

Constrained Sampling of Markov Chains

D. Bulger^a, T. Powles^b

^a*Statistics Department, Macquarie University, Sydney, NSW 2109*

^b*Spacejunk III Studios, PO Box 3041, Putney NSW 2112*
Email: david.bulger@mq.edu.au

Abstract: Homogeneous Markov chains with discrete state-space and time are very straightforward to simulate, due to the memorylessness property. In this talk, we consider sampling from the normalised restriction of a Markov chains joint distribution to a subset defined by two kinds of constraints: univariate constraints, restricting individual chain variables to belong to certain sets, and multivariate constraints, restricting non-adjacent groups of chain variables to be unequal but equivalent under a predefined equivalence relation. The multivariate constraints, in particular, present some challenges. An efficient sampling method involving Metropolis-Hastings with some pre-tabulated distributions is described.

We demonstrate an application to text generation, and in particular the random generation of rhyming, scanning lyrics, as a component of an algorithmic songwriting project. In this setting, the states of the Markov chain are syllables (that know which words they are from, so that for instance the final syllables of *perilous* and *marvellous* are two different states). The univariate constraints enforce the metre of the text, by indicating which syllables must be stressed or unstressed (for correct scansion) and which syllables must be word-final (so that words do not straddle lines). The multivariate constraints enforce a rhyming scheme, by requiring certain groups of syllables to rhyme without being equal.

This work relies on the Carnegie Mellon Pronouncing Dictionary (CMUdict; Weide, 1998) and sample text. CMUdict contains over 130000 entries, comprising words, proper nouns et cetera as used in spoken American English. This dictionary indicates the phonemic and stress pronunciation of each entry.

A Markov chain including the entire dictionary would require over 330000 syllable states. We have used sample texts, both to reduce the state space, and to build a first-order Markov model of word sequence, or equivalently, syllable sequence. The problem of interest is to sample from the Markov chain, subject to the constraints imposed by a given rhyming scheme.

Our algorithm firstly samples the multivariately constrained chain variables. After location of a feasible subsequence of rhymed syllables, ratios of joint probabilities are calculated and compared in a Metropolis-Hastings algorithm, where each candidate arises