

Automating the Detection of Poetic Features: The Limerick as Model Organism

Almas Abdibayev

Department of Computer Science
Sudikoff Lab
Dartmouth College
Hanover, NH 03755

almas.abdibayev.gr@dartmouth.edu

Yohei Igarashi

Department of English
University of Connecticut
Storrs, CT 06269-4025

yohei.igarashi@uconn.edu

Allen Riddell

Department of Information & Library Science
Indiana University Bloomington
Bloomington, IN 47408

riddella@indiana.edu

Daniel Rockmore

Department of Computer Science
Dartmouth College
Hanover, NH 03755

daniel.n.rockmore@dartmouth.edu

Abstract

In this paper we take up the problem of “limerick detection” and describe a system to identify five-line poems as limericks or not. This turns out to be a surprisingly difficult challenge with many subtleties. More precisely, we produce an algorithm which focuses on the structural aspects of the limerick – rhyme scheme and rhythm (i.e., stress patterns) – and when tested on a culled data set of 98,454 publicly available limericks, our “limerick filter” accepts 67% as limericks. The primary failure of our filter is on the detection of “non-standard” rhymes, which we highlight as an outstanding challenge in computational poetics. Our accent detection algorithm proves to be very robust. Our main contributions are (1) a novel rhyme detection algorithm that works on English words including rare proper nouns and made-up words (and thus, words not in the widely used CMUDict database); (2) a novel rhythm-identifying heuristic that is robust to language noise at moderate levels and comparable in accuracy to state-of-the-art scansion algorithms. As a third significant contribution (3) we make publicly available a large corpus of limericks that includes tags of “limerick” or “not-limerick” as determined by our identification software, thereby providing a benchmark for the community. The poetic tasks that we have identified as challenges for machines suggest that the limerick is a useful “model organism” for the study of machine capabilities in poetry and more broadly literature and lan-

guage. We include a list of open challenges as well. Generally, we anticipate that this work will provide useful material and benchmarks for future explorations in the field.

1 Introduction

The application of natural language processing methods to the field of poetry and poetics has led to much promising work. In particular, advances in machine learning-based language models have spurred investigations into various kinds of automated poetic analysis and synthesis. There has been recent work pursuing automated scansion (“scansion” denoting the marking of feet, accents, and caesuras or pauses in lines of poetry) (Agirrezabal et al., 2017a; Anttila and Heuser, 2016; de la Rosa et al., 2020; Marco et al., 2021; Greene et al., 2010; Hammond, 2013; McCurdy et al., 2015; Lau et al., 2018) and probing the ability of language models to encode rhyme (Abdibayev et al., 2021). Other work is devoted to the automatic classification of poetic kinds, like the haiku in English (Long and So, 2016) and free verse (Baumann et al., 2018). On the generative or machine writing side of the problem, models have been trained to output synthetic poems that are increasingly difficult to tell apart from human-made poetic compositions; such work includes the generation of sonnets (Ghazvininejad et al., 2017) and, especially relevant to the present paper, limericks (Wang et al., 2021). These computational approaches to poet-

ics share many of the same objects of inquiry as recent scholarship in literary criticism, especially those associated with the “New Formalism” and “Historical Poetics” (Adams et al., 2016; Levinson, 2007; Culler and Glaser, 2019). Such scholarship and computational poetics together prompt us to attend once again to the particularities of poetic form – meter and rhythm, poetic sound generally, aspects of lyric poetry, and related concerns. This kind of joined work promises to advance our understanding of the formal elements of poetry while also making progress in our understanding of the capabilities of language models to gain what might be called poetic knowledge.

In this paper we hope to contribute to computational poetics and formalist literary criticism alike by offering an account of our work in the automatic detection of poetic features, using the limerick form. The limerick is a comedic, light, and primarily oral poetic kind. While there is but a scattering of scholarship on the genre (see e.g., (Baring-Gould, 1988; Belknap, 1981; Bibby, 1978; Legman, 1969, 1980)) there are nevertheless important reasons to take up the limerick. State-of-the-art language models still tend to perform better with short texts (Liu et al., 2019), and the limerick is a short verse form. Even better, limericks have a high density of conspicuous poetic-formal features compared to other short verse forms (Wang et al., 2021). Haiku, for example, do not have rhymes or consistent accentual patterning, and epigrammatic couplets are less interesting than limericks in terms of rhyme scheme and rhythmic variation between lines. For these reasons, the limerick is a particularly promising case study for computational poetics. In biological terms, it is a useful model organism, an object of study tactically chosen for its relative simplicity and potential to provide insights into broader phenomena (Hedges, 2002; Poovey, 2001) – in our case, poetry more broadly.

