Logan Tidstrom

University of Michigan - Ann Arbor

Keystroke Logging in Second Language Writing Research

RWTH Aachen University
Department of English Linguistics
Elma Kerz, Marcus Ströbel

UROP – Undergraduate Research Opportunities Program

Aachen
July 19, 2019

Table of Contents:

1.1 Abstract

The study of writing is an important tool to examine the underlying cognitive processes of language production. However, a lot of writing research focuses on written text as an end product. Conversely, keystroke logging provides data on the writing process as a text develops. Keystroke logging data was collected on texts produced by a group of university students. 660 subjects participated, producing a total of 7012 files, which were collected using the Etherpad text editor for keystroke logging. The keystroke logging data was then compared with data regarding text complexity to analyze whether a correlation exists between the writing process and the complexity of a text.

1.2 Introduction

Keystroke logging is a valuable tool in writing research because it offers a view of a writer’s fluency and flow that may relate to underlying cognitive processes in text production [1]. While much writing research has been conducted to examine writing from the perspective of completed written text, keystroke logging provides a different perspective by offering data on the actual writing process. By way of timestamps, we can see the exact steps a writer took to produce a final text. Instead of only looking at the final text, the goal of this study is to look into the *process* of text production to determine what connections exist between the manner in which a text is produced and the complexity of the text output.

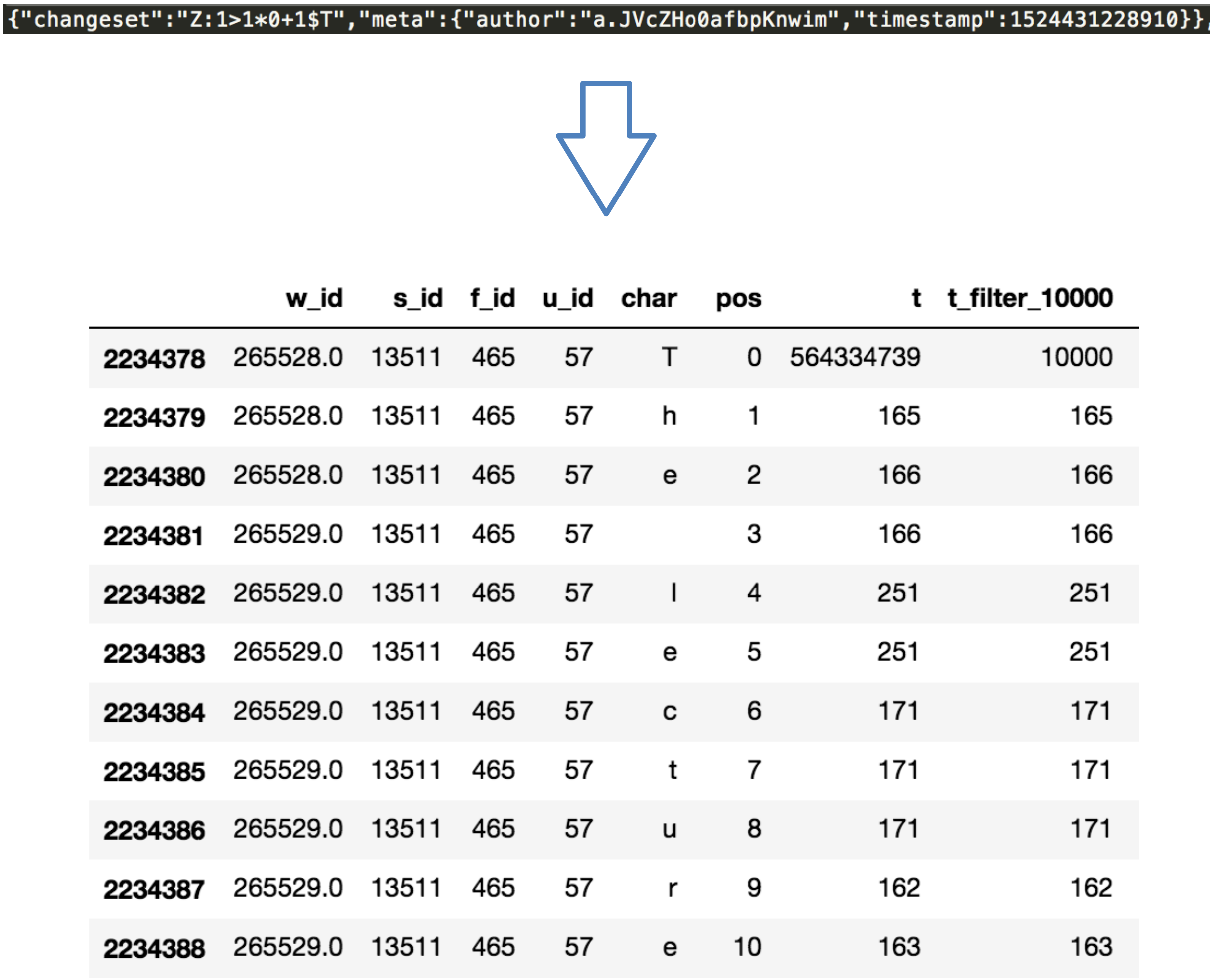
Much keystroke logging research tends to focus on the difference between “bursts” (fluid typing) and “pauses" (breaks in typing) to evaluate the writing process [2]. Studies have shown that pauses during writing can be physiological (i.e. fatigue) but are more often cognitive, related to planning or revising [3]. A challenge facing keystroke logging research is that although the data provides proof of when and where a pause occurs, it is difficult to determine the underlying reason for a pause [4]. Some researchers choose to include Talk-Aloud Protocols (TAPs) in their methods, in which writers verbally explain what they are doing as they write [5]. Unlike TAPs, keystroke logging allows for similar data collection without interrupting or altering the natural writing process. For the clearest depiction of the writing process, keystroke logging is the method of data collection in this study.

