For some years I have been disappointed by the lack of a complete, digitized version of Emily Dickinson’s poems. Although some fraction of her poems are available, for instance on Project Gutenberg[[1]](#footnote-1) or Wikipedia[[2]](#footnote-2), these sources are explicitly not authoritative, include spurious titles, and are not aligned to either the Johnson, Franklin, or fascicle/set numbering.

There are numerous reasons why a digital variorum edition of Dickinson’s poems is highly desirable. In the first place, such a file could be used to discover concordances between poems and periods that are difficult for the human eye to perceive; these relationships, in turn, could be used for more accurate dating and placement of the poems, and would of course support the interpretation of the works themselves. Given Dickinson’s popularity among the public, I was surprised to learn that there is as yet no digital edition of either the Franklin or Johnson assemblages. At the same time, databases clearly exist: both the Emily Dickinson Lexicon and the Emily Dickinson Archive yield database records (texts, metadata, and/or images) that are clearly derived from Franklin or Johnson. Unfortunately, contact with the administrator of the EDL and Harvard University Press yielded no leads. Thus, I was forced to create my own.

## Digitization

I began with the 1955 Johnson edition for three reasons:

1. The reading edition had the fewest marginal notes of all editions currently in print, a factor which would simplify the digitization process.
2. The paperback edition was both low-cost and perfect-bound, desirable traits as my digitization process required defacement of the text.
3. I had first been exposed to Dickinson’s poems through this edition, and I supposed that this familiarity would aid me in cleaning and constructing the dataset.

What follows is a description of the digitization workflow. Improvements could certainly be made. I tended to value economy of time over preservation of resources

1. A small handsaw was used to cut the spine and glued pages away from the book.
2. A stopped paper cutter was used to align the rough-sawn edges of the pages.
3. The pages were digitized using the top-feed feature of a <model> scanner, about 200 pages (100 sheets) at a time at 400 DPI. Higher resolution would likely aid in the OCR process described below.

## OCR

OCR was first attempted using Adobe Acrobat; however, the results were unsatisfactory, so recourse was made instead to the ABBYY FineReader software. Specific steps for this portion of work were as follows (all commands refer to FineReader):

1. Open the scanned PDF(s).
2. Enter “Edit Image” mode.
   1. “Crop” margins, headers, and footers from all pages such that only the poem text and poem numbers remain.
   2. Apply “Recommended Preprocessing” to all pages.
3. Leave “Edit Image” mode.
   1. Draw a single “Text Area” around the whole of page 1.
   2. “Save Area Template…” with any name
   3. Select all pages
   4. “Load Area Template…” (check “Apply to All Pages”)
4. Click “Read”
5. Save as plaintext with line breaks.

## Corrections

While FineReader produced a substantially more accurate output than Acrobat, there were still numerous transcription, OCR, and artifact errors in the text. While many of these errors were resolved by hand, a number of regular expressions streamlined the process substantially and provide a basis for a more programmatic approach to future digitization:

find \r\n([0-9])

repl \r\n\t\t\t\t\t\t\t\t\t\t\t\t$1

find °

repl 0

find -

repl -

find -

repl -

find -

repl -

find -

repl -

find \r\n(\d\d\d)\r\n

repl \r\n0\1\r\n

find \r\n(\d\d)\r\n

repl \r\n00\1\r\n

find \r\n[a-z]

repl BY HAND

find \r\n[^'"a-zA-Z\d\s:]

repl BY HAND

find \t[\w]

repl

find “

repl "

find ”

repl "

find ’

repl '

find 1

repl I

find ~

repl -

find \|

repl

find [[DOT]]

repl

find ■

repl

find [^aI\W]\r\n

repl

find ;

repl

find s

repl 's

find \\*

repl BY HAND

find \^

repl BY HAND

find ([a-z])([A-Z])

repl \1\r\n\2

Note that these regexes were written for Notepad++ and are not guaranteed to work for, e.g., Python or Perl implementations.