Specifically, with this paper we offer the following. First, a large, clean limerick data set as a resource to other researchers: Wang et. al, noting the unavailability of large limerick data sets, have offered a novel corpus of their own, made up of 2,000 limericks (Wang et al., 2021) We offer a data set of 98,454 highly regular limericks drawn from the *The Omnificent English Dictionary in Limerick Form* (“OEDILF”) website. Second, we outline our methods for detecting two of the three main features of limericks: their rhyme scheme and ac-

centual pattern. (We discuss the third feature of the limerick, its triplet meter, which is closely related to its accentual pattern, in our “Future Directions” section.)

The “limerick decision” problem, of deciding whether or not a poem is a limerick through the automatic discernment of formal features, may appear to be of only narrow interest. But, as per the model organism conceit we see it as not only enabling the construction of a well-defined poetic, even literary laboratory, but also as a direct pathway to a range of broader potential applications for computational poetics and formal feature detection. These include the automated discernment of latent poetic features of songs, advertising and marketing copy, political speeches, and more. We take this up in the concluding “Future Directions” section. The case or model organism of the limerick has implications that reach well beyond the form itself.

2 Recognizing The Limerick

According to the *New Princeton Encyclopedia of Poetry and Poetics* the limerick is “the most popular form of comic, light verse in English.” The origins of the form and its very name “limerick” are obscure, but the form was popularized by Edward Lear’s *A Book of Nonsense* (1846) and became widespread by the early 20th century (Preminger et al., 1993).

The limerick form has three main formal features:

Rhyme scheme. The limerick’s rhyme scheme is *AABBA*. In written form, limericks are almost always arranged as five-line poems, but it is also possible to view the limerick as a four-line poem by combining the third and fourth lines as a pair of *hemistichs* (half-lines of verse) that are separated by a *caesura* (a pause). It is even possible to understand the limerick as comprised of two long verse lines, the first (merging lines 1 and 2) having six accents and the second (merging lines 3-5) having seven accents. (Preminger et al., 1993; Attridge, 1982). Whatever its written form, all recited limericks conform aurally to the *AABBA* rhyme scheme.

Accents or Stresses. The number of stresses in each line is organized as follows: line 1 (3 stresses), line 2 (3 stresses), line 3 (2 stresses), line 4 (2 stresses), and line 5 (3 stresses). The 3-stress lines—lines 1, 2, and 5—typically have seven to ten syllables, while the two shorter lines—lines 3 and 4—typically have five to seven syllables. But, being

an instance of accentual verse like the traditional ballad, the limerick allows for much variation in the total number of syllables per line in realizing the necessary 3-3-2-2-3 stress pattern.

Meter. The main metrical foot of limericks is debatable. Depending on how one hears or recites limericks, the limerick's metrical foot is either anapestic, amphibrachic (Preminger et al., 1993), or (if one treats the initial unstressed syllable as something like a musical pick-up note or *anacrusis*) dactylic. What is unequivocal is that there are trisyllabic feet (or, triplets, musically speaking) driving the rhythm. Another interesting aspect of limerick rhythm is known to all reciters and auditors of limericks: the pauses at the end lines, which themselves can be sorted into the typically lengthier pauses at the end of lines 1, 2, and 5 and the shorter pauses at the end of lines 3 and 4. These lines can be classified as "catalectic" (an incomplete line of verse). But isochronic approaches to meter, though out of fashion today, have historically incorporated musical notation, and they are therefore best at revealing these important rests which are not registered by scansion (Attridge, 1982; Lanier, 1979).

The content of limericks is usually bawdy, which can raise problems in data curation. Nevertheless, it is a poetic form whose relative brevity, rhyme scheme, distinct accentual pattern, and meter make it not only an ideal form for introducing poetry to elementary school students, but a valuable testbed for computational poetics. We can see these primary formal elements at work in one of Lear's most well-known limericks, "There was an Old Man with a Beard" (Foundation, 2021):

There was an Old Man with a beard,
Who said, "It is just as I feared! –
Two Owls and a Hen,
four Larks and a Wren,
Have all built their nests in my beard!"

Note the adherence to the *AABBA* rhyme scheme – "beard," "feared," "Hen," "Wren," "beard" – and how the final line's end rhyme circles back to the *A* rhyme, giving a sense of circularity and closure. We can also hear the 3-3-2-2-3 accentual pattern: in line 1, "there **was** an Old **Man** with a **beard**," and in line 3, "Two **Owls** and a **Hen**." The dominance of trisyllabic feet (dactylic, anapestic, or amphibrachic) is audible too: if predominantly dactylic, the initial syllable of each line is unstressed, then followed by a series of dactyls ("**all** built their **nests** in my . . .").

Herein we focus on the detection of limerick rhyme scheme and accentual pattern. While these appeared to be the most tractable formal elements (compared to the ambiguities of the metrical system), even rhyme and accent detection turn out to be nontrivial problems.