2. Project Description

2.1 Data Set

Text sources were collected from 660 university students writing between 6-13 weekly “Learning Journals” in English, summarizing lecture material. In total, 7012 text files were produced. These files had an average length of 529 words, and the average total typing time taken to complete the text was 35 minutes. The subjects were asked to submit these files using Etherpad, which is a text editor tool that tracks the writers’ keystrokes and provides “changeset data” representing the type and timestamp of each character typed. An example of the changeset data can be seen in Figure 2.1.1.

The first step was to preprocess the data. First, we used the Stanford CoreNLP sentence splitter to divide each text file by sentence. We then used Node.js to handle the Etherpad changeset data and set it up in a format that was easier to understand. This involved using Pandas data frames (Python) to store the data in five layers: characters, words, sentences, files, and users. From the changeset data, we developed a keystroke data frame that stored the type, timestamp, and position of each character typed (see Figure 2.1.1).



```
{"changeset":"Z:1>1*0+1$T","meta":{"author":"a.JVcZHo0afbpKnwim","timestamp":1524431228910}}
```

	w_id	s_id	f_id	u_id	char	pos	t	t_filter_10000
2234378	265528.0	13511	465	57	T	0	564334739	10000
2234379	265528.0	13511	465	57	h	1	165	165
2234380	265528.0	13511	465	57	e	2	166	166
2234381	265529.0	13511	465	57		3	166	166
2234382	265529.0	13511	465	57	l	4	251	251
2234383	265529.0	13511	465	57	e	5	251	251
2234384	265529.0	13511	465	57	c	6	171	171
2234385	265529.0	13511	465	57	t	7	171	171
2234386	265529.0	13511	465	57	u	8	171	171
2234387	265529.0	13511	465	57	r	9	162	162
2234388	265529.0	13511	465	57	e	10	163	163

Figure 2.1.1. *An example of the handling of "changeset" data. The changeset (top) shows all data pertaining to a single character "T" typed at timestamp 1524431228910. The data was stored in Pandas data frames (bottom). This is a slice of the keystroke data frame, which stores the time, type, and position of each character typed, as well as the word, sentence, file, and user id numbers each character belongs to.*

Additionally, preprocessing the data was very important to access the data that most accurately represents the writing process. Therefore, it was essential to clean the data by getting rid of any text that was copy/pasted into Etherpad. If a subject copy/pasted text into Etherpad, the entire chunk of text that was pasted showed up as a single keystroke, produced at a single instant. Since this text could not be analyzed for its writing process, we discarded any such data.