The cleaned OCR output included Johnson numbers and the text of the poem (with line breaks intact though often erroneous; see below). Several other data fields were added via a regex (\r\n([0-9]+)\r\n, EOT\r\n\1\r\n0000\r\n0\r\n0\r\nA\r\n0000\r\n0000\r\n0000\r\n0000\r\nSOT):

0 Johnson Number

1 Franklin Number

2 Fascicle (0=None)

3 Stanza Structure (0x0, 0=No Stanzas)

4 Rhyme Scheme (TBD programmatically)

5 Johnson Year Written

6 Johnson Year Published

7 Franklin Year Written

8 Franklin Year Published

9 Johnson Editorial Note

10 Franklin Editorial Note

11 Title

12 \SOT

13: \POEM TEXT

\EOT

Where “SOT” means “Start of Text” and “EOT” means “End of Text.” This resulted in a final plaintext file consisting of records, each of which was formatted as follows:

0000

0000

0

0000

A

0000

0000

0000

0000

J\_note

F\_note

SOT

POEM TEXT

EOT

These data fields have been changed in the final version and, in retrospect, constituted unnecessary inclusions: it proved simpler instead to construct a separate list of data fields and join them, programmatically, to the plaintext output.

## Assembly

The remainder of the data cleaning and shaping took place via a Python script. The following steps were applied in a process which is repeatable for any new data:

* The plaintext dataset was read into Python as a set of lists of data elements, e.g.
  + johnson\_number = [1, …, 1775]
  + cmp\_yr = [1850, …, 9999]
* Franklin numbers, fascicle/set locations, and year of first publication figures were scraped from Wikipedia and mapped to new variables.
* The expanded dataset was exported to a new plaintext file with the new fields included, and old, redundant fields excluded.

## Stanzas

Line breaks were inconsistently detected in the scanning and digitization process (see Appendix). Therefore, it was necessary to manually review each poem and note down its stanza structure, a task facilitated somewhat by Dickinson’s highly regular style. For the curious, 568 of her poems include no stanzas/a single stanza, 998 of her (multistanza) poems are in quatrains, and the remaining ~200 are in various formats, most frequently sets of triplets and sextets.

## Rhyme and Meter

Dickinson’s novel approach to rhyme is, along with her employment of dashes, one of the most immediately noticeable aspects of her poems. It also poses serious difficulties for a digitization effort[[3]](#footnote-3). Rhymes were identified using the following heuristic:

1. For each poem, map ARPAbet phonemes and standard (non-poetic) stress values to each word in each line via the CMU Pronunciation Dictionary.
   1. ### unidentifiable words were manually tagged.
2. For each poem:
   1. Instantiate a rhyme array filled with the value “0”
   2. Extract the final vowel phonemes from each line
   3. For each phoneme, if the phoneme occurs >=1 time in any other line, assign a letter (“A”) to the rhyme array for that line and all matching lines
   4. that phoneme occurs 0 times in any other line, assign the line *None*.
3. Compare rhyme arrays against a selection of Dickinson’s known, hand-keyed, “standard” rhyme schemes (e.g., “ABCB”)
4. Return a list of poems that do not match a standard scheme

## Meter

In contrast to her rhyme, Dickinson’s meter is highly regular, and furthermore tends to be syllabic, making the attribution of standard forms (e.g., the common “hymn meter”) quite simple, especially once elided syllables are added to the CMU Pronunciation Dictionary.

1. See for example <https://www.gutenberg.org/ebooks/12242>, which collects three series [↑](#footnote-ref-1)
2. <https://en.wikipedia.org/wiki/List_of_Emily_Dickinson_poems> (note that while the poem transcripts are of doubtful accuracy, the large table on the linked page has proven useful for collating numbering between the poems’ various organizational regimes). [↑](#footnote-ref-2)
3. <http://ischoolreview.com/iSR_Grav/entries/entry-6> provides a useful overview of an approach to rhyme detection, one I employ here in a novel way. See also Sravana, R., Knight, K. 2011. Unsupervised Discovery of Rhyme Schemes. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: shortpapers. [↑](#footnote-ref-3)