Using Natural Language Processing to Quantify the Efficacy of Language Simplification as a Communication Strategy

Brian Nalley
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

Part of the Statistics and Probability Commons

Recommended Citation
Nalley, Brian, "Using Natural Language Processing to Quantify the Efficacy of Language Simplification as a Communication Strategy" (2023). All Graduate Theses and Dissertations. 8824.
https://digitalcommons.usu.edu/etd/8824

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
USING NATURAL LANGUAGE PROCESSING TO QUANTIFY THE EFFICACY OF LANGUAGE SIMPLIFICATION AS A COMMUNICATION STRATEGY

by

Brian Nalley

A thesis submitted in partial fulfillment of the requirements for the degree

of

MASTERS OF SCIENCE

in

Statistics

Approved:

Alan Wisler
Major Professor

Brennan Bean
Committee Member

Stephen Walsh
Committee Member

D. Richard Cutler, PhD
Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023
ABSTRACT

Using Natural Language Processing to Quantify the Efficacy of Language Simplification as a Communication Strategy

by

Brian Nalley, Master of Science
Utah State University, 2023

People with communication disorders often experience difficulties being understood by unfamiliar listeners or in noisy environments. A common strategy for effectively communicating in these scenarios is to use simpler and more predictable language. Despite the prevalence of this strategy there has been little to no research to date focused on the effectiveness of language simplification as a communication strategy. This study seeks to begin filling that gap by using natural language processing to determine whether speakers with early-stage Parkinson’s disease and age-matched neurotypical speakers are able to successfully simplify their language while still maintaining the original message.

Simplification was measured by several lexical diversity (MSTTR, HDD, MTLD) and lexical sophistication (Advanced Guiraud) measures. Natural language processing methods were deployed to automatically compute the above metrics for text transcriptions of a story simplification task by each participant. Word2Vec word embeddings were combined with soft cosine similarity to assess the similarity of the original and simplified texts. Preliminary box plots for each measure indicated potentially non-normal distributions so Shapiro-Wilk and Brown-Forsythe tests were utilized to
assess the normality of each distribution and whether the two groups had homogenous variances. The three lexical diversity measure control groups failed the Shapiro-Wilk test and passed the Brown-Forsythe test while the lexical sophistication measure passed both tests for both groups. Because several of the measures do not display normal distributions, 1-sample Wilcoxon rank sum and 2-sample Mann-Whitney $U$ tests were conducted in order to determine the success of the simplification task and whether either group was more successful than the other, respectively.

Each measure indicated that both groups showed statistically significant reductions in the complexity of their language in the rephrased passage relative to the original text. Additionally, the soft cosine similarity scores indicated that, on average, participants were able to maintain the original story’s message. These results provide strong preliminary evidence for the efficacy of language simplification as a communication strategy.
PUBLIC ABSTRACT

Using Natural Language Processing to Quantify the Efficacy of Language Simplification as a Communication Strategy

Brian Nalley

People with communication disorders often experience difficulties being understood by unfamiliar listeners or in noisy environments. A common strategy for effectively communicating in these scenarios is to use simpler and more predictable language. Despite the prevalence of this strategy, there has been little to no research to date focused on the effectiveness of language simplification as a communication strategy. This study seeks to begin filling that gap by using natural language processing to determine whether speakers with early-stage Parkinson’s disease and age-matched neurotypical speakers are able to successfully simplify their language while still maintaining the original message.

Simplification was measured by several lexical diversity and lexical sophistication measures. Natural language processing methods were deployed to automatically compute the above metrics for text transcriptions of a story simplification task by each participant. A similarity score was also calculated to measure how closely each retelling mapped to the original story.

Each measure indicated that both groups showed statistically significant reductions in the complexity of their language in the rephrased passage relative to the original text and were overall able to maintain the original story’s message. These results provide strong preliminary evidence for the efficacy of language simplification as a communication strategy.
ACKNOWLEDGMENTS

This research was funded by a grant from the ASH Foundation. Thank you to Dr. Annalise Fletcher for trusting me with this portion of her project as well as to her language lab and students for transcribing the participant interviews. Last but far from least, Nicole, without whose support and patience none of this would have been possible.

Brian Nalley
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>PUBLIC ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
</tbody>
</table>

## CHAPTER

1. Introduction ................................................. 1
   - Lexical Richness ........................................ 2
   - Lexical Diversity ....................................... 3
     - TTR & Transformations .................................. 3
     - $D$ & Hypergeometric $D$ ................................ 5
     - MTLD .................................................. 9
   - Lexical Sophistication .................................. 11
     - Advanced Guiraud ...................................... 11
     - LFP .................................................. 12
     - $P_{Lex}$ ............................................. 15
   - Textual Similarity ...................................... 15
     - Soft Cosine Similarity ................................ 18
   - Review .................................................. 19

2. METHODS ...................................................... 21
   - Data Collection ......................................... 21
   - Natural Language Processing Pipeline ................. 21
     - Pre-Processing ........................................ 21
     - Lexical Measures ...................................... 22
     - Post-Processing ....................................... 23
   - Statistical Testing ..................................... 24
   - Textual Similarity ...................................... 25

3. RESULTS ....................................................... 29
   - Exploratory Plots ........................................ 29
   - Preliminary Analysis .................................... 29
   - Assessing Simplification ................................ 32
   - Textual Similarity ...................................... 32

4. DISCUSSION .................................................... 36
   - Limitations & Future Work ............................... 38
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Means and (Standard Deviations) for each lexical richness measure</td>
<td>31</td>
</tr>
<tr>
<td>3.2 Normality and homogeneity of variance test results for each lexical richness measure. Each column includes the test statistic and (p-value).</td>
<td>31</td>
</tr>
<tr>
<td>3.3 Statistical test results for each lexical richness measure. The test statistic is listed first followed by the (p-value.)</td>
<td>32</td>
</tr>
<tr>
<td>3.4 Ordinary Least Squares Regression results using group and soft cosine similarity score to predict lexical richness. P-values which are bold are significant following the Benjamini-Hochberg procedure.</td>
<td>34</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.1</td>
<td>Alice in Wonderland Type-Token Ratio (TTR) Curve (Tweedie &amp; Baayen, 1998)</td>
</tr>
<tr>
<td>1.2</td>
<td>Empirical Type-Token Ratio (TTR) curve (solid) and Best Fitting Theoretical TTR Curve (dashed) (McCarthy &amp; Jarvis, 2007).</td>
</tr>
<tr>
<td>1.3</td>
<td>Empirical Type-Token Ratio (TTR) curve (solid) and Best Fitting Theoretical TTR Curve (dashed) combined with Hypergeometric $D$ (HDD) curve (dot-dash line) (McCarthy &amp; Jarvis, 2007).</td>
</tr>
<tr>
<td>1.4</td>
<td>Word2Vec architectures. CBOW predicts a given word based on its context while Skip-gram predicts a given word’s context (Mikolov, Chen, et al., 2013).</td>
</tr>
<tr>
<td>2.1</td>
<td>Text processing pipeline flow chart. The terms in parentheses represent the Python packages and custom functions needed for each step. For brevity, Advanced Guiraud has been abbreviated as AG.</td>
</tr>
<tr>
<td>3.1</td>
<td>Box plots for each lexical richness measure with observations overlaid. Proceeding clockwise beginning in the upper left-hand corner these are: Mean Segmental Type-Token Ratio (MSTTR), Hypergeometric $D$ (HDD), Advanced Guiraud and Measure of Textual Lexical Diversity (MTLD).</td>
</tr>
<tr>
<td>3.2</td>
<td>Box plots of Soft Cosine Similarity scores for Parkinson’s and control groups with comparison lines for a synonym-only retelling of the original story and “The Princess and the Pea”</td>
</tr>
<tr>
<td>3.3</td>
<td>Scatter plots comparing lexical richness and textual similarity scores for each participant with linear regression lines overlaid. The red line represents the regression equation for the Parkinson’s group, while the blue line is that for the control group, while the observations for each group are in the same colors. Proceeding clockwise beginning in the upper left-hand corner these are: Mean Segmental Type-Token Ratio (MSTTR), Hypergeometric $D$ (HDD), Advanced Guiraud and Measure of Textual Lexical Diversity (MTLD).</td>
</tr>
<tr>
<td>4.1</td>
<td>Mean Segmental Type-Token Ratio (MSTTR) Linear Regression Diagnostic Plots</td>
</tr>
<tr>
<td>4.2</td>
<td>Hypergeometric $D$ (HDD) Linear Regression Diagnostic Plots</td>
</tr>
<tr>
<td>4.3</td>
<td>Measure of Textual Lexical Diversity (MTLD) Linear Regression Diagnostic Plots</td>
</tr>
<tr>
<td>4.4</td>
<td>Advanced Guiraud Linear Regression Diagnostic Plots</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

Parkinson’s disease primarily manifests itself via impaired motor control for its sufferers. While the stereotypic perception among the laity is of difficulty walking or of a particular gait, Parkinson’s affects all parts of one’s life that require fine motor control, including the ability to speak. As such, many people with Parkinson’s encounter difficulties being understood by others, particularly by new people or in loud environments. To address this, early speech therapy interventions often include advising patients to speak louder, more clearly, and to use simpler, more predictable language. There have been many studies focused on acoustic interventions, for example speaking more loudly or more slowly (Fletcher et al., 2017). Two highly researched acoustically-based examples are Lee Silverman Voice Treatment (LSVT) (LSVT Global, n.d.) and Speak Out! (Parkinson Voice Project, n.d.). LSVT is quite intensive, requiring several speech therapy sessions per week with additional homework, and primarily focuses on helping people with Parkinson’s speak more loudly. Speak Out!, as its name suggests, also primarily focuses on speech volume with the added component of “speaking with intent” (Parkinson Voice Project, n.d., About Speak Out/Individual Speech Therapy) In spite of, or perhaps because of, this focus on helping people with Parkinson’s speak more loudly, we are not aware of any research to date that investigates whether people, including those with Parkinson’s, are actually able to intentionally simplify their language, despite its use as an early intervention.

The primary research question under consideration is whether participants are able to simplify their language when directed to do so. Secondly, if they are indeed able to simplify their speech, was either of the Parkinson’s or control groups better able to do so? From personal, informal experience, we expect those without any neurological disorder to be able to successfully complete the task; however, it is important both to formally establish the veracity of this assumption as well as to see if those with Parkinson’s are also able to do so. As all tasks which involve fine motor control require more cognitive effort to complete successfully for those with Parkinson’s, there is a possibility that adding the additional task of simplification on top of the cognitive demands required to speak loudly...
and intelligibly enough may make this task more difficult for them. A third research question is, if simplification was successful for either group, has the simplified version maintained the meaning of the story they are tasked with simplifying? Of these research questions, the first two stem directly from the research of which this project is a piece, while the third while likely remain exclusive to this thesis.