Furthermore, although keystroke logging tools are valuable in gathering objective language data without interrupting the writing process, they are limited by how quickly they can keep up with a writer. Etherpad is only able to update every 500 ms, meaning sometimes two or three characters typed very quickly were grouped together as a single keystroke even if they were not pasted. Since these keystrokes were still valid in regard to the writing process, we decided to keep the data by dividing the timestamp for the clumped keystrokes evenly among the single keystrokes. For example, if two keystrokes were recorded at the same timestamp lasting 500ms, we updated the data so that each keystroke was labeled with time 250ms. We believe these methods offer the most accurate depiction of the writing process provided by Etherpad.

2.2 Extraction of Keystroke Measures

Using the data stored in the keystroke data frame, we were able to build data frames for words, sentences, files, and users. For example, by adding up the time taken for all characters belonging to a single word, we found the time taken for each word. Then, we could do the same by adding all the word times belonging to a single sentence, then all the sentence times belonging to a single file. We found that it was most important to develop keystroke measures pertaining to the sentence level because our complexity analysis tool also evaluated each file by sentence.

Other keystroke logging research suggests evaluating text based on pauses and bursts, however, we developed several specific measures to evaluate the writing process based on the production of each sentence. For instance, instead of classifying all time spent not typing as a “pause”, we computed the amount of time per sentence spent typing words, time per sentence spent on a space or new line character, and time per sentence spent revising. Based on keystroke position, we were also able to determine how many times a writer jumped between locations within a sentence, as well as how many times a writer returned to a sentence from other locations in the text. These measures were not limited to simple time-related measures like bursts or pauses, so they were useful in examining the detailed writing process of a text. (See Table 2.5.1 for definitions of the specific keystroke measures used in this study.)

One challenge we noted in our data collection was extensive pause length. Since there was no time limit on how long the students could write their learning journals, it was possible for them to start writing, leave the assignment, then return much later to resume writing. To account for this, we adjusted our time measures so that there was a ceiling of 10 seconds on each keystroke. In other words, any keystroke recorded with a time of more than 10 seconds was reduced to 10 seconds in our keystroke data frame. We believe that a pause beyond 10 seconds is not reflective of the underlying cognitive process of writing, and is rather indicative of distraction from the writing assignment.