3 Data

We scraped from the website *The Omnificent English Dictionary in Limerick Form* ("OEDILF"). The OEDILF, established in 2004, is an amateur, crowd-sourced project whose goal is to have at least one limerick for every meaning of every word found in the *Oxford English Dictionary*. User-submitted limericks are subject to approval by moderators, and, if approved, are published on the website.¹ We turned to OEDILF in order to get a large number of limericks which we could then sort according to different categories of metadata.² At the time of writing this paper, OEDILF contained 112,609 submitted limericks, of which 110,610 were approved and posted to the site.

On the website, the limericks are organized according to different categories:

Authors. Our scrape of OEDILF contains limericks by 1624 unique authors. The distribution of limericks by author is not uniform (Figure 1), with the top ten authors responsible for ca. 40% of all limericks.

Topics. The topics are generated by the contributors to the website as well as by the moderators.

Words. OEDILF is interested in words as found in the dictionary, as described above. The website allows users to search limericks by keywords: for instance, there are 52 limericks featuring the word, "bridge," spanning its denotations as card game, infrastructure, a platform found on vessels, segment of a pop song, and so on. Currently, OEDILF is focusing on words beginning with the letters "Aa-" through "Ha-," and indicates the projected date of completion (presumably having a limerick for every meaning of every word found in the OED) as November 3, 2063.³

For our detection experiments below, we first filtered limericks based on simple criteria: limericks must have 5 lines and must use words (as opposed to symbols, such as emojis or formulae). This excludes a range of unorthodox limericks such as

¹<http://http://www.oedilf.com/db/Lim.php?View=About>

²<http://www.oedilf.com/db/Lim.php>

³<http://http://www.oedilf.com/db/Lim.php?View=About>

ASCII art limericks. The result was a corpus of 98,454 limericks, dated 1998 to the present.

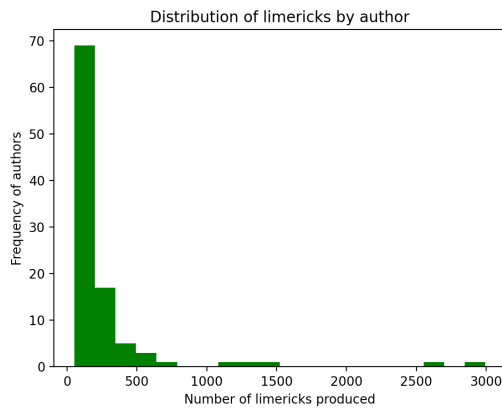


Figure 1: Histogram plotting distribution of top 100 authors within the entire scraped dataset. The y-axis shows the number of authors within a bin, while the x-axis shows the number of limericks those authors produced.

4 Method

4.1 Rhyme Detection

Detecting the correct rhyme scheme (*AABBA*) was the first step we took in order to sort out limericks that did not adhere to the rhyme scheme. We employed two models trained on CMUDict⁴: one to predict phoneme structure and another to predict stress patterns across these phones. Using this method, we determine that two words rhyme if after breaking them down into phonemes with stressed vowels, the last stressed vowels in the syllables match and any following phonemes also match. For example, words *fire* and *rewire* are broken down into F AY1 ER0 and R IY0 W AY1 ER0. The spaces separate phones and numbers indicate stress pattern, 0 for no stress, 2 for secondary stress, and 1 for primary stress. In this case the algorithm would walk through the phonemes until encountering AY1 in both phoneme sequences and then compare AY1 ER0 with AY1 ER0, concluding that the words are in fact rhyming. In some cases words may have several pronunciations which results in two different phone sequences. Here, we assume that if one of the phoneme sequences matches with the phoneme sequence of another end-of-the-line

⁴<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>. For a different method based on eSpeak, see Alnajjar and Hämäläinen 2018

word in the limerick, then that is the intended pronunciation.

We choose a two-model approach since joint models of phoneme and stress tend show lower accuracy for the stress prediction task. The first model, the phoneme predictor, is an open source Transformer (Vaswani et al., 2017) model (out of the box implementation available in the Tensor2Tensor toolkit (Vaswani et al., 2018)) that was trained on CMUDict⁵. Transformer is a neural network that consists of several layers of attention (Bahdanau et al., 2014) interspersed with fully connected layers. The variation of the model trained to predict phonemes was released by researchers at Carnegie Mellon University, under the umbrella name CMU Sphinx. It features 3 layers with an embedding size of 256. The second model is a custom phoneme LSTM (Hochreiter and Schmidhuber, 1997) that we also trained on CMUDict to predict the stress pattern for a sequence of phonemes produced by the previous grapheme to phoneme model. In particular, our model is a simple bidirectional LSTM (biLSTM) with 3 layers and hidden size of 30 units, followed by a linear projection layer that outputs probabilities for all three possible stress patterns. We do not use dropout or other regularization techniques (such as variational dropout) as we found they lower performance on the test set. We trained this network for 10 epochs using learning rate of 10^{-4} using AdamW on a test set of 107,498 examples, validation set of 13,437 examples, and test set of 13,438 examples, until validation loss plateaued. We obtain 84% accuracy on our test set. This model has a total of 60,393 parameters. We have experimented with other more complex architectures (such as a vanilla GPT model of varying number of layers and embedding sizes) and hyperparameter setups, but found that biLSTM performs the best (our best Transformer model achieved 70% accuracy on the task), perhaps due to the size of the data set, while also featuring 2 orders of magnitude fewer parameters.