We will be combining traditional measures of complexity with natural language processing (NLP) methods in order to assess our participants. We have decided to incorporate NLP, first and foremost, for reasons of objectivity. By using less subjective metrics that can be automatically evaluated via computer, we do not have to be as concerned that an individual’s opinions or perceptions may cloud the analysis, either via inter-rater reliability or a host of other potential complications. Secondly, once properly initialized, the computer can score multiple measures for each participant in a matter of minutes, significantly reducing the amount of time needed to complete a study such as this. Lastly, once the initial code is written, it can be easily adapted to, for example, include additional participants or be used by other researchers to answer a similar question.

Lexical Richness

Lexical richness is the umbrella term which encompasses each of the measures of language simplification we will be using in this study. Generally, each measure was developed to study language acquisition or skill in either children learning their first language or second language learners (L2). Additionally, most were primarily intended to be used with written texts though we will see some examples of others doing as we will be and using them to measure oral texts.

Within the category of lexical richness there are three main branches: lexical diversity, lexical sophistication and lexical density (M. Daller & Xue, 2007; Lu, 2012). We will define lexical diversity as the number of different words used by a person, lexical sophistication as how often low frequency or “advanced” words are used and lexical density as the frequency of “lexical” words. Lexical words are defined by Lu (2012) as nouns, adjectives, verbs, excluding modal verbs, auxiliary verbs, “be” and “have,” and adverbs with an adjectival base, including those that can function as both an adjective and adverb and those formed by attaching an “-ly” suffix to an adjectival root. Of these three categories, we will primarily focus on lexical diversity and lexical sophistication as the majority of previous research has focused on these two forms of lexical richness, with fewer recent research examples utilizing lexical density.
Lexical Diversity

Three of our four metrics fall under the category of lexical diversity: Mean Segmental Type-Token Ratio (MSTTR), $D$, more specifically Hypergeometric $D$ (HDD), and Measure of Textual Lexical Diversity (MTLD). This is the oldest, and in many forms, simplest branch of the lexical richness tree beginning with the original lexical diversity measure, Type-Token Ratio (TTR) and its many mathematical transformations.

**TTR & Transformations**

As shown in Equation 1.1, TTR

$$\frac{T}{N}$$

(1.1)

is a simple formula where the number of types ($T$), or unique words, are divided by the number of tokens ($N$), or total words, in a text. As types and tokens feature in the majority of the following formulae, for simplification purposes I will define types to be $T$ and tokens as $N$ in the formulae as well.

TTR has been in use since at least the 1940’s and its limitations are well known, namely that longer texts tend to produce lower scores (Lu, 2012; Hout & Vermeer, 2007; Tweedie & Baayen, 1998; Johnson, 1944; M. Daller & Xue, 2007; Malvern & Richards, 2002; Jarvis, 2002; Yu, 2010; H. Daller et al., 2003; McCarthy & Jarvis, 2007). This makes sense, as the longer a text, spoken or written, continues, there will be fewer and fewer new words which can be introduced, especially if we are constructing a logical, coherent text. An example can be seen in Figure 1.1, taken from Tweedie & Baayen (1998), which displays how the TTR score changes over the course of *Alice in Wonderland*. On the Y-axis we have the text TTR value, labeled as $P(N)$ and the X-axis has the number of tokens in the text; we can clearly see how TTR decreases systematically as the text continues. While the length of this text is obviously much longer than any text we will be analyzing in this study, the overall pattern of lower scores for longer texts is one which is seen repeatedly. Despite this weakness, TTR is still relatively widely used, generally as a baseline comparison but also occasionally as a legitimate lexical diversity measure (M. Daller & Xue, 2007).

Owing to this text-length dependence problem, many researchers have attempted many mathematical transformations on the basic TTR formula, with varying degrees of success. As will become a running theme, each of these measures is also dependent on text length to various degrees, where instead of staying relatively constant for different text lengths, scores will decrease the longer a text
continues, even if at perhaps a slower rate than with TTR. Additionally, Tweedie & Baayen (1998) point out an issue that is often ignored, that each of the lexical diversity methods we will discuss, save MTLD, treat each text as a bag-of-words wherein each word is considered equally likely to appear in any location, which is clearly not the case for any text that requires logical ordering and syntax in order to make sense.

Two formerly commonly used transformations that we will only touch on are those of Herdan (1960) (Log TTR) and Maas (1972). As we see in Equation 1.2,

\[
\frac{\log_{10} T}{\log_{10} N}
\]  

(1.2)

Herdan proposed taking the logarithm of both the types and tokens, while Maas’s formula (Equation 1.3)

\[
a^2 = \frac{\log_{10} T - \log_{10} N}{(\log_{10} T)^2}
\]  

(1.3)

is quite a bit more complicated than any we have discussed thus far. Tweedie & Baayen (1998) conducted Monte Carlo simulations with both Herdan and Maas and found that, just as with TTR, as text length increased, their scores decreased monotonically.

A basic but relatively successful and widely used transformation is Guiraud or Root TTR (Equation 1.4),

\[
\frac{T}{\sqrt{N}}
\]  

(1.4)

here the number of types is now divided by the square root of the number of total tokens. The
addition of the square root allows Guiraud to better maintain the TTR curve over the entirety of a document (H. Daller et al., 2003; M. Daller & Xue, 2007).

Hout & Vermeer (2007) preferred Guiraud among the traditional transformations though they did express concern at its performance with very short (under 100-200 words) or very long (∼1,000+) texts. While the reason for their concern is warranted, all of our texts will be several hundred words long, so this particular behavior does not strike us as problematic with regards to this project. In Tweedie & Baayen’s (1998) simulations, counterintuitively, Guiraud increased to a maximum and then decreased as text length increased. M. Daller & Xue (2007) point out a similar difficulty as Guiraud can overcompensate for the falling TTR curve and actually increase in value in certain contexts. As with the majority of the other measures discussed, Guiraud scores are still strongly correlated with text length (McCarthy & Jarvis, 2007; Malvern & Richards, 2002; Jarvis, 2002).

Another TTR alternative is Mean Segmental Type-Token Ratio (MSTTR) (Johnson, 1944), which subdivides a text into even length segments, 50 tokens being the most common number, computes the TTR for each segment and then averages each of these TTRs into one final score. An advantage to MSTTR is that it is much less dependent on text length than other TTR transformations (Malvern & Richards, 2002). Additionally, there is evidence that MSTTR works well for texts over 1,000 words (McCarthy & Jarvis, 2010), which is much longer than most other measures.

However, different segment lengths are still not comparable. As an example, if one study used 100 word segments and another 50 word segments we would not be able to directly compare their results. Additionally, Malvern & Richards (2002) found that segment lengths of less than 100 tokens may produce distorted results as repetitions which occur in different segments will not be counted as such; for example, if a text repeated the word “elephant” but the repetition happened to fall in different segments it would be considered as a unique word in each. Another issue is that most texts are not evenly divisible, leading to the loss of potentially valuable information in the final, discarded partial segment. When tuning MSTTR, it can be difficult to choose an appropriate segment length which gives sensitive results while discarding the least amount of information possible. And, as with the other TTR-based measures, MSTTR is still sensitive to text length (Jarvis, 2002).

**D & Hypergeometric D**

_D_ was designed to remedy the text length issues and does so via a more complex mathematical model involved in fitting the TTR curve (Malvern & Richards, 2002). An additional common shortcoming for all previous methods that _D_ was designed to address is that:
the relationship between number of types and number of tokens for any individual sample of speech or writing is a dynamic one. That is to say, an MSTTR value represents only a single point on a curve representing the way in which TTR falls with increasing token size for that sample. (Malvern & Richards, 2002, p. 5)

In order to represent the whole curve, $D$ uses repeated random sampling from the text in order to take into account repetition that may occur spread throughout the document, also ensuring that no data is discarded. The $D$ formula

$$TTR = \frac{D}{N} \left[ \left( 1 + \frac{2N}{D} \right) - 1 \right]$$

(1.5)

attempts to model the entirety of the TTR curve for a text by varying the parameter $D$ to obtain the theoretical TTR curve which best maps to the observed curve; here $N$ corresponds to the number of tokens in a text. The original authors describe the process of obtaining a value for $D$ as:

The method for obtaining $D$ values from transcripts depends on producing a graph of the way the TTR in a given transcript falls with increasing token size within the language sample, and comparing this empirical graph with the theoretical curves obtained from the mathematical model, i.e., from the equation. The best fit between the two, obtained by adjusting the value of $D$ until the theoretical curve matches the empirical curve as closely as possible, yields a measure of the person’s vocabulary diversity represented by the value of $D$ for optimum fit. A higher $D$ represents greater diversity. (Malvern & Richards, 2002, p. 90)

To obtain these curves, $D$ or $vocd$, as McCarthy & Jarvis (2007) refer to it, uses random sampling without replacement in a text. The process is to take 100 samples of 35 tokens and compute the mean TTR of these samples, then repeat the process for samples of 36-50 tokens each. In this way we will have 16 points to which to fit our ideal theoretical curve. These means are used with the formula based around the $D$ coefficient to produce the theoretical curve that best fits the randomly sampled TTR curve. However, no inferential statistical tests are performed to see if the produced curve is close enough. To smooth out the random sampling, this procedure is repeated three times and the mean of these $D$ values is the final reported value. Using the mean TTR from the samples will approximate the mean TTR for all possible combinations of words in the text.

An example of the empirical and theoretical TTR curve obtained from varying $D$ is shown in Figure 1.2, taken from McCarthy & Jarvis (2007). Here we can see how the empirical curve,
Figure 1.2: Empirical Type-Token Ratio (TTR) curve (solid) and Best Fitting Theoretical TTR Curve (dashed) (McCarthy & Jarvis, 2007).

represented by the solid line, decreases as larger random samples are taken from the underlying text with the dashed line representing the best-fitting theoretical curve obtained by adjusting the $D$ parameter.

One significant advantage that $D$ provides over methods discussed so far is that its curve fitting procedure allows, for the first time, the comparison of texts of differing lengths (M. Daller & Xue, 2007).

Yu (2010) explored the use of $D$ to examine English language proficiency scores for written and spoken examples. He notes that while $D$ does not directly depend on text length as TTR and its derivatives do, it also may not be completely free from text length related issues; a result confirmed by McCarthy & Jarvis (2010). Yu found that $D$ explained around 11% of the variability in overall quality ratings for the written examples; however, the number of long words (which he does not define) was a better predictor than $D$. Interestingly, $D$ was less useful for predicting female writers’ scores as it only explained 5% of the variance as opposed to 34% for male writers.