2.3 Applications of Keystroke Logging in Writing Research

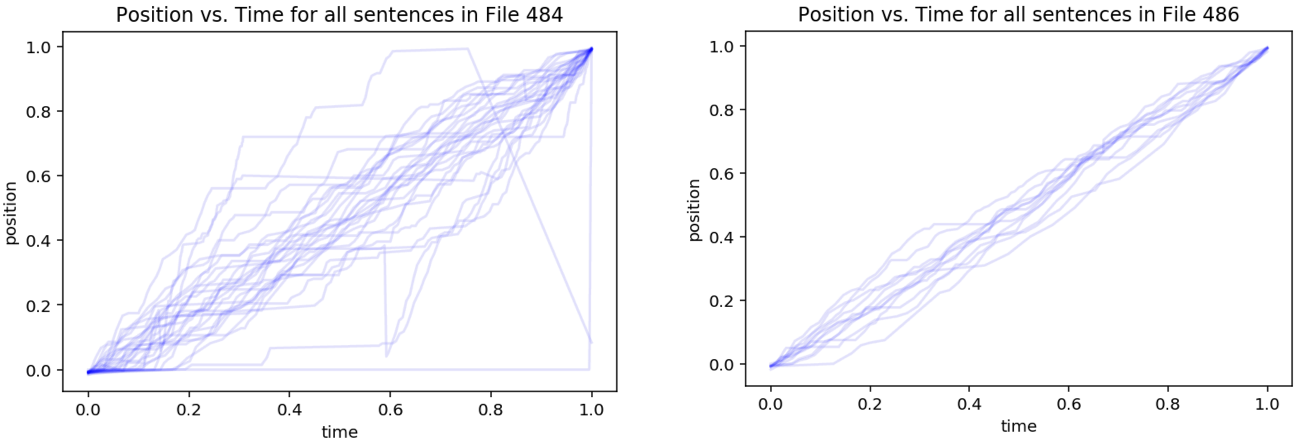


Figure 2.3.1. *A strength of keystroke logging is that it provides insight into the steps that went into producing a text. Both File 484 and File 486 were written by the same user, but it is clear that there was more variation and revision in the writing process for File 484. Data like this is useful to examine how a writer's writing process changes over time.*

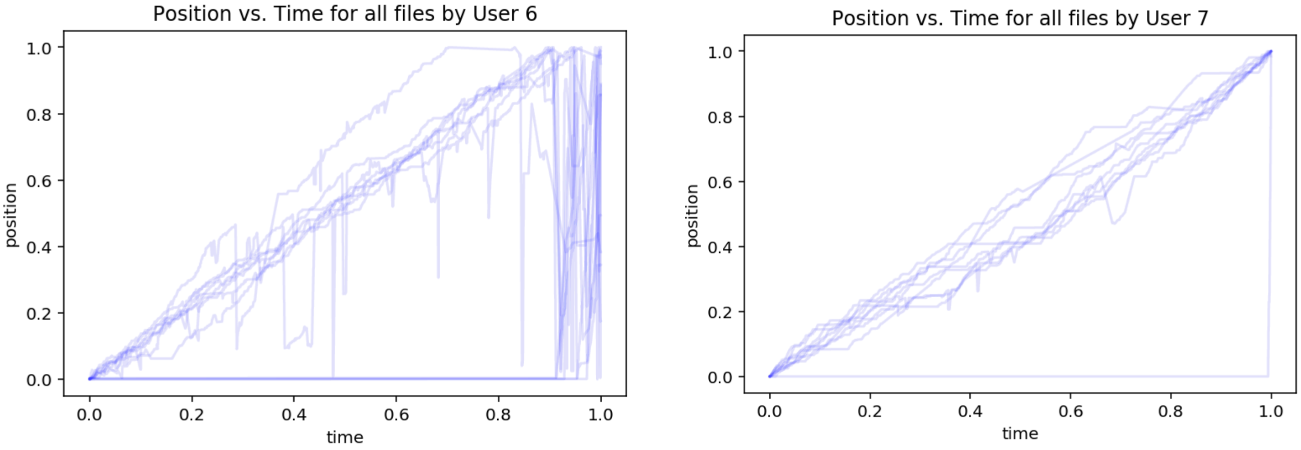


Figure 2.3.2. *Keystroke logging can also help us to identify writing strategies. For example, User 6 tends to write their files straight through, with a few revisions, then revises thoroughly once the text is complete. Comparatively, User 7 consistently writes their files straight through, with minimal revisions. Data like this can help us to determine whether certain writing strategies are more effective than others.*

2.4 Extraction of Complexity Measures

The next step was to use the Complexity Contour Generator (CoCoGen) to evaluate each of the final texts based on text complexity. CoCoGen is a linguistic analysis tool created by researchers at RWTH Aachen University and the University of Amsterdam to measure text complexity using a unique "sliding window technique" [6]. This technique allows the user to change the number of sentences in a text that are evaluated together. CoCoGen also produces complexity values for various different linguistic characteristics pertaining to lexical, syntactic, and morphological complexity. In this way, CoCoGen provides an overview of how complexity changes throughout a text in regard to several different linguistic traits, instead of simply offering a general complexity score for an entire text. This was particularly useful for our study because it allowed us to narrow our focus to the sentence level, rather than relying on average complexity measures for a text as a whole.

With the sliding window set to one sentence, we ran CoCoGen on each separated text file and stored all resulting complexity measures in the sentence data frame.

2.5 Connecting Keystroke and Complexity Measures

With access to keystroke measures pertaining to the writing process of each sentence of a text, as well as CoCoGen measures pertaining to the complexity of each sentence in the end product of the text, it was possible to pair up the measures to search for a correlation between the two categories.