As a result of rhyme scheme filtering using predicted stressed phonemes, we were left with 65,971 limericks out of 98,454 of the original data set (a 70% yield). The associated labeled data set (“limerick” or “non-limerick”) can be found at <http://dx.doi.org/10.5281/zenodo.5520078>.

A random sample of limericks that did not pass

⁵<https://github.com/cmuspinx/g2p-seq2seq>

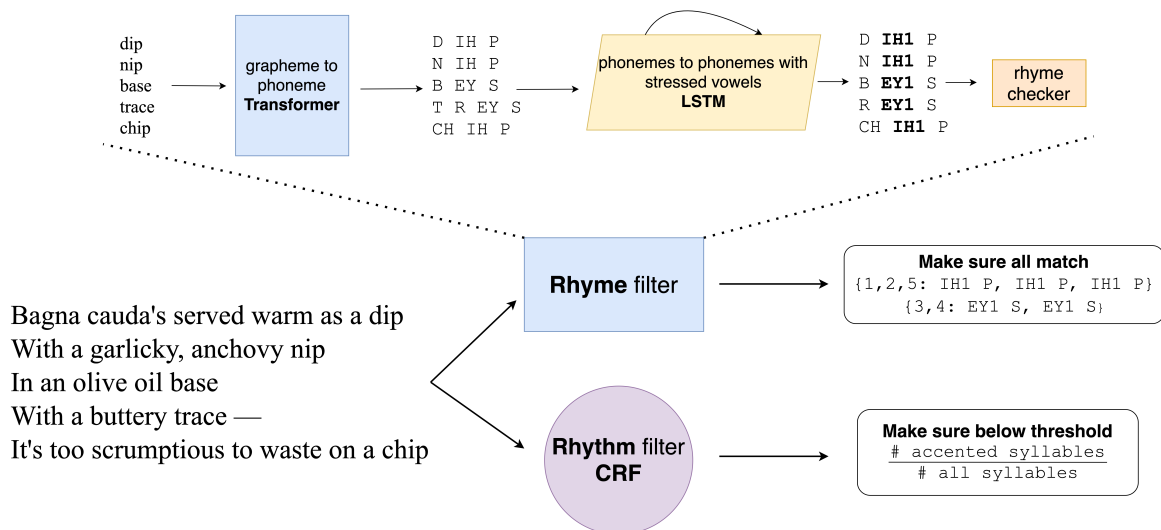


Figure 2: Overview of our filter.

the rhyming test (30% of all limericks) yielded fifteen examples where the non-rhyme could be attributed to non-standard or difficult word spellings (and thus an associated difficulty in producing the phones). There were four examples of “near” or “imperfect” rhymes (e.g., “groaned” and “crooned”) and one example of an “eye” or “sight” rhyme (e.g., “prove” and “love”) (Abrams, 1999). In Table 1, there are examples of these, while in the Appendix we include five other examples of “failures.” The detection of near and eye rhymes is an outstanding challenge for computational poetics.

4.2 Rhythm (Accentual Pattern) Detection

To characterize the rhythm of a poem, we “scan” a poem (that is, perform a scansion). Numerous studies have documented that it is possible to use familiar models from computational linguistics to scan poetry (Haider, 2021; Agirrezabal et al., 2017b; Estes and Hench, 2016). These models are trained on poems that have been scanned by human annotators. We use a conditional random field (CRF) model (Lafferty et al., 2001; Okazaki, 2007), trained on the *For Better for Verse* corpus (Bet, Accessed July 1, 2021). We extract a range of simple features from each word (e.g., character trigrams, syllable count) and predict each word’s “label”—the word-specific scansion. For example, in a particular poem the word *evening* might be scanned as +−, where “+” indicates a stressed syllable and “−” an unstressed one. The CRF, sometimes called a discriminative random field, jointly models

the input features and the labels. The CRF model has the virtue of being more than a decade old and being widely used in computational linguistics for named entity recognition, among other applications (Murphy, 2012, Sec. 19.6).

