual educational context: The case of Luxembourg. York, UK: University of York dissertation.

Treffers-Daller, Jeanine. 2017. Do balanced bilinguals exist? Lecture presented at the University of Hildesheim Multilingualism and Diversity Lectures.

Yi, Irene. n.d. Sociolinguistic factors of Mandarin-English codeswitching: Language attitudes, age, and other factors used for computational modeling. Manuscript. Unpublished.