Among the keystroke measures we developed, seven were chosen to compare against the CoCoGen measures: 'word_count', 't_filter_10000', 'jumps', 'chunks', 'word_t/t_filter_10000', 'separator_t/t_filter_10000', and 'revision_t/t_filter_10000' (defined below in Table 2.5.1).

Keystroke Measure	Definition
word_count	Number of words in sentence
t_filter_10000	Total time spent composing the sentence, with all keystrokes over 10 seconds set to 10000ms
jumps	Number of times the writer jumped to non-consecutive positions within the sentence
chunks	Number of times the writer returned to the sentence from a different sentence in the file
word_t/ t_filter_10000	Percentage of total filtered sentence time that was spent within a word (i.e. sum of time spent on all *letter* keystrokes)
separator_t/ t_filter_10000	Percentage of total filtered sentence time that was spent on a space or new line character
revision_t/ t_filter_10000	Percentage of total filtered sentence time spent on a sentence excluding the first draft (i.e. excluding the time spent typing the first character at each position)

Table 2.5.1. *Definition of keystroke measures used in comparison with complexity measures.*

These measures were compared with the CoCoGen complexity measures using the Pearson correlation coefficient, which offers a score representing how correlated two datasets are. A score approaching 1 represents a strong positive correlation, a score approaching -1 represents a strong negative correlation, and a score approaching 0 represents no correlation.