A simple approach to identifying five-line poems that feature the limerick rhythm would be to use a model which infers, for each line, a probability distribution over the number of accented syllables. If the number of accents conspicuously departs from the (3, 3, 2, 2, 3) pattern, the poem would be rejected as a non-limerick. Unfortunately, such an approach is not available at present because the best models that perform scansion frequently make mistakes when considering whole lines (Haider, 2021). No method exists which has line-level accuracy above 90% for English poetry. Moreover, the limerick form poses special difficulties because its pattern of stresses and unstresses oftentimes depends on a departure from word or lexical stress: for example, ordinarily, in the line, “There was an Old Man with a beard,” “Old Man” would be two consecutive stresses (a spondee), but the three beats of this line dictate that “Old” is demoted a to non-stressed syllable in recitation, while “was” becomes stressed. Limericks and their comedic effects depend on such “wrenched accents” (Abrams, 1999), which mark the difference between lexical stress and the recited poetic rhythm, but this makes word-based scansion impractical.

We therefore choose a simpler approach and instead attempt to identify poems which have, in

Near Rhyme	Eye Rhyme
This sign, where the road intersects With the train track, is shaped like an X. It's a crossbuck, which works As a warning to jerks Who might otherwise end up in wrecks.	In my audio-lingual exam, I recited from Omar Khayyam. I am fortunate, still, For this Arabic drill: Jug of wine, loaf of bread, lots of jam.

Table 1: Two limericks whose rhyming defeated our filter. The rhyming of “X” with “intersects” and “wrecks” is a near rhyme and the rhyming of “Khayyam” rhyming with “exam” and “jam” is an eye rhyme. Both limericks by Chris Doyle, available at OEDILF as limericks #60115 and #9508 (accessed July 1, 2021).

aggregate, more accented syllables than would be expected given the (3, 3, 2, 2, 3) pattern found in conventional limericks. We use the ratio of accented syllables to total syllables as our measure of the prevalence of stressed syllables. Because this approach considers the entire poem, it averages over the line-level inaccuracy of the scansion model.

To determine a cutoff beyond which we say a poem contains too many accented syllables, we look at the ratios associated with a small number of published conventional limericks. The mean and standard deviation of these ratios are 0.47 and 0.038 respectively. Assuming that the ratio follows a normal distribution in conventional limericks, we set our cutoff to be 2.33 standard deviations higher than the mean ($\Phi^{-1}(0.99) \approx 2.33$). So any poem associated with a ratio higher than this value, 0.562, will be rejected as featuring too many accented syllables to be a limerick.⁶

Of the 98,454 poems in the OEDILF corpus, 4,773 (5%) were flagged as having more accented syllables than would be expected in a limerick. To estimate the performance of this method, we randomly sampled 20 flagged limericks as well as 20 limericks which were not flagged. We then manually assessed whether or not our approach succeeded in identifying poems which did not feature the conventional limerick rhythm. We estimate that this method has an accuracy of about 95% and an F1 score of about 0.096. Because the vast majority of limericks in the OEDILF corpus use the expected rhythm, estimating the true negative rate of this classifier is a challenge. (A corpus of almost-correct limericks would be helpful here.) The only poem correctly flagged as flawed was the

⁶Note that these values are all calculated using our particular CRF model. The particular values (mean, standard deviation, cutoff) should be re-calculated if a different model is used.

following:⁷

Upset tum from a dubious chef?
 Coughing? Lame? Itchy? Sore? Sleepy? Deaf?
 Deaf? My old British National
 Formulary ration 'll
 List the pills for your ills — BNF.

We view this poem as flawed in terms of rhythm because it is extremely difficult to recite the third line of the poem using only two accented syllables.

Clearly there is considerable opportunity for improvement here. A corpus of obviously flawed limericks (in terms of rhythm) would be particularly valuable for future research as such a corpus would allow researchers to accurately estimate the performance of a classifier designed to detect flawed rhythm.

5 Results

After applying all of our rhyming and meter filters we identify a final clean set of 65,971 limericks, which constitutes 67% of our original set of limericks that we scraped (Table 2).

One implication of our results is that in the limerick form, at least as understood by the poetizing users and editors of OEDILF, accentual pattern predominates as a necessary formal trait. 95% of the limericks in the data set satisfied our accent measure, and 70% cleared our rhyme filter. Ideally, a poem is an "organic whole" (Poovey, 2001) with its different formal features and the work's meaning working harmoniously together. Yet our findings suggest that, practically speaking, there is a hierarchy of formal features. All formal features are not emphasized equally by writers and readers of poems: some features matter more than others in the quiddity of the form – in the limerick-ness of

⁷By Chris Young, Limerick #83205 in the OEDILF.

the limerick. Another way to put this is that we (humans) have laxer expectations about some formal elements of poems (like syllables), a higher toleration for variation among these elements while still identifying the work as an instance of the poetic kind. (For example, a sonnet with 13 lines may seem less like sonnet than a sonnet whose rhyme scheme is slightly aberrant: in other words, in practice, the number of lines is a more essential feature of the sonnet than strict adherence to the rhyme scheme.)