More germane to this project, Yu (2010) found that $D$ was a significant predictor of spoken interview overall scores, explaining 23.4% of the variance: “In all cases D had substantial and significant correlations with the producers’ writing and speaking abilities, and overall language proficiency, and this is particularly prominent in the interview data” (Yu, 2010, p. 252). Importantly, as we will be measuring spoken texts with metrics primarily designed for use with written ones, he also found that there was a significant correlation for $D$ between the same author’s spoken and written samples.

However, Hout & Vermeer (2007) found some counterintuitive results using $D$ in a comparison
of Finnish and Swedish students with 4 and 6 years of ESL education with 7th grade native English speaking students and found that the second language speakers generally received higher scores. Jarvis (2002) is helpful as he compares the then new measure, $D$, to other existing measures like TTR, Guiraud, Herdan and MSTTR in modeling the TTR curves of teenage native Finnish, Swedish and English speakers’ English language narratives. Jarvis found Guiraud and Log TTR to be the least effective of the models described above for this purpose and so proceeds only with $D$. Similarly to Hout & Vermeer (2007), he found that high scores on $D$ were negatively correlated with holistic writing ratings. He hypothesized that this was because texts with high levels of lexical diversity “preclude the amount of lexical repetition that is necessary for the writer to maintain adequate discourse coherence” (Jarvis, 2002, p. 78). Lastly, while the original authors advocate for randomizing word order in the calculation of $D$, he does not see a similar need.

The strongest criticism of $D$ comes from McCarthy & Jarvis (2007, 2010). The original authors of $D$ used random sampling for practical, computational reasons; however, McCarthy and Jarvis see no reason not to find these TTR values for all possible word combinations. They do this using the hypergeometric distribution; wherein they are able to model the probability of any word occurring at least once in a certain sized sample of tokens. This led them to design their own version of $D$ for comparison, which they call HD-D, that estimates the probability for each type to encounter one of its tokens for a random sample of 42 words drawn from the text. They chose 42 words as it is the midpoint of the range of random samples suggested for $D$ by McCarthy & Jarvis (2007). The probabilities for all of the types are added together and used as the lexical diversity index. This hypergeometric version of $D$ gives nearly the same results as the original formulation, with correlations of around 0.97. They attribute this difference to noise induced by random sampling in the original $D$ method. Figure 1.3 overlays the HDD curve (dot-dash line) onto Figure 1.2 to provide graphical evidence for how closely HDD replicates the results of the original $D$ formula. As McCarthy and Jarvis note, “Vocd’s use of the D formula effectively only converts random-sampling TTR values to D values. The values themselves are determined by probabilities, and the D formula simply changes these values to a different scale” (McCarthy & Jarvis, 2007, p. 468).

McCarthy & Jarvis (2007) go on to say that $D$ is an idealized formula of the relationship between TTR and the number of tokens in a text as its length increases. By using curve fitting, one can use a single value of $D$ to compare texts of multiple lengths. Further correlational tests determine that $D$ and their hypergeometric version are actually measuring sums of probabilities. HDD is the sum of probabilities while “vocd-D output is essentially sums of probabilities converted to type–token ratios and, then again, from type–token ratios to a D value” (McCarthy & Jarvis, 2010,
The random sampling and curve-fitting in the original $D$ lead to somewhat imprecise and inefficient estimations of these sums. Because $D$ is ultimately a measure of the sums of probabilities, it will inevitably rise for longer texts and overcompensate for the falling TTR curve. To back this up, they show that HDD dips in value between 50 and 100 tokens but then slowly rises for the remainder of a text and $t$-test results show that this increase is significant, despite its small value.

To see if $D$, despite the fact that it does vary by text length, is in fact better than other older measures, McCarthy & Jarvis (2007) compare its results with many possibilities, including Guiraud, Herdan, and Maas, as well as basic TTR. All measures, including $D$, were significantly correlated with text length. The Maas index was one of the best performing as only 2% of its variance was explained by text length. They also determined the effective range of each of these metrics, with $D$ being one of the longest at between 100 and 400 tokens.

Another potential issue with $D$ is that long, highly diverse texts can return inaccurately high $D$ values that repeated iterations are not always able to smooth out (McCarthy & Jarvis, 2010). This suggests that HDD may be an improvement because its result is the same no matter how many times it is calculated as it does not contain a stochastic element like $D$ does.

**MTLD**

McCarthy & Jarvis (2010) developed their metric, measure of textual lexical diversity (MTLD), in order to counteract the shortcomings they observed in $D$: that it significantly varied as a function of text length and that it essentially replicates a hypergeometric distribution. Additionally, they abandoned the bag-of-words approach which strips away any textual structure. Instead, the authors
prefer to process text sequentially, as someone would read it, instead of randomly.

According to McCarthy and Jarvis, MTLD:

is calculated as the mean length of sequential word strings in a text that maintain a
given TTR value (here, .720). During the calculation process, each word of the text is
evaluated sequentially for its TTR. For example, . . . of (1.00) the (1.00) people (1.00)
by (1.00) the (.800) people (.667) for (.714) the (.625) people (.556) . . . and so forth.
However, when the default TTR factor size value (here, .720) is reached, the factor count
increases by a value of 1, and the TTR evaluations are reset. Thus, given the previous
example, MTLD would execute . . . of (1.00) the (1.00) people (1.00) by (1.00) the
(.800) people (.667) |||FACTORS = FACTORS + 1||| for (1.00) the (1.00) people (1.00)
. . . and so forth. (McCarthy & Jarvis, 2010, p. 384)

Unlike MSTTR, partial factors are also calculated for the words at the end of a text which
do not form a full factor. These partial factor scores are added on to the full factor score as a
remainder; for example, “If a text contains 4 full factors and a remainder that has a TTR of .887,
then the final factor count is 4.00 + 0.404 = 4.404” (McCarthy & Jarvis, 2010, p. 384). They settled
on a default TTR value of 0.720 to begin a new factor after finding that TTR trajectories tended
to stabilize at around that value. MTLD can be used with texts as short as 100 tokens long. To
increase consistency and accuracy, MTLD goes forward and backward through the text with the
final score being the average of the two runs. The scores themselves are computed as the number of
words divided by the number of factors.

MTLD is different from MSTTR’s segmented approach in that varying segment lengths are
selected automatically based on the text. MTLD also incorporates the idea of a stabilization point
where “neither the introduction of repeated types nor even a considerable string of new types can
markedly affect the TTR trajectory” (McCarthy & Jarvis, 2010, p. 386). At its most basic level,
MTLD calculates the average number of words required to reach this point of stabilization.

In their comparison of MTLD, D (or HDD) and Maas, McCarthy & Jarvis (2010) found that
all of these measures contain unique information. As such, different “LD indices cannot be assumed
to be assessing the same latent trait, and each index might contribute to a better understanding of
the characteristics of a text” (McCarthy & Jarvis, 2010, p. 389-90).
Lexical Sophistication

In contrast to lexical diversity which focuses on the number of unique words in a text, lexical sophistication is more concerned with the ratio of infrequently used to common words. As an example, H. Daller et al. (2003) believe strongly that lexical sophistication measures are useful as a person’s use of rare words can indicate a high level of language proficiency. Additionally, “measures which include a qualitative dimension give more insight into lexical aspects of language proficiency than purely quantitative measures like the TTR or Guiraud” (H. Daller et al., 2003, p. 203). Adding a more qualitative approach to this project will potentially provide an additional perspective alongside the more purely quantitative lexical diversity measures to ensure we are capturing all aspects of lexical richness. In this section we will focus on Advanced Guiraud, Lexical Frequency Profile (LFP) and $P_{\text{Lex}}$.

**Advanced Guiraud**

Advanced Guiraud uses the same basic formula as the previously discussed Guiraud measure, as shown in Equation 1.6, though with the number of advanced types replacing all types in the numerator, still divided by the square root of the total number of tokens

$$\frac{T}{\sqrt{N}}.$$  \hspace{1cm} (1.6)

Advanced words are generally defined as words that are not among the 2,000 most common words in the language under investigation.

To counteract the shortcomings they observed with $D$, Hout & Vermeer (2007) advocated for the inclusion of a word-list based measure, like Advanced Guiraud or their own Measure of Lexical Richness (MLR). MLR distinguishes between 9 levels of richness, although because it focuses on the language acquisition of elementary school aged children, who are not the focus of this study, it will not be discussed in further detail here. Unlike MLR, or some other lexical sophistication measures discussed below, Advanced Guiraud only distinguishes between two word levels: advanced and basic.

H. Daller et al. (2003) compared the effectiveness of basic versions of TTR and Guiraud alongside advanced versions which compare the number of advanced types (beyond 2,000 most common) to total tokens. While the basic versions were able to show some differences between their two groups of Turkish-German bilinguals, the advanced versions returned highly significant results which implies that these advanced versions can “therefore be regarded as more powerful measures
of lexical richness” (H. Daller et al., 2003, p. 212). They conclude that even with their small sample sizes, the advanced measures provide greater explanatory power and also correlate well with other tests. Advanced versions of Guiraud and TTR are able to perform better because they utilize information which is not available to purely quantitative measures.

M. Daller & Xue (2007) were interested in analyzing the English language proficiency of Chinese college students studying in the UK along with similar students who are taking English as a Foreign Language (EFL) instruction as part of their education in China. They used measures of both lexical diversity and lexical sophistication to compare the UK group, which consisted of 26 students, with 24 students in their comparison group in China. Of the measures they considered, \( D \), Guiraud, Advanced Guiraud and LFP showed significant differences between the two groups, with \( D \) and Guiraud the most strongly significant. They also computed effect sizes and determined that the most appropriate measures for their task were, in order, Guiraud, \( D \), Advanced Guiraud and then LFP. Daller and Xue hypothesize that Guiraud and \( D \) were most useful in this case because they do not require a word list which makes them more flexible to different uses. However, for the word list dependent measures they conclude that “researchers have to make sure that the word list itself is appropriate for the task” (M. Daller & Xue, 2007, p. 161).

LFP

Laufer & Nation (1995) introduce Lexical Frequency Profile (LFP). They avoid TTR-like measures because they are most interested in the size of a person’s productive vocabulary and they feel that all “a high type/token ratio may show is how well a learner can express himself with the vocabulary he knows, not what types of words he knows” (Laufer & Nation, 1995, p. 310).