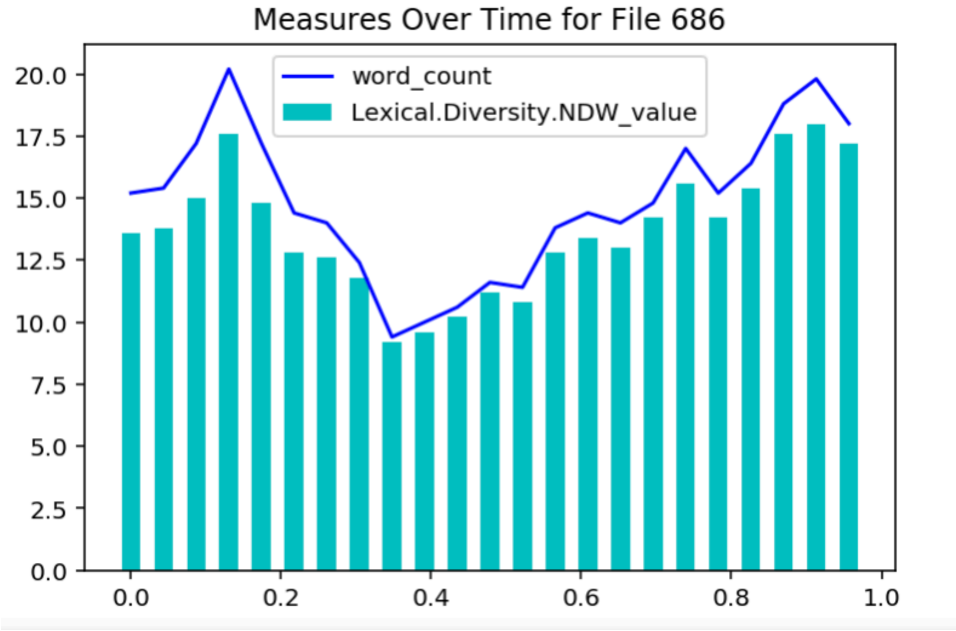
3. Results

3.1 Data Tables

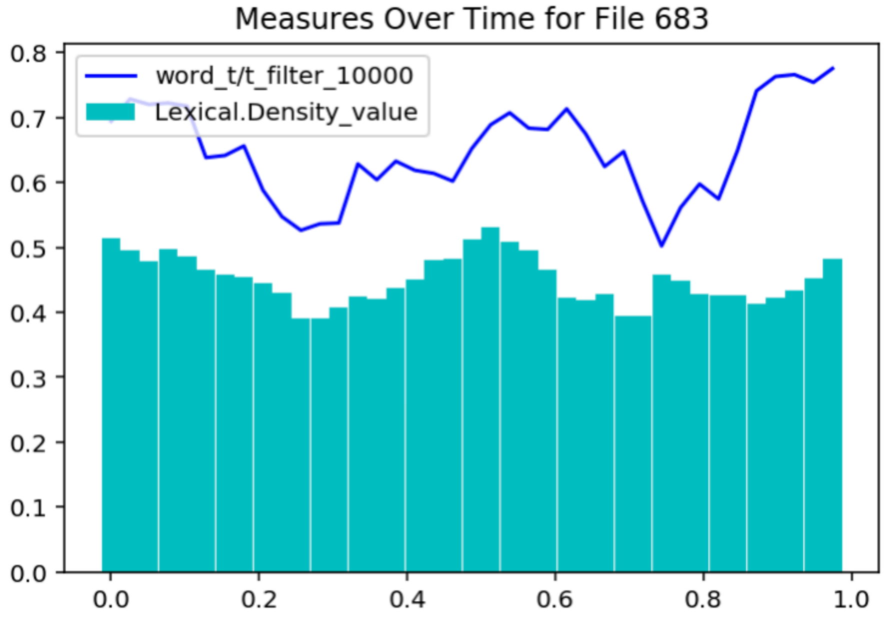
	word_count	t_filter_10000	jumps	chunks	word_t/t_filter_10000	separator_t/t_filter_10000	revision_t/t_filter_10000
KolmogorovDeflate_value	0.88	0.67	0.29	0.17	0.05	0.12	0.04
Lexical.Density_value	-0.09	0.06	0.02	0.01	0.10	-0.12	-0.02
Lexical.Diversity.CNDW_value	-0.59	-0.39	-0.18	-0.10	0.03	-0.14	-0.04
Lexical.Diversity.CTTR_value	0.67	0.52	0.22	0.14	0.02	0.12	0.03
Lexical.Diversity.NDW_value	0.95	0.70	0.29	0.18	-0.00	0.17	0.05
Lexical.Diversity.RTTR_value	0.72	0.56	0.23	0.14	0.01	0.14	0.04
Lexical.Diversity.TTR_value	-0.59	-0.38	-0.17	-0.10	0.03	-0.14	-0.04
Lexical.Sophistication.AFL_value	0.14	0.08	0.03	0.02	0.01	0.01	0.00
Lexical.Sophistication.ANC_value	-0.11	0.06	0.04	0.02	0.03	-0.03	-0.02
Lexical.Sophistication.BNC_value	-0.19	-0.01	0.01	0.00	0.06	-0.09	-0.03
Lexical.Sophistication.NAWL_value	-0.01	0.06	0.02	0.01	0.04	-0.05	-0.00
Lexical.Sophistication.NGSL_value	-0.09	0.04	0.03	0.02	-0.01	0.02	-0.03
Morphological.KolmogorovDeflate_value	-0.84	-0.65	-0.27	-0.16	-0.07	-0.11	-0.04
Morphological.MeanLengthWord_value	-0.13	0.10	0.04	0.02	0.20	-0.23	-0.02
Morphological.MeanSyllablesPerWord_value	-0.11	0.10	0.04	0.03	0.16	-0.20	-0.02
NounPhrasePostModificationWords_value	0.36	0.32	0.16	0.11	-0.04	0.11	0.02
NounPhrasePreModificationWords_value	0.03	0.05	0.03	0.01	0.01	0.01	-0.01
Syntactic.ClausesPerSentence_value	0.59	0.39	0.17	0.10	-0.02	0.12	0.05
Syntactic.ComplexNominalsPerSentence_value	0.70	0.55	0.24	0.15	0.01	0.11	0.03
Syntactic.CoordinatePhrasesPerSentence_value	0.35	0.30	0.14	0.07	0.02	0.04	0.01
Syntactic.DependentClausesPerSentence_value	0.51	0.34	0.14	0.09	0.00	0.09	0.04
Syntactic.KolmogorovDeflate_value	-0.76	-0.59	-0.25	-0.15	-0.06	-0.10	-0.03
Syntactic.VerbPhrasesPerSentence_value	0.60	0.40	0.16	0.10	0.01	0.10	0.05