6 Difficult Examples

The previous subsections take the limerick and break out fundamental structural components that need to be identified for limerick detection. That said, the human – poet or not – can readily create new difficult limericks that might defeat these state-of-the-art poetic classification engines.

We tested our rhyme detection and accent detection algorithms on the following poems⁸ with made-up words for end rhymes. These are words that are definitely not included in CMUDict. Our rhyme and accent detectors both correctly evaluated these examples, accepting all three limericks as featuring legitimate limerick rhyme schemes and stress patterns.

My favorite food is red schnook,
It is best when eaten with glook,
Or with sauce from meef,
Or with sides of dreef,
It always makes me feel so trook.

I like to walk in the dark fraze,
At my side a furry brown braze,
It lopes on its broops,
And shakes his sharp croops,
That stir up the hidden small snays.

It's so tough to fagrewzin a kloom,
Its frantabulous frinks are a croom,
To brillig the shneezles,
Or traming the dresazels,
The sturmping coughs up shnayble and stoom.

7 Future Directions

As mentioned, in this paper we have only addressed the problem of the structural identification of the

⁸All three poems were written by one of the authors of this paper.

limerick. But, of course there is much more to the limerick than just patterns of rhyme and rhythm.

The limerick is typically a narrative. These brief, comic stories fall into different narrative types and structures. For example, consider the paired examples below: the one on the right (“Old Man with a Beard”) exhibits a more linear or teleological structure, while the one on the left (“Old Person of Dean”)⁹ is circular in that it ends with a return to the first line, thereby resembling the refrains found in ballads and villanelles:

There was an Old Man with a beard,
Who said, 'It is just as I feared!
Two Owls and a Hen,
Four Larks and a Wren,
Have all built their nests in my beard!'

There was an Old Person of Dean,
Who dined on one pea and one bean;
For he said, “More than that
would make me too fat,”
That cautious Old Person of Dean.

Thus the narrative aspects of limericks warrant further study, including their narrative structure, punchlines and jokes (which can range from the subtle to the vulgar), twists in the story, and more. A related feature of limericks is their semantic integrity: despite the limerick form’s association with nonsense verse because of Edward Lear’s *A Book of Nonsense* and their oftentimes silly content, good limericks still make sense, cohere, and have aesthetic qualities that can be appreciated. For example, can a machine discern that the limerick immediately below¹⁰ is more successful than the one below it¹¹ because of the latter’s non-sequitur conclusion? Moreover, can it readily detect the non-sequitur?

There was an Old Man in a tree,
Who was horribly bored by a Bee;
When they said, 'Does it buzz?'
He replied, 'Yes, it does!'
'It's a very regular brute of a Bee!'

⁹by Edward Lear, 1920

¹⁰“There was an Old Man in a tree”, by Edward Lear, in *The Book of Nonsense* (1887)

¹¹Derived from “There was an Old Man with a beard”, by Edward Lear, in *The Book of Nonsense* (1887), by changing the last sentence.

	Filter Statistics	
Original data set	Rhyming	Rhythm
98454	69446 (0.70)	93681 (0.95)
Original data set	Rhyme & Rhythm	
98454	65971 (0.67)	

Table 2: Filter statistics. This is a summary of our results. We analyze our data set using various filters to satisfy two of the conventional criteria of a limerick. In other words, (1) it has to follow the AABBA rhyming scheme, and (2) it has to follow the 3-3-2-2-3 accentual pattern.

There was an Old Man with a beard,
 Who said, "It is just as I feared!
 Two Owls and a Hen,
 four Larks and a Wren,
 and mutton is best boiled, not seared.

A third potential area of investigation is word-play, some of which depends on poetic devices such as assonance, alliteration, and puns and inuendo. The problem of machine recognition and subsequent classification of such literary devices pose interesting challenges for a next generation of machine limerick readers.