LFP is designed to show the relative proportion of words from different frequency levels which come from Nation’s (1990) word lists. The original measure uses the percentage of the text which falls within four bands: the first 1,000 most common words, the second 1,000, an academic word list and those words “not-in-the-lists.” There is also a version of LFP which is organized more closely to Advanced Guiraud in that it only considers two bands: those within and those beyond the 2,000 most common words (H. Daller et al., 2003; Lu, 2012). Laufer and Nation’s overall results with the original configuration suggest that, unsurprisingly, less language proficient authors make more use of the first 1,000 words. They also found differences between low, intermediate and advanced students regarding their use of the more sophisticated bands (academic and “not-in-the-lists”). While Laufer and Nation found that texts needed to be a minimum of 200 words to produce a stable profile, others have fine-tuned that range further to be between 200 and 400 words (M. Daller &
Xue, 2007), which can be problematic if a study’s texts exceed 400 words. An additional weakness of LFP is that scores tend to be stable only for the low and intermediate proficiency authors as the advanced proficiency group’s “vocabulary apparently becomes too varied to remain stable across different samples of writing” (Laufer & Nation, 1995, p. 317).

Meara (2005) investigated Laufer and Nation’s claims regarding LFP (Laufer & Nation, 1995). Meara takes issue with the way LFP treats different aspects of a profile as completely separate when they are interrelated; for example, as the percentage of most common words used influences the percentages available for less frequent words. Additionally, most of the texts he has used in research settings do not vary in length as much as it seems Laufer and Nation expect writing samples to vary. For example, the proportions for each frequency level may only vary by a few percentage points between authors and, with texts of only one or two hundred words, these percentage point differences may represent only a few words difference. As such, he struggles to “see how the occurrence or non-occurrence of just four or five words could possibly carry the sort of informational burden that Laufer and Nation are asking them to carry (Meara, 2005, p. 34).” With these criticisms in mind, Meara performed Monte Carlo simulations to further investigate his concerns, as a rigorous comparison of a large enough group of learners would be difficult to implement.

To begin the simulations, in recognition that not all words are equally likely to occur in a text, Meara (2005) logarithmically weights words so that the most common are also the most likely to appear. The first claim under scrutiny is that LFP profiles can distinguish between people with differently sized vocabularies. In regards to this claim he finds confirmation for Laufer and Nation’s original assertion of being able to discriminate between 3,000, 6,000 and 8,000 word vocabularies though he cautions that his simulations indicate that LFP is not able to routinely tell between vocabularies which only differ by a few hundred to a thousand words.

Next he simulates whether there is a significant correlation between LFP profiles and vocabulary size which he determines to only be the case for large heterogeneous groups. As the groups become more focused around a specific vocabulary and/or the vocabulary size increases, significant correlations are rarely found. The next area to be simulated is whether two texts from the same author significantly correlate with each other. The results suggest that only with a high degree of variability in the writing and small author vocabularies are significant correlations likely. Here he also calls into question Laufer and Nation’s methods, suggesting that they took a non-significant t-test between two texts by the same author to indicate that the texts were effectively the same, resulting in a “very weak claim: it is basically a null hypothesis, with a very high probability of being confirmed by chance data” (Meara, 2005, p. 44). Meara suggests they used a similar method.
to determine if the percentages of words in each band correlate between texts. His simulations show that no matter how different two vocabularies were, one will almost never obtain statistically significant differences.

In a direct response to Meara (2005), Laufer (2005) disregards the need to use simulations in evaluating measures and also rebuts his primary operating perspective that LFP is designed to measure vocabulary size. Instead she claims that “LFP... does not tell us whether learners can produce certain words when prompted to do so, but what proportion of frequent vs. infrequent vocabulary they choose to use in their writing” (Laufer, 2005, p. 583, emphasis in original). She also disagrees with his assertion, somewhat less convincingly, that LFP should be able to discriminate between profiles with relatively small differences in vocabulary, countering that because LFP is concerned with vocabulary use, not size, it may not discriminate between learners with 500 or 1,000 word differences in their vocabulary simply because they may show few differences in how they utilize their vocabularies.

She counteracts Meara’s claim that Laufer & Nation (1995) should have calculated correlations between LFP and vocabulary in smaller, more homogeneous groups instead of for the entire sample, reiterating the view that a small increase in vocabulary knowledge does not equal an increase in vocabulary use. Instead, lexical knowledge and lexical use develop at different rates. Additionally, regarding Meara’s belief that they should have used correlations between pairs of texts by the same author, she provides the detail that they used paired t-tests instead. However, she went back to their original data, calculated the correlations and these also turned out to be significant. Lastly, to Meara’s claim that only group 1 differed from groups 2 and 3 in their original ANOVA calculations, she details that the original study performed a “comparison of the three groups on two compositions by means of twelve Duncan post-hoc tests [which] showed that only one test did not reveal a significant difference (between group 2 and 3) while eleven tests did” (Laufer, 2005, p. 587).

LFP is the most heavily researched lexical sophistication measure, with both supporting and conflicting evidence. These studies suggest that LFP profiles are most likely to be stable when the text is between 200 to 400 words and the authors have low to intermediate language proficiency. This creates problems for our study as our average story simplification is slightly beyond the ideal range at 428 words but, more importantly, all of our participants were native English speakers who had been speaking the language for presumably more than 60 years, and as such, they would certainly qualify as having advanced proficiency. With these two pieces of information in mind, we have decided to utilize Advanced Guiraud as our lexical sophistication measure.
Our final lexical sophistication measure under consideration is \texttt{P.Lex} (Meara & Bell, 2001). Unsurprisingly given Meara (2005), Meara and Bell primarily compare their new method with LFP (Laufer & Nation, 1995). Meara and Bell take issue with LFP’s ability to differentiate between texts produced by authors with different proficiency levels. Additionally, they find the 200-400 word text length window for stable LFP profiles tends to be higher than many of the texts that are provided by L2 learners, calling into question LFP’s ability to be successfully utilized with this population.

In contrast, \texttt{P.Lex} looks at the distribution of “difficult” words in a text to determine how likely these words were to appear. Their process is to first divide a text into 10 word segments, discarding any words at the end of the text that do not complete a new segment. In order to be able to compare their method directly to LFP, they used the same word lists curated by Nation (1990). They subdivide these lists into basic and advanced categories. Their basic list includes the 1000 most common words, along with proper nouns, numbers and geographical features, while any other word would be considered to be less frequent or “hard.” To analyze a text, \texttt{P.Lex} counts the number of segments containing less frequent words; for example, how many contain zero less frequent words, one less frequent word, two less frequent words and so on. As most texts contain few “hard” words, their distribution is strongly right-skewed so they use a Poisson distribution as their model. Unlike the ratios necessary for LFP, this allows them to compare texts based solely on the lambda parameter value used to attain the Poisson distribution which best matches the data. This distribution also allows them to produce stable results with shorter texts, only requiring around 120 words.

\textbf{Textual Similarity}

Word embeddings are dense, low-dimensional vector representations of words which allow each text to be represented as a bag of words in a vector space where words with similar meanings will have similar vectors (Kilimci et al., 2018). The significance of a given word in each document must also be provided a weight, with the most common methods being Boolean, term frequency and term frequency-inverse document frequency (TF-IDF). TF-IDF multiplies the frequency of each word by the inverse of its frequency in the entire document, reducing the importance of common words while boosting that of less common ones. There are several TF-IDF formulations, the one used for this project will be discussed in more detail in the Methods section. Many studies, including this one, use pre-trained embeddings such as Word2Vec, GloVe and fastText, whose goal is to capture as much
Word2Vec (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013) was developed by researchers at Google to maximize the accuracy of vector operations on words by developing new architectures that preserve the relationships among words. Older models, such as latent semantic analysis (LSA) and latent Dirichlet allocation (LDA), had previously been used for estimating a continuous representation of words. However, neural network-based methods have been shown to significantly outperform LSA. Additionally, LDA becomes increasingly computationally expensive with large datasets.

In this process, Mikolov, Chen, et al. (2013) proposed two new models: continuous bag of words (CBOW) and Skip-gram, which are depicted visually in Figure 1.4. In CBOW, so-called because “it uses continuous distributed representation of the context” (Mikolov, Chen, et al., 2013, p. 4), the order of words is not taken into account as the vectors for all words are averaged together. CBOW is similar in design to a feed-forward neural network except that the hidden layer has been removed and the projection layer is shared by all words. Its goal is to use the context around a given word (several words immediately before and after) to predict that word. On the other hand, the Skip-gram model is used to do the opposite: use a given word to predict its context. In this case, the specified word is used as input to a log-linear classifier with a continuous projection layer. Mathematically, given a sequence of training words \( w_1, w_2, w_3, \ldots, w_T \), the Skip-gram model’s goal is to maximize the average log probability.

![Figure 1.4: Word2Vec architectures. CBOW predicts a given word based on its context while Skip-gram predicts a given word’s context (Mikolov, Chen, et al., 2013).](image)
\[
\frac{1}{T} \sum_{t=1}^{T} \sum_{c,j \neq 0} (w_{t+j} | w_t)
\]

(1.7)

where \( c \) is the context and \( w_t \) the target word. While the researchers originally employed a softmax activation function, this became computationally impractical with the large training sizes employed (on the order of \( 10^5 - 10^7 \)). Instead they use negative sampling where \( p(w_{t+j} | w_t) \) is defined by

\[
\log \sigma (v'_{wO}^T v_{wI}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)}[\log \sigma (-v'_{w_i}^T v_{wI})]
\]

(1.8)

where \( \sigma(x) \) is the standard sigmoid activation function \( 1/(1 + \exp(-x)) \). Additionally, \( v_w \) and \( v'_w \) represent the input and output vector representations of \( w \). As such, the model is attempting to use logistic regression to distinguish the target word \( w_O \) from draws from the noise distribution \( P_n(w) \) with \( k \) negative samples for each data sample. Additionally, as the most frequent English words ("the", "a", "in", etc.) occur many times more frequently than the more information-rich words, they employed a subsampling approach where each word \( w_i \) in the training set was discarded with probability

\[
P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}
\]

(1.9)

where \( f(w_i) \) is the frequency of word \( w_i \) and \( t \) is a chosen threshold, typically around \( 10^{-5} \). As a result of this prediction task, the randomly initialized word vectors, with enough training time, can also eventually learn semantics. One can also utilize analogies and other semantic relationships between words to navigate the vector space, for example between a country and its capital. An interesting side effect of the Skip-gram model is that it allows for the vector addition of words, the classic example being "King" - "Man" + "Woman" = "Queen."