Table 3.1.1. *CoCoGen complexities (rows) were compared with seven keystroke complexities (columns) to evaluate the correlation between each pair. Using Pearson correlation coefficients, a value of 1 signifies strong positive correlation, -1 signifies strong negative correlation, and 0 signifies no correlation. Green boxes represent coefficients between 0.1 and 0.7 (somewhat positive correlation). Yellow boxes represent coefficients with an absolute value greater than 0.85 (strong correlation).*

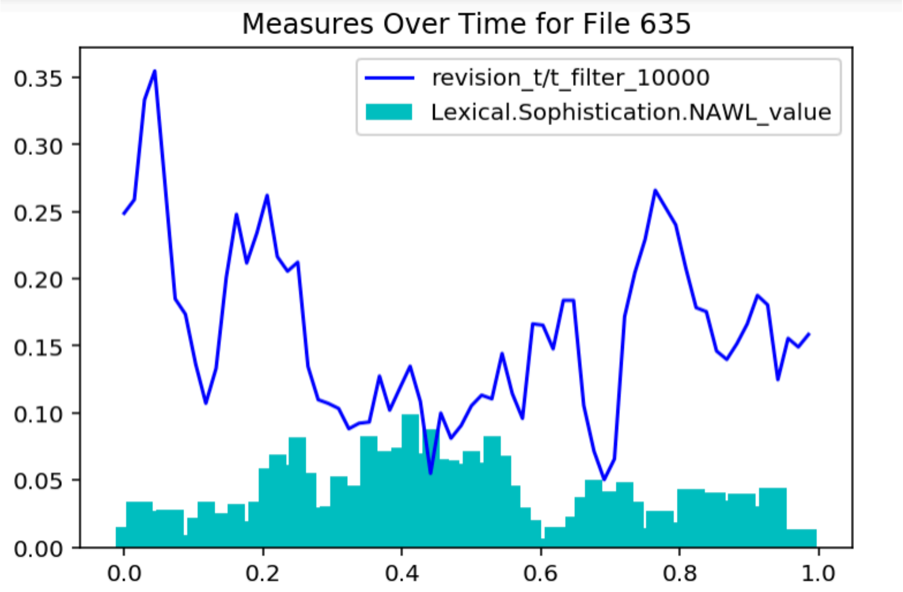
3.2 Graphs



Graph 3.2.1. *Word_count and Lexical.Diversity.NDW_value over the course of File 686. These measures have a strong positive correlation (0.95) over all files.*



Graph 3.2.2 *Word_t/t_filter_10000 and Lexical.Density_value over the course of File 683. These measures have a weak positive correlation (0.1) over all files.*



Graph 3.2.3. *Revision_t/t_filter_10000 and Lexical.Sophistication.NAWL_value over the course of File 635. These measures have no correlation (0.0) over all files.*

3.3 Discussion

The computation of Pearson correlation coefficients revealed a wide array of correlation between the keystroke and complexity measures. Some of the very high correlation coefficients were expected. For example, the value of 0.95 correlation between word_count and Lexical.Diversity.NDW_value is intuitive; a text with a higher word count is bound to have more diversity in its vocabulary. Similarly, many of the comparisons revealed little to no correlation (between -0.1 and 0.1). The correlation values of interest for this study are the somewhat positive correlations (between 0.1 and 0.7), which are highlighted in green in Table 3.1.1.

The results of particular interest pertain to lexical diversity and syntactic complexity. The table reveals that t_filter_10000, jumps, chunks, and separator_t/t_filter_10000 have somewhat positive correlation with Lexical Diversity. It is important to note that Lexical Diversity is split into several different categories. TTR refers to Type-Token Ratio (the number of unique words / the number of total words). CDNW refers to Children's Number of Distinct Words. Both CTTR and RTTR are "corrected" versions of TTR, and NDW (Number of Distinct Words) is a more general and relevant version of CDNW, so we will focus on CTTR, RTTR, and NDW, all of which have positive correlation coefficients with the keystroke measures mentioned above.