The long range goals we mention above – indeed the entire setting of this work – places these kinds of investigations and challenges within the context of machine learning and language and thus, broadly in the space of natural language processing (NLP). In that setting, there is a strong thread of work focused on probing the linguistic knowledge of the increasingly complex language models (e.g., Transformer-based deep learning architectures). Of interest are the “minimal pair” challenges in which the architecture is “asked” – after suitable training – to assign a likelihood of correctness (in the sense of a model’s “preference”) between an original and presumably correct instance of some phenomenon and a minimally corrupted (relative to a given grammatical rule) version as evidence of some ability to distinguish a particular linguistic characteristic. In (Warstadt et al., 2020) 12 kinds of phenomena are investigated and each phenomenon is further deconstructed into “paradigms” and paradigm is the source of 1000 instances of minimal pairs.¹² This minimal pairs approach is being extended to poetic devices – specifically, in the investigation of language models to encode rhyming information (Abdibayev et al., 2021). One extension of this work would be the comparison of a “correct” lim-

erick and a variation minimally corrupted in terms of meter or rhyme scheme to test the poetic knowledge in language models and whose results will produce benchmarks (and a fixed test set) for the deep learning community. It is our general sense that poetic knowledge is a challenging domain that productively probes the linguistic capabilities of advanced language models. Moreover, we could well imagine an even broader set of minimal pair benchmarks to interrogate the “writerly” or “stylistic” knowledge of language models.

It is also possible to envision applications for language models that possess poetic knowledge. As Roman Jakobson showed, political slogans (like “I Like Ike”) can have rich aural, orthographic, and rhythmic patterning which augment their catchiness (Jakobson et al., 1987). Such poetic knowledge could extend to the large scale analysis of popular music, advertising and marketing copy, and perhaps even political speeches, propaganda, and disinformation – all with an eye toward examining what makes certain non-poetic instances of language which are nevertheless suffused with poetic features so effective or memorable to large audiences.

At its most speculative, this kind of work trains an odd light on the seemingly impenetrable process of how we learn to write even the simplest forms of poetry like the limerick. A young child learns to rhyme, decide syllable breaks, and can develop a sense of poetic meter very quickly – say, in a single grade school class (if the plethora of teachers’ worksheets available on the internet is any indication). That a language model with billions of parameters and thousands of examples isn’t yet perfect says a lot about both the models we develop and the human process of learning as well.

References

Accessed July 1, 2021. For Better for Verse. <https://prosody.lib.virginia.edu/>.

¹²For example one minimal pair that tests subject-verb agreement (see Table 1 of (Warstadt et al., 2020)) is *These casseroles disgust Kayla. v. These casseroles disgusts Kayla.*

- Almas Abdibayev, Allen Riddell, and Daniel N. Rockmore. 2021. [BPoMP: The benchmark of poetic minimal pairs – limericks, rhyme, and narrative coherence](#). In *Proceedings of Recent Advances in Natural Language Processing*, pages 1–9. Association for Computational Linguistics.
- M. H. Abrams. 1999. *A Glossary of Literary Terms*, 7th ed edition. Harcourt Brace College Publishers.
- V. Joshua Adams, Joel Calahan, and Michael Hansen, editors. 2016. *Special Issue: Historical Poetics*, volume 77 of *Modern Language Quarterly*.
- Manex Agirrezabal, Iñaki Alegria, and Mans Hulden. 2017a. [A comparison of feature-based and neural scansion of poetry](#). In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 18–23, Varna, Bulgaria. INCOMA Ltd.
- Manex Agirrezabal, Iñaki Alegria, and Mans Hulden. 2017b. [A comparison of feature-based and neural scansion of poetry](#). *arXiv preprint arXiv:1711.00938*.
- Khalid Alnajjar and Mika Hämmäläinen. 2018. [A master-apprentice approach to automatic creation of culturally satirical movie titles](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 274–283, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Arto Anttila and Ryan Heuser. 2016. [Phonological and metrical variation across genres](#). *Proceedings of the Annual Meetings on Phonology*, 3(0).
- Derek Attridge. 1982. *The Rhythms of English Poetry*. English language series. Longman, London; New York.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. [Neural machine translation by jointly learning to align and translate](#). *arXiv preprint arXiv:1409.0473*.
- William S. Baring-Gould. 1988. *The Lure of the Limerick: An Uninhibited History*. Clarkson Potter.
- Timo Baumann, Hussein Hussein, and Burkhard Meyer-Sickendiek. 2018. [Style detection for free verse poetry from text and speech](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1929–1940, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- George Belknap. 1981. History of the limerick. *The Papers of the Bibliographical Society of America*, 75(1).
- Cyril Bibby. 1978. *The Art of the Limerick*. The Research Publishing Co., London.
- Jonathan D Culler and Ben Glaser, editors. 2019. *Critical Rhythm: The Poetics of a Literary Life Form*. Fordham University Press. OCLC: 1076879939.
- Javier de la Rosa, Alvaro Perez, Laura Hernandez, Salvador Ros, and Elena Gonzalez-Blanco. 2020. [Rantanplan, fast and accurate syllabification and scansion of Spanish poetry](#). *Procesamiento del Lenguaje Natural*, 65:83–90.
- Alex Estes and Christopher Hench. 2016. [Supervised machine learning for hybrid meter](#). In *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*, pages 1–8, San Diego, California, USA. Association for Computational Linguistics.
- Poetry Foundation. 2021. [There was an Old Man with a Beard by Edward Lear](#).
- Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. [Hafez: an interactive poetry generation system](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, pages 43–48.
- Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. [Automatic analysis of rhythmic poetry with applications to generation and translation](#). In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 524–533, Cambridge, MA. Association for Computational Linguistics.
- Thomas Haider. 2021. [Metrical tagging in the wild: Building and annotating poetry corpora with rhythmic features](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3715–3725, Online. Association for Computational Linguistics.
- Michael Hammond. 2013. [Calculating syllable count automatically from fixed-meter poetry in English and Welsh*](#). *Literary and Linguistic Computing*, 29(2):218–233.
- S. Blair Hedges. 2002. [The origin and evolution of model organisms](#). *Nature Reviews Genetics*, 3(11):838–849.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Computation*, 9(8):1735–1780.
- Roman Jakobson, Krystyna Pomorska, and Stephen Rudy. 1987. *Language in Literature*. Belknap Press, Cambridge, Mass.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. [Conditional random fields: Probabilistic models for segmenting and labeling sequence data](#). In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