Mikolov, Sutskever, et al. (2013) build on the Skip-gram model to not only consider individual words but phrases as well. Their previous work showed that distributed representations of words in a vector space are able to "explicitly encode many linguistic regularities and patterns" (Mikolov, Sutskever, et al., 2013, p. 1). However, basic word representations are limited in that they are not able to represent phrases where the words together do not mean what they would otherwise mean individually; as seen, for example in company names like "Boston Globe" or places like "New York City." With this in mind, they expanded their previous work on word-based models to incorporate phrases as well. To do this, they first identify words that appear frequently together but are infrequent in other contexts. So, for example, "Buffalo Bills" would be replaced by unique tokens in the training data while a phrase like "this is" would remain the same. They formed these unique
phrases based on the unigram and bigram counts using

\[ \text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j) - \delta}{\text{count}(w_i) \times \text{count}(w_j)}. \]  

(1.10)

Here \( \delta \) is a discounting coefficient which prevents the formation of too many phrases consisting of infrequent words. Any bigrams, or combinations of two words, which pass a certain threshold are entered into the training data as phrases.

In their training process conducted on 30 billion words, they use the cosine distance between vectors to determine if the model had successfully accomplished the task set for it, which consisted of analogies (e.g. “Germany”::“Berlin”::“France”::“quick”::“quickly”::“slow”::“slowly”). Because in this task the word vectors are trained to predict the words which occur around the target word, these vectors can be thought of as representing the distribution of the word’s context. The values contained in the vectors are logarithmically related to the probabilities given by the output layer which means that the sum of any two word vectors is also related to the product of their two context distributions. This means that if words frequently appear together, for example if “Volga River” frequently appears in the same sentence as “Russian” and “river,” the sum of these last two word vectors will yield a result that is close to the vector of “Volga River.” Vector spaces like these which are able to mathematically maintain the semantic relationships between words and phrases allow one to exploit these relationships to, as we will discuss next, measure the degree of concordance between two texts which have been modeled in this way.

**Soft Cosine Similarity**

The soft cosine similarity measure was introduced by Sidorov et al. (2014). As we have seen above, the most common way to represent objects in NLP tasks is a Vector Space Model (VSM) where each object is represented as a vector of values of features. These values are numeric and can include mappings of symbolic meanings. Generally, these features are words or n-grams and the feature values are some version of TF-IDF. The standard way to compare different features is Equation 1.11, the cosine similarity

\[ \cos = \frac{X \cdot Y}{\|X\| \|Y\|} \]  

(1.11)

which considers different features, in our case words and phrases, as independent and unrelated. This means that if we were to compute the cosine similarity between two sentence vectors, we would receive a null result if the vectors had no words in common. As such, this method has its limitations.
in most NLP tasks where the components of our sentence vectors are not necessarily independent in that many words can be different and yet have similar meanings. To address this, the soft cosine method introduces new features in the VSM in the form of a similarity matrix between pairs of features. This matrix uses the Levenshtein distance, which is the number of operations (insertions, deletions, rearrangements) required to convert one string into another, to determine how similar different features are. This similarity matrix is a sparse matrix as only a small number of words or n-grams will be related to each other.

This work was built on by Charlet & Damnati (2017) who add the relation matrix to the original cosine similarity equation. In this formulation, two vectors that share a semantic similarity will return a value for the metric, even if they do not share any words. In the case where the vectors share no semantic relationship except for any shared words, the relation matrix would become the identity matrix and the result is the same as if the basic cosine similarity were computed. The resulting equation is shown in Equation 1.12. They used a pre-trained CBOW Word2Vec model to obtain the semantic relationship among words based on the basic cosine similarity.

\[
\cos_M = \frac{X^t \cdot M \cdot Y}{\sqrt{X^t \cdot M \cdot X} \sqrt{Y^t \cdot M \cdot Y}}
\]

(1.12)

where

\[
X^t \cdot M \cdot Y = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i m_{i,j} y_j
\]

(1.13)

with \( M \) being a matrix that expresses a similarity between word \( i \) and word \( j \).

Novotný (2018) put the above work into practice, developing the version of soft cosine similarity that we will ultimately utilize via Gensim Řehůřek & Sojka (2010). Sidorov et al. (2014) used basis vectors that did not directly correspond to words but instead to n-grams constructed by following paths in syntactic trees. They derived the inner product of two basis vectors from the edit distance between corresponding n-grams. Charlet & Damnati (2017) build on this work, though they used basis vectors that do directly correspond to terms. However, they derived the inner product of two basis vectors from both the edit distance and from the inner product of corresponding Word2Vec embeddings.

Review

As we have seen, there are many potential metrics available which would allow us to measure the complexity of our participant’s language in service of our first two research questions. From the
lexical diversity family, some of the best include MSTTR, HDD and MTLD, with each providing a different way to determine how varied someone’s language is, where less variety would indicate a simpler text. We can also employ lexical sophistication metrics to determine the overall complexity of words used in a text, with the most applicable of these being Advanced Guiraud.

To answer our third research question, we can employ a combination of word embeddings and soft cosine similarity. Word2Vec word embeddings provide a method by which we can turn spoken text into vector representations which encode the semantic relationships amongst words. With these vector representations in hand, we can compute the soft cosine similarity between each text. For our task, this method is an improvement over the standard cosine similarity in that the soft cosine version is able to utilize the semantic relationships between words contained within the Word2Vec embeddings.
CHAPTER 2
METHODS

Data Collection

Speech samples were collected from 11 speakers with a diagnosis of Parkinson’s disease (mean age = 68, SD = 9) and 15 neurotypical older speakers (mean age = 69, SD = 10). All participants were speakers of standard American English, over the age of 18, and passed the Universal Mini-Cog assessment. It should be noted here that the participants with Parkinson’s disease were all early in their disease progression and exhibited few speech-related impairments.

In the baseline condition, participants read a story containing 29 sentences from the Natural Stories Corpus (Futrell et al., 2018), which is included as Appendix . Following this, speakers received verbal instructions on how to rephrase statements and were provided examples of similar rephrased stimuli. In the rephrased condition, speakers were given the same stimuli and prompted to make each sentence easier to understand, using different words.

Natural Language Processing Pipeline

Before we could begin processing our texts, we had to load several Python (Van Rossum & Drake, 2009) packages into our workspace. We needed the Path function from the pathlib package (Pitrou, n.d.), the spacy package (Honnibal & Montani, 2017), we imported the lexical_div function, renamed as ld, from the lexical-diversity package (Kyle, 2020), imported the pandas library (pandas development team, 2020) as pd, imported the collections.abc base package and set collections.Iterable = collections.abc.Iterable, as well as the nbimporter package (Sturm, n.d.) which allowed us to import our c_adv_guiraud function from the Custom_Adv_G notebook.

Pre-Processing

To begin processing any of our text collections, we first needed to name and load the correct pre-trained spacy English language model, which, in our case, is en_core_web_sm. Next we created
an empty data frame which we used to store the results of our lexical richness calculations; for completeness, we decided to store the results of every metric available via the lex_div function. As each text was stored as a .txt file in different folders for each story type (original, Parkinson’s group and control group) we began our for loop by using the Path function to find each .txt file in the desired folder. In order to count the number of words in each text, we next started a counter at 0. We then used a with statement to open each file in the specified path one-by-one. The advantage of the with statement for opening files is that it will automatically close the file once it is no longer needed. Now we were able to read the .txt file, split each line of the text and, by counting the length of these lines, we were able to determine how many words are in each story.

At this point, the textual pre-processing could begin. The first step was to apply our previously loaded pre-trained spacy model to our text. We then created an empty list in which to store our tokenized and lemmatized story. Here tokenization separates the text into individual words, or tokens, while lemmatization removes all conjugation, prefixes and suffixes from a word. As an example, for the verb “shop”: “shopped”, “shopping” and “shops” would all become “shop” after lemmatization. We entered into a small for loop in order to accomplish these two tasks as well as to remove all line break characters and punctuation before appending our now pre-processed text to our previously empty list.

**Lexical Measures**

Now that our text was tokenized and lemmatized, we were able to calculate our lexical richness metrics. First, we created another empty list where we could store the results of these computations. Next, we calculated each metric that the lex_div function had available: TTR, Herdan’s index, Guiraud, Maas, MSTTR, moving average type-token ratio (MATTR), HDD, and three versions of MTLD (basic, moving average with textual wrap and bidirectional). While we did not include each of these metrics in our analysis, it was important that we did include multiple lexical diversity measures as McCarthy & Jarvis (2010) have shown that though each was developed to represent the entirety of the lexical diversity picture, in fact, each captures slightly different pieces. By including multiple of these measures, we can be more confident that our analysis is able to paint a full lexical diversity picture. As we computed each score we appended that score to our empty list and added in the number of words from before.

The final metric to compute was Advanced Guiraud which required us to build a custom function which was based after an existing function in the pelitk Python package created by the University of Pittsburgh English Language Institute (Zheng & Naismith, 2020). Customization was
deemed necessary as the pelitk metrics, including their implementation of Advanced Guiraud, were
designed to be used with second language texts; this means that the word lists they used are specific
to that task. As our purposes were different, it was important to choose a word list that reflects
the words used by native American English speakers, as recommended by M. Daller & Xue (2007),
so we chose to use the largest freely available American English language corpus, The Corpus of
Contemporary American English (COCA) (Davies, 2008-). To create this function, we first had to
import the base math package to be able to take a square root, then used a with statement to open
a saved .txt version of the COCA most common English words list, removing all white spaces and
setting all the words to lower case. We then created an empty set to hold all advanced words from a
particular text, used a for loop to move through the text, and added any words that were not in the
original word list to the advanced set. Lastly, we calculated Formula 1.6 and returned the result,
which we appended to our list with the other scores.

Now that all of the necessary calculation had been made, we matched the locations of our list
of measures with our empty named data frame in order to place each score in the correct column.
The above loop repeated for each text in the specified folder. The last step was to save our now
complete data frame as a .csv file to enable further analysis and application.

Post-Processing

Box plots were constructed for each metric to help determine the shape of the underlying
spread and to visualize the degree of simplification achieved by participants in each group. To create
the plots we first needed to import the pandas package, again as pd, matplotlib.pyplot (Hunter, n.d.)
as plt to build our plots, seaborn (Waskom, 2021) as sns to style them and PIL (Umesh, 2012) to
save them. As these plots also required all of our data to be in one data frame, we also needed
to add a “Group” column to both the Parkinson’s and control data frames specifying which group
the observation belonged to and then concatenate these two together to form one data frame which
contained all participants. We saved this data frame as a .csv and then read it in.

To create the plots, we first defined a figure and axis within our subplots and set our seaborn
style to “darkgrid” and palette to “rainbow.” We also defined our resolution to ensure the plots
would export well as images. To build the box plot itself, we specified that our x-values should come
from the “Group” column, our y-values from one of the lexical richness columns and in which data
frame these columns could be found. We also requested a mean line, for the plot to show this line
and specified the appropriate axis in which to place the plot. As we will be displaying the data
points themselves on top of the box plot, we also declared that the plot should not “showfliers,” or
outliers. To show the data points, we used seaborn’s stripplot function. The x, y and data steps are the same as for the box plot with the addition of a command to dodge, or jitter, the points to avoid overlap. We also specified the color for the points, black in this case, and again specified the correct axis. In order to include a comparison line representing the value for a given metric in the original story, we set a horizontal line to occur at the value selected from that column of the original story data frame, colored it red, and gave it an appropriate label. We then added a y-axis label specifying the appropriate measure and added a legend to display the horizontal line label.

For additional clarity, a flowchart is included in Figure 2.1 which graphically depicts the process used to move from .txt transcripts to the exploratory plots described above. In the flowchart, the Python packages and custom functions needed to perform a given step are in parentheses and for space reasons Advanced Guiraud has been abbreviated as AG.

Statistical Testing

A Shapiro-Wilk (Shapiro & Wilk, 1965) test was performed to assess each metric’s degree of normality. Additionally, as after viewing the plots we suspected we may have some non-normal distributions, a Brown-Forsythe test (Brown & Forsythe, 1974) was utilized to check whether we have homogeneity of variance between our Parkinson’s and control group for each metric. The Brown-Forsythe test was chosen as it is less sensitive to departures from normality compared with similar tests. If a given metric passed tests for normality and homogeneity of variance, t-tests were computed. A 1-sample t-test compared each group’s score with the value obtained from the original story to determine if participants’ simplifications were successful. In the case of significant results from the 1-sample t-test, a 2-sample t-test was used to see if either group was more successful at
the task than the other. Each of the above tests were computed using pre-built functions within
the SciPy package (Virtanen et al., 2020). However, should a given metric fail the Shapiro-Wilk or
Brown-Forsythe tests, then non-parametric measures were utilized instead. In this case, a 1-sample
Wilcoxon rank sum test (Wilcoxon, 1945) took the place of the 1-sample t-test and a 2-Sample
Mann-Whitney U test (Mann & Whitney, 1947) replaced the 2-sample t-test. The Mann-Whitney
test is also built into the SciPy package, though the Wilcoxon rank sum test was conducted using
the test_wilcoxon_os package (Stikpet’s functions, n.d.). Lastly, to assess effect size, Cohen’s d was
calculated manually utilizing a custom Numpy (Harris et al., 2020) function.

Textual Similarity

Similar to previous steps, our first step in computing soft cosine similarity scores between our
simplified and original texts was to load all necessary Python libraries and functions. As before, we
needed to once again import spacy, pathlib as Path and pandas as pd. In addition, we needed to
download several packages and functions from the Gensim library (Rehůřek & Sojka, 2010). These
are: from gensim we needed to import corpora and models, then from gensim.corpora we imported
Dictionary and from gensim.models selected TfidfModel, Word2Vec and KeyedVectors. We also
imported api from gensim.downloader and lastly from gensim.simplesimilarity we imported SparseTerm-
SimilarityMatrix and WordEmbeddingSimilarityIndex.

Once all of the above were loaded, we again initialized the same spacy English language model
as before and then downloaded the pre-trained Word2Vec model trained on the 3 billion word Google
News data set, resulting in 3 million, 300-dimensional word vectors. A model this large took several
hours to download so we saved the resulting vectors themselves for future use. Our last setup step
was to create a similarity index using our stored word vectors.

To begin the next portion, we first processed our original story using a very similar process
as described above but with the addition, once the text was read in, tokenized and lemmatized, of
a dictionary creation step which built word ↔ id mappings between the words in this story. Lastly,
we utilized our dictionary to convert our story into a bag-of-words corpus.

In order to compare our Parkinson’s or control group stories to the original, we first created
an empty data frame to hold our similarity scores once they were computed. Then, within a for
loop, we read in and process our group stories as before, though with additional steps added within
the loop to calculate the similarity. Once the story had been tokenized and lemmatized, we first
imported the dictionary we created from the original story and then created another dictionary and
A bag-of-words corpus for this particular retelling. We were then able to add the words from the simplified story to our existing dictionary and combine our two corpora which we used to perform TF-IDF weighting of each term in our combined corpus. Gensim utilizes Equation 2.1 to do this weighting,

$$w_{i,j} = f_{i,j} \times \log_2 \frac{D}{DF_i} \quad (2.1)$$

where \( w \) represents the TF-IDF weighting of term \( i \) in document \( j \), \( f \) is the frequency, \( D \) is the number of documents in the corpus and \( DF \) is the number of documents containing term \( i \). Once we had a TF-IDF representation, we subset the portions of that overall model for both the original and group story for ease of use when computing the similarities.

Next we were able to begin the actual soft cosine similarity component of the loop. To begin with, we used the SparseTermSimilarityMatrix function to create our similarity matrix between the two stories, using the previously created term similarity index, our dictionary, and the just created TF-IDF weights. A note at this point: As the participants simplified the original story on a sentence-by-sentence basis, we also computed the similarity between each sentence of a given retelling and the equivalent sentence in the original story. To perform the comparison, we first initialized a sum counter at 0 and created an empty list to eventually store the similarity score for this retelling. We then entered into another for loop to compute the sentence by sentence soft cosine similarity score, adding each sentence’s score to our sum counter, exiting the loop once all of the sentences have been compared in this way. We then took the average of the summed similarities and appended this value to our previously empty data frame. This process then repeated for each story in the chosen group and followed the same steps for the other group. Lastly, we exported the completed data frame as a .csv file for future use. In the process of computing these scores, it was discovered that two participants, one in each group, skipped a sentence in the process of simplification. This meant that for these two participants there was no longer concordance between the retold and original stories which would allow for a direct sentence comparison; as such, these two participants were excluded from the similarity analysis.

To provide benchmarks for the soft cosine similarity scores, we created our own simplification of the original story, included in Appendix B, which only replaced individual words with appropriate synonyms and did not attempt to otherwise change the structure of the story in any way. An additional benchmark was provided in the form of “The Princess and the Pea,” an unrelated but similar length fairy tale. Similarity scores were calculated for these stories using the same method as previously described.
In order to visualize the spread of each group’s similarity scores, and to compare them with our benchmarks, we constructed box plots via the matplotlib.pyplot and seaborn libraries and saved the images utilizing the PIL package. To create the box plots, we first import the necessary libraries and read in our similarity score .csv files. We then defined our figure and axis to be subplots within our figure, set our seaborn style (“darkgrid”) and palette (“deep”) and specified our figure resolution. To build the box plots themselves, we selected the appropriate, and only, column from each group’s similarity data frame, set our patch_artist to be True in order to display our legend, specified that we desired a mean line, that we wanted to show that mean line and declared that we wanted each plot to take up half of the figure. Next we created a horizontal red line at the similarity value for the “Synonym-Only” benchmark and a blue line for “The Princess and the Pea”’s similarity value. Lastly, we specified the appropriate x and y-axis labels, extended the y-axis limit to fully include “The Princess and the Pea,” specified our legend placement and saved the resulting image.

As we suspected there may be a relationship between a participant’s similarity and lexical richness scores, we also conducted ordinary least squares regression analyses for each combination of lexical richness measure and similarity. To construct the preliminary and result plots, we needed to import Numpy as np, pandas as pd, seaborn as sns, matplotlib.pyplot as plt and SciPy.stats as stats. The regression modelling itself, along with portions of the diagnostic plots, necessitated several functions from the statsmodels library (Seabold & Perktold, 2010). These included, statsmodels.api as sm, ProbPlot from statsmodels.graphics.gofplots, ols from statsmodels.formula.api, variance_inflation_factor from statsmodels.stats.outliers_influence and lowess from statsmodels.nonparametric.smoothers_lowess. We also needed to import OneHotEncoder from sklearn.preprocessing.

Once our libraries and functions were loaded, we next needed to combine our Parkinson’s and Control data frames, excluding the two participants for whom we were unable to calculate a similarity score, with the addition of a new “Group” column which specified a particular participant’s group status. We then applied the “OneHotEncoder” to the “Group” column in order to turn our categorical variable into one that was appropriately numerically represented. Our regression results themselves were computed using the ols function, wherein we specified a formula to predict a specific lexical richness metric with one’s similarity score and group status, then fit the model. We will not go into detail regarding the creation of the diagnostic regression plots, though each plot can be found in Appendix D.

To visualize the results we created scatter plots with overlaid regression lines, one for each group status. To begin with, we first needed to specify a Numpy “linspace” which spanned all
possible similarity scores. We then created a 2x2 array of subplots, defined our desired resolution and specified our seaborn style and palette as before. Next, we created a seaborn scatter plot wherein we specified our x values to be the similarity scores, our y values to be our desired lexical richness measure, specified that the hue of the points should be done based on group status, defined our points to be blue or red accordingly and declared that we did not want this plot to create a legend. To create the lines, we plotted the formula and coefficients extracted from our linear model over the previously created “linspace,” one line for each group, and labelled our control group blue and Parkinson’s group red to match the points in our scatter plot. We then defined appropriate x and y-axis labels and repeated this process for each lexical richness measure. Our last step was to define the placement of our legend and save the resulting image.

Those wishing to perform similar analyses to any of the above, can find example code for this project at: https://github.com/bcnalley/lexical-simplification-nlp.
CHAPTER 3
RESULTS

Exploratory Plots

Exploratory box plots which compare the Parkinson’s and control groups for each metric are included in Figure 3.1. A note about the plots: the horizontal red line is the value for each metric in the original story; within the boxes the solid line represents the group median while the dashed line is the mean. When viewing the box plots, we first notice that the vast majority of participants were able to obtain scores for each measure which were lower than in the original story. It is also apparent that each of the lexical diversity measures (MSTTR, HDD and MTLD) have multiple potential outliers which fall beyond the whiskers of our plots, mostly within the control group, and tend to be left-skewed. We also observe that the mean for each of the control group metrics tends to be higher than that of the Parkinson’s group, though there is considerable overlap.

In the top row, which shows the plots for MSTTR and HDD, we see very similar patterns in both where the potential outliers on the low end of the control group appear to be pulling the mean lower and introducing some left skew. A different pattern is apparent with our MTLD plot, located in the lower left corner, where now we have potential outliers in both groups. Interestingly though, the Parkinson’s group outliers are on the high end of the scale, while they remain on the low end for the control group. Other than the outliers, the main bodies of each group’s spread seem to be more evenly distributed than for the previous two metrics. Lastly, in the lower right corner we see our first metric without any potential outliers, though again we have a small minority in the control group who have scored higher than the value in the original story. These spreads also appear to be relatively normal, certainly more so than the lexical diversity measures.

Preliminary Analysis

Table 3.1 displays the mean and, in parentheses, the standard deviation for each lexical diversity measure. In Table 3.2 we have the results of the normality and homogeneity of variance
Figure 3.1: Box plots for each lexical richness measure with observations overlaid. Proceeding clockwise beginning in the upper left-hand corner these are: Mean Segmental Type-Token Ratio (MSTTR), Hypergeometric $D$ (HDD), Advanced Guiraud and Measure of Textual Lexical Diversity (MTLD).
Table 3.1: Means and (Standard Deviations) for each lexical richness measure

<table>
<thead>
<tr>
<th>Lexical Richness Measure</th>
<th>Control Group</th>
<th>Parkinson’s Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTTR</td>
<td>0.69 (0.04)</td>
<td>0.67 (0.04)</td>
</tr>
<tr>
<td>HDD</td>
<td>0.74 (0.04)</td>
<td>0.72 (0.04)</td>
</tr>
<tr>
<td>MTLD</td>
<td>39.99 (8.13)</td>
<td>35.71 (6.88)</td>
</tr>
<tr>
<td>Advanced Guiraud</td>
<td>0.96 (0.35)</td>
<td>0.85 (0.25)</td>
</tr>
</tbody>
</table>

Table 3.2: Normality and homogeneity of variance test results for each lexical richness measure. Each column includes the test statistic and (p-value).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Control Shapiro-Wilk</th>
<th>Parkinson’s Shapiro-Wilk</th>
<th>Brown-Forsythe</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTTR</td>
<td>0.77 (0.002)</td>
<td>0.94 (0.482)</td>
<td>0.05 (0.818)</td>
</tr>
<tr>
<td>HDD</td>
<td>0.78 (0.002)</td>
<td>0.91 (0.231)</td>
<td>0.06 (0.813)</td>
</tr>
<tr>
<td>MTLD</td>
<td>0.88 (0.044)</td>
<td>0.89 (0.151)</td>
<td>0.37 (0.551)</td>
</tr>
<tr>
<td>Advanced Guiraud</td>
<td>0.93 (0.226)</td>
<td>0.88 (0.097)</td>
<td>0.89 (0.356)</td>
</tr>
</tbody>
</table>

checks for each of our lexical richness measures. As we will be preforming multiple statistical tests for each metric, the Benjamini-Hochberg (Hochberg & Benjamini, 1990) procedure

\[(i/m) \ast \alpha \tag{3.1}\]

was utilized to adjust significance levels accordingly. Here \(i\) represents the p-value rank and \(m\) is the number of tests. In this procedure, the largest \(p\)-value which is less than the critical value given by the formula is the demarcation for statistical significance, so that this instance and all lower ranked ones are considered significant. We used \(\alpha = 0.05\) and grouped tests by function so that \(p\)-values were ranked separately for the Shapiro-Wilk, Brown-Forsythe, 1-sample wilcoxon rank sum and 2-sample Mann-Whitney \(U\) tests as well as for the regression analysis. A note on the format: the test statistic is reported first with the \(p\)-value in parentheses, which is in bold font when significant following the Benjamini-Hochberg procedure. We can see a general pattern for the lexical diversity measures in that two of them (MSTTR and HDD) failed the Shapiro-Wilk test, meaning that their control groups are not normally distributed enough to be able to employ parametric tests. However, each of their Brown-Forsythe tests does indicate homogeneity of variance between each metric’s Parkinson’s and control groups. As a result, Advanced Guiraud, our lexical sophistication measure, is the only one which fulfills all the assumptions to allow for parametric tests. We can see that both Advanced Guiraud groups are approximately normally distributed and also maintained homogeneity of variance between groups.
Table 3.3 shows the statistical test results for each of our lexical richness measures. As half of our measures failed the Shapiro-Wilk test, we will proceed with non-parametric tests for all metrics in order to ensure that we are proceeding with the appropriate caution. These non-parametric tests were: 1-sample Wilcoxon Rank Sum Tests for each group and 2-sample Mann-Whitney $U$ Tests to compare groups.

We see a pattern in that for all metrics our 1-sample tests indicated that both groups were able to simplify their retelling of the original story at a highly statistically significant level. However, the 2-sample tests did not find any significant between group differences on their success at the task. We also found mild-moderate (HDD and Advanced Guiraud) to moderate (MSTTR and MTLD) effect sizes according to Cohen’s $d$, indicating that the average lexical richness scores for our groups were separated by between 0.358 and 0.561 standard deviations.

### Textual Similarity

The mean soft cosine similarity score for the Parkinson’s group was a 0.511 with a standard deviation of 0.234, while the control group was slightly higher with a mean of 0.603 and standard deviation of 0.226. To provide some context to these numbers, we also computed the similarity between the original story and two alternatives. The first alternative was a version of the original which only used synonyms, as appropriate, and did not attempt to change any of the story’s structure. Second was an unrelated story, “The Princess and the Pea” (Andersen, 2020). The synonym-only retelling can be found in Appendix A with “The Princess and the Pea” in Appendix B. The similarity scores for the synonym version of the original story and “The Princess and the Pea” were 0.618 and 0.111, respectively. These similarity scores were computed sentence-by-sentence, using the same method as when comparing the participants’ simplifications.

In Figure 3.2, we can see a graphical representation of the similarity metric. As in previous
Figure 3.2: box plots of Soft Cosine Similarity scores for Parkinson’s and control groups with comparison lines for a synonym-only retelling of the original story and “The Princess and the Pea” box plots, the dashed line within the box itself represents the group mean while the solid line represents the median. However in these plots, the solid red line indicates the similarity score for the synonym-only version of the story and the blue line indicates the same for “The Princess and the Pea.” We can see that, at a minimum, all participants’ similarity scores are higher than that for the “The Princess and the Pea,” indicating that their retellings were more similar to the original than our unrelated fairy tale. In addition, several participants’ simplifications scored higher than the synonym-only version; in fact, the control group mean is only slightly below this benchmark. Both group plots appear to be relatively symmetric with no obvious outliers.

As we suspected that there may be a relationship between a given participant’s soft cosine similarity and lexical richness scores, we conducted an ordinary least squares regression analysis. The results are displayed in Table 3.4, scatter plots with regression lines overlaid are in Figure 3.3, while the regression diagnostic plots are contained in Figures 4.1 - 4.4 in Appendix D. From the table we can see that group status is not a significant predictor for any metric; conversely, similarity scores are significant for each after applying the Benjamini-Hochberg procedure. These results suggest that we have a significant positive linear relationship between similarity and lexical richness. This means that as participants’ similarity scores increase, so do their lexical richness scores. Despite the differences in scale for the coefficients, the plots show us that the slope of these lines are relatively
Table 3.4: Ordinary Least Squares Regression results using group and soft cosine similarity score to predict lexical richness. *P*-values which are bold are significant following the Benjamini-Hochberg procedure.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Model Component</th>
<th>Coefficient</th>
<th><em>P</em>-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTTR</td>
<td>Parkinsons(T)</td>
<td>-0.018</td>
<td>0.261</td>
</tr>
<tr>
<td></td>
<td>Similarity</td>
<td>0.098</td>
<td><strong>0.009</strong></td>
</tr>
<tr>
<td>HDD</td>
<td>Parkinsons(T)</td>
<td>-0.009</td>
<td>0.515</td>
</tr>
<tr>
<td></td>
<td>Similarity</td>
<td>0.104</td>
<td><strong>0.002</strong></td>
</tr>
<tr>
<td>MTLD</td>
<td>Parkinsons(T)</td>
<td>-1.686</td>
<td>0.504</td>
</tr>
<tr>
<td></td>
<td>Similarity</td>
<td>23.624</td>
<td>&lt; <strong>0.001</strong></td>
</tr>
<tr>
<td>Advanced Guiraud</td>
<td>Parkinsons(T)</td>
<td>-0.077</td>
<td>0.490</td>
</tr>
<tr>
<td></td>
<td>Similarity</td>
<td>0.813</td>
<td><strong>0.003</strong></td>
</tr>
</tbody>
</table>

constant between measures and all have roughly equal separation between groups, with MSTTR being a possible exception.
Figure 3.3: Scatter plots comparing lexical richness and textual similarity scores for each participant with linear regression lines overlaid. The red line represents the regression equation for the Parkinson’s group, while the blue line is that for the control group, while the observations for each group are in the same colors. Proceeding clockwise beginning in the upper left-hand corner these are: Mean Segmental Type-Token Ratio (MSTTR), Hypergeometric $D$ (HDD), Advanced Guiraud and Measure of Textual Lexical Diversity (MTLD).
CHAPTER 4
DISCUSSION

The goal of this project was to determine if people, especially those with Parkinson’s disease, are able to simplify their language when prompted. To test this, both a group of people with early Parkinson’s disease and age-matched neurotypical controls were provided with a relatively complicated story that they were asked to simplify. We used four metrics to measure simplification: Three under the umbrella of lexical diversity with one lexical sophistication measure. One sample tests for all four metrics were highly significant, indicating that, overall, both groups were able to successfully simplify their speech. This provides strong preliminary evidence for the efficacy of language simplification as a communication strategy.

Unsurprisingly, owing partially to the small sizes of both groups, we did not find any significant between group differences, though the Parkinson’s group appeared to be slightly more successful at the task. Translating these differences into practical terms, as measured by MSTTR, the control group used roughly two fewer types per 50 word segment compared with the original story while the Parkinson’s group used three fewer. Likewise, according to MTLD, the control group’s TTR stabilized 8 words sooner than in the original story and the Parkinson’s group 12 words sooner. As far as “advanced” words are concerned, while the original story featured 37, the control group used on average 26 and the Parkinson’s group 23.

Comparing our metrics, as we can see in Figure 3.1, overall our participants were most successful at simplifying their lexical sophistication as the average Advanced Guiraud score of the two combined groups was 36.50% lower than in the original story. Meanwhile they were least successful according to HDD; here participants’ retellings only increased the probability that any word would encounter another example of itself by 4.85%. Taken together, this suggests that when attempting to simplify their language, participants were more likely to replace “advanced” words with more common ones than to replace unique words with previously used ones.

Our box plots also show that there appear to be potential outliers within the control group for each of the lexical diversity measures. These outliers could potentially be explained by socioeconomic
or educational differences which are intended to be explored further in future studies. While this
data was collected as part of the study protocol, it was not available at the time of this analysis.