First, perhaps it seems intuitive that sentences that take more time would consist of a more diverse lexicon. However, the positive correlation between lexical diversity and separator_t/t_filter_10000 suggests that a more diverse lexicon relies on more time spent *between* words. This implies that more lexical diversity demands more planning time.

Second, there is a positive correlation between lexical diversity and both jumps and chunks. This suggests that a writer's vocabulary may be more limited when composing a first draft, but diversifying the lexicon of a text generally requires some revision, both while initially writing the sentence, and after the first draft of the sentence has been written.

In regard to syntactic complexity, again, word count and time seem like fairly obvious connections: longer sentences are likely to have more clauses. However, jumps have a positive correlation with all syntactic measures regarding the amount of phrases or clauses in a sentence. This connection reveals that the presence of more jumps within the same sentence acts as an indicator of more clauses in the sentence. Furthermore, the fact that jumps are correlated with syntactic complexity but chunks are not suggests that the composition of more complex sentences involves more revision at the initial time of writing the sentence rather than after the fact.

Finally, it is interesting to note that revision_t/t_filter_10000 has no correlation with any of the complexity measures. For this reason, we suggest that jumps and chunks are more useful indicators of revision in the writing process than the revision time measure we developed.

4. Evaluation

4.1 Alterations / What I Learned

To improve this research project in the future, a few changes could be made. First, the subjects could be supervised and timed during the writing process to ensure there is no copy/pasted text and to discourage long pauses. While this method of data collection was not feasible for our purposes, it may provide data with less of a need for preprocessing/cleaning. Additionally, a potential extension of the experiment would be to use the data we have gathered to train a machine learning algorithm to predict text complexity based on certain writing characteristics. This possible study would be applicable in identifying which writing strategies produce the most complex text.

Through the course of this project, I have improved both my technical and interpersonal skills. This is by far the largest dataset I have ever handled, and it required me to learn how to use the Stanford CoreNLP software, how to interpret changeset data, how to maneuver the Pandas library in Python, and how to create and analyze graphical data representing general trends in the dataset. Additionally, I bettered my ability to work in a group, specifically learning how to contribute to a group of students with different backgrounds and skill levels.

4.2 Conclusion

The data reveals that it is possible to connect characteristics of the writing process to lexical and syntactic complexity of the end-product text. Therefore, keystroke logging does offer a method to discover which writing strategies and cognitive processes produce the most complex text. The results suggest that in-sentence revisions (jumps), post-writing revisions (chunks), and longer pauses between words and sentences (separator_t/t_filter_10000) are all indicators of more complex text. Based on these results, more research should be done regarding the cognitive processes that are related specifically to revising and planning while writing, as well as how they relate to text complexity.

5. References

[1] Alves, Rui Alexandre, et al. “Execution and Pauses in Writing Narratives: Processing Time, Cognitive Effort and Typing Skill.” *International Journal of Psychology*, vol. 43, no. 6, 2008, pp. 969–979.

[2] Spelman Miller, Kristyan, et al. “The Psycholinguistic Dimension of Second Language Writing: Opportunities for Research and Pedagogy Using Computer Keystroke Logging.” *Tesol Quarterly*, vol. 42, no. 3, 2008, pp. 433-454.

[3] O’Brien, Sharon. “Pauses as Indicators of Cognitive Effort in Post-Editing Machine Translation Output.” *Across Languages and Cultures*, vol. 7, no. 1, 2006, pp. 1–21.

[4] Baaijen, Veerle M., et al. “Keystroke Analysis.” *Written Communication*, vol. 29, no. 3, 2012, pp. 246–277.

[5] Leijten, Mariëlle, and Luuk Van Waes. “Keystroke Logging in Writing Research.” *Written Communication*, vol. 30, no. 3, 2013, pp. 358–392.

[6] Ströbel, Marcus, et al. “CoCoGen - Complexity Contour Generator: Automatic Assessment of Linguistic Complexity Using a Sliding-Window Technique.” *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity*, 2016, pp. 23–31.