- Sidney Lanier. 1979. *The Science of English Verse*. Folcroft Library Editions, Folcroft, Pa.
- Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. [Deep-speare: A joint neural model of poetic language, meter and rhyme](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1948–1958, Melbourne, Australia. Association for Computational Linguistics.
- G. Legman. 1969. *The Limerick. 1700 Examples, With Notes, Variants and Index*. Bell Publishing Co., New York.
- G. Legman. 1980. *More Limericks. 2750 Unpublished Examples, American and British*. Bell Publishing Co., New York.
- Marjorie Levinson. 2007. [What is new formalism?](#) *PMLA*, 122(2):558–569.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#).
- Hoyt Long and Richard Jean So. 2016. Literary pattern recognition: Modernism between close reading and machine learning. *Critical inquiry*, 42(2):235–267.
- Guillermo Marco, Javier de la Rosa, Julio Gonzalo, Salvador Ros, and Elena González-Blanco. 2021. [Automated metric analysis of spanish poetry: Two complementary approaches](#). *IEEE Access*, 9:51734–51746.
- Nina McCurdy, Vivek Srikumar, and Miriah Meyer. 2015. [RhymeDesign: A tool for analyzing sonic devices in poetry](#). In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 12–22, Denver, Colorado, USA. Association for Computational Linguistics.
- Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, MA.
- Naoaki Okazaki. 2007. [CRFSuite: A fast implementation of conditional random fields \(CRFs\)](#).
- Mary Poovey. 2001. The model system of contemporary literary criticism. *Critical Inquiry*, 27(3):408–438.
- Alex Preminger, Terry V.F. Brogan, and Frank J. Warnke, editors. 1993. *New Princeton Encyclopedia of Poetry and Poetics*. Princeton University Press.
- Ashish Vaswani, Samy Bengio, Eugene Brevdo, François Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. [Tensor2tensor for neural machine translation](#). *CoRR*, abs/1803.07416.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.
- Jiayou Wang, Xiaoxuan Zhang, Yuren Zhou, Christopher Suh, and Cynthia Rudin. 2021. [There once was a really bad poet, it was automated but you didn’t know it](#). *CoRR*, abs/2103.03775.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The benchmark of linguistic minimal pairs for English](#). *Transactions of the Association for Computational Linguistics*, 8:377–392.

8 Appendix: A Random Selection of “Failed” Limericks

Here is a random sample of an additional five limericks that did not pass our rhyme test. Author names listed along with OEDILF number.

- Beriberi’s a scary disease;
It appears when one vitamin B’s
Too deficient. You need
To remember to feed
On more egg rolls and liver and peas.
Author: Chris Doyle, #21528
- Lulu Glaser, a lucky chorine,
Shone on Broadway at just 17.
Did she dream, as a child,
That the stalls would go wild,
Shouting, “Lulu! The great actorine!”?
Author: Janet McConnaughey,
#100961
- God a code virus bloggig by doze,
Ad by ’puter’s god dutz. I subboze
Dat a virus idfected
Ids code. I’b dejected.
Wud virus is bad. Doo just blows.
Author: Janet McConnaughey,
#99468
- A caper who’s all caper-witted
Is not a corsair who’s well fitted
For fightin’ and luffin’.
He’ll cut capers on nuffin’
When brainier pirates have flitted.
Author: Janet McConnaughey,
#63637
- In Osoyoos, BC (this is rare),
All four tires went flat! We’d no spare.

What did CAA say?
“You are kidding us, eh?
Like this desert, you’re full of hot air.”
Author: MikeAq, #110550