Regarding our final research question, all of our participants were able to maintain enough of
the context present in the original story that their soft cosine similarity scores exceeded the unrelated
fairy tale benchmark set by “The Princess and the Pea.” This indicates that their simplified stories
were at least more similar than not to the original story. Additionally, several participants in
both groups exceeded the similarity score set by the synonym-only retelling, indicating that their
retold versions maintained a great many of the details present in the original story. Interestingly,
the scatter plots and regression analyses presented in Figure 3.3 and Table 3.4 indicated an overall
strong positive linear relationship between participants’ lexical richness and textual similarity scores.

As some of the participants’ similarity scores were closer to that of the unrelated fairy tale
than expected, the lowest scoring retelling from each group was subjectively assessed to determine
how well each was able to maintain the meaning of the original story.

The lowest scoring participant overall was in the Parkinson’s group with a similarity score
of 0.191. This participant appears to have oversimplified, to the point where their retelling would
be difficult to follow if one was not familiar with the original story. About halfway through their
retelling they also began repeating themselves within sentences and many of their sentences do not
entirely make sense on their own. For example, “Someone no expected someone no expected took
the lead.” Additionally, they added a power dynamic interpretation in the final few sentences that
does not necessarily match what is present in the original story: “Those in power had been caught
flat footed / Many people were outraged / The power structure was upset / Although some people
sensed the chance to to get ahead / There was there was an opportunity for new players to to to
take power / Leadership was up for grabs.” These observations suggest that the combination of this
participant potentially experiencing some difficulty in selecting or uttering their intended words and
their addition of new material were the primary culprits for their low soft cosine similarity score.

Within the control group the lowest similarity score was a 0.289. In reading through this
participant’s transcript, it appears as if this participant, overall, accomplished the task as requested.
Their primary oversimplification was almost exclusively using pronouns, and often the same ones,
to refer to different birds without specifying to whom they applied. For example, “So he thought
he had it in the bag that he was going to be king so he wasn’t worried about it / He was excited
he didn’t think he’d be king but maybe there was a chance.” In both sentences “He” refers to a
different bird: the first “He” is the Eagle and the second is the Hawk. Similar to the other retelling
so examined, this version is so simplified that it is difficult to follow without already knowing the
source material. However, it is clear when comparing the two that this second participant achieved a higher similarity score than the first.

**Limitations & Future Work**

The largest limitation of this study is the small number of participants, all recruited from a mid-size college town in the US Intermountain West. While we think it likely that our results will generalize to the population at large, more participants would need to be recruited, including those of more diverse backgrounds and locations, to confirm this hypothesis. An additional benefit of a larger pool of participants is that it would provide us with more insight as to whether there are any between group differences for this task.

An additional limitation related to the lexical similarity measurement. As the participants were asked to rephrase the original story one sentence at a time, the similarity was calculated between each rephrased and original sentence. However, for unknown reasons, two participants, one in each group, omitted one sentence from their retelling. This led to a mismatch between the retelling and original story, requiring their simplifications to not be included in this part of the analysis.

Were there more time to continue with this project, we would have liked to incorporate a measure of discourse structure into the analysis. Discourse structure takes into account word ordering, phrase adjacency and the connection between segments to determine text coherence and complexity (Davoodi & Kosseim, 2016). As it stands, this project takes into account multiple metrics of word choice-related complexity and also measures how similar the retold story was to the original example. A discourse structure analysis would provide both another potential measure of complexity as well as a way to build on the similarity score to also assess the coherence of each simplification.

Although this study provides strong preliminary evidence of the ability of participants to simplify their language, this does not confirm the end-goal of this communication strategy (that simplifying language will make their speech easier for listeners to understand). Subjective listening tests to measure improvement in intelligibility following language simplification is crucial to the advancement of this research. If similar questions to these cannot be answered affirmatively then these results are interesting but of little practical use. Additionally, many factors can play a role in one’s ability to simplify their language, besides a Parkinson’s disease diagnosis, which need to be determined and explored further.
REFERENCES


Fletcher, A., McAuliffe, M., Lansford, K., Sinex, D., & Liss, J. (2017, November). Predicting intelligibility gains in individuals with dysarthria from baseline speech features. *Journal of Speech and Hearing Disorders, 60*(11), 3043–3057. (Funding Information: This work was supported by a Fulbright New Zealand Graduate Award, granted to Annalise R. Fletcher. Publisher Copyright: © 2017 American Speech-Language-Hearing Association.) doi: 10.1044/2016_JSLHR-S-16-0218


Once upon a time the birds took it into their heads that they would like a master and that one of their number must be chosen king. A meeting of all the birds was called and though they understood the birds who were from the most distant lands would be unable to come many birds came from far away meadows and woods. The eagle who already thought himself the de facto king arrived fashionably late. It was a hawk who was most excited about the meeting because he was the dark horse for king. The small fry came too and the robin the bluebird the owl the lark and the sparrow who had only a chance in a million to be king were all present at the meeting. The cuckoo who was almost not invited because his call so annoyed the other birds came too. It was the very little bird who had no name at all however that would end up overturning the balance of power among the birds. That there would be great confusion and noise among the birds at the meeting was to be expected given the sheer numbers of birds that have gathered. There was piping hissing and clacking but finally it was decided that the bird that could fly the highest should be king. The little bird laid low near the eagle at first but the eagle did not notice the bird hopping onto his back right as the competition was about to commence. Into the air in a great flock all the birds flew when the signal was given. The air was full of dust and it seemed as if a black cloud were floating over the field. You could hear the birds chirping and flapping from fields that were miles away. The little birds that soon grew tired fell back quickly to earth. The fact that the larger ones held out longer and flew higher and higher but the eagle flew highest of any surprised no one. Could anyone stop the eagle that seemed to be flying straight into the sun. The other birds gave out one by one and when the eagle saw this he thought What is the use of flying any higher. This victory is in the bag and I am king. Then the birds below called with one accord Come back come back. It is you who must be our king because no one can fly as high as you. Except me cried a shrill shrill voice and out of the blue the little bird without a name rose from the eagle’s back where he had lain hidden in the feathers. His guile was the ace up his sleeve and he laughed to himself at how easy it had been to outwit the other birds. Higher and higher he mounted until he was lost to sight and then folding his wings together he sank to earth crying shrilly I am king I am king. The eagle tricked by the little bird had not sensed the bird in his feathers and so had not at all expected the bird to come flying out like that. The birds back on earth were all up in arms. You and not the eagle our king the birds cried fuming with anger. You have done this by breaking every rule in the book so we will not have you who are simply tricky and cunning to reign over us. The bird without a name then decided to clear the air and said Then let everyone start with a clean slate and perform a new challenging task. Then we can decide who should be the real king.
Once upon a time the birds took it into their heads that they would like a ruler and that one of their own must be chosen monarch. A gathering of all the birds was convened and though they realized the birds who were from the farthest lands would be unable to come many birds came from distant fields and forests. The eagle who already thought himself the unofficial king came fashionably late. It was a hawk who was most interested in the meeting because he had an outside chance to be leader. The smallest came too and the robin the bluebird the owl the lark and the sparrow who had the smallest chance to be leader all came to the gathering. The cuckoo who was almost not welcome because his voice so upset the other birds also was there. It was the very little bird without a name however that would end up changing the balance of power among the birds. That there would be a lot of confusion and ruckus among the birds at the gathering was not surprising with how many birds that were there. There was piping hissing and clacking but at last it was decided that the bird who could rise the farthest would be leader. The small bird laid down close to the eagle at the beginning but the eagle did not see the bird jumping onto his back right as the contest was about to start. Into the sky in a great group all the birds rose when the alarm was given. The sky was full of dirt and it looked as if a dark cloud was hovering above the meadow. You could hear the birds chirping and flapping from yards that were miles away. The small birds that quickly were exhausted descended back rapidly to ground. The fact that the bigger ones lasted longer and rose farther and farther but the eagle rose farthest of all shocked no one. Would anyone prevent the eagle that looked to be rising directly into the sun. The rest of the birds retired one by one and when the eagle observed that he thought what is the point of rising any further. This triumph is in the bag and I am ruler. Then the birds underneath yelled with one voice return return. It is you who should be our ruler since no one can rise as far as you. Except me yelled a high high voice and out of nowhere the small bird without a name stood from the eagle’s back where he had reclined concealed in the feathers. His sneakiness was the ace up his sleeve and he chuckled to himself at how simple it had been to outsmart the other birds. Farther and farther he rose until he was invisible and then bending his wings together he dropped to ground yelling highly I am ruler I am ruler. The eagle misled by the small bird had not noticed the bird in his feathers and so had not at all foreseen the bird to come flying out like that. The birds on ground were all up in arms. You and not the eagle our ruler the birds yelled stewing with anger. You have done this by destroying every law in the book so we will not have you who are only sneaky and conniving to rule over us. The bird with no name then thought to clear the air and stated then let all begin with a fresh slate and undertake a new difficult challenge. Then all can decide who will be the true ruler.
APPENDIX C

The Princess & the Pea

There was, once upon a time, a prince who wanted to marry a princess, but she must be a true princess. So he travelled through the whole world to find one, but there was always something against each. There were plenty of princesses, but he could not find out if they were true princesses. In every case there was some little defect, which showed the genuine article was not yet found. So he came home again in very low spirits, for he had wanted very much to have a true princess. One night there was a dreadful storm; it thundered and lightened, and the rain streamed down in torrents. It was fearful! There was a knocking heard at the palace gate, and the old king went to open it. There stood a princess outside the gate; but oh! What a sad plight she was in from the rain and the storm! The water was running down from her hair and her dress into the points of her shoes and out at the heels again. Yet she said she was a true princess. "Well, we shall soon find that out!" Thought the old queen. But she said nothing, and went into the sleeping room, took off all the bedclothes, and laid a pea on the bottom of the bed. Then she put twenty mattresses on top of the pea, and twenty eiderdown quilts on the top of the mattresses. This was the bed in which the princess was to sleep. The next morning she was asked how she had slept. "Oh, very badly!" Said the princess. "I scarcely closed my eyes all night! I am sure I don’t know what was in the bed. I laid on something so hard that my whole body is black and blue. It was dreadful!" Now they perceived that she was a true princess, because she had felt the pea through the twenty mattresses and the twenty eiderdown quilts. No one but a true princess could be so sensitive. So the prince married her, for now he knew that at last he had gotten hold of a true princess. The pea was put into the Royal Museum, where it is still to be seen, if no one has stolen it. There that is a true story.
Figure 4.1: Mean Segmental Type-Token Ratio (MSTTR) Linear Regression Diagnostic Plots
Figure 4.2: Hypergeometric $D$ (HDD) Linear Regression Diagnostic Plots
Figure 4.3: Measure of Textual Lexical Diversity (MTLD) Linear Regression Diagnostic Plots
Figure 4.4: Advanced Guiraud Linear Regression Diagnostic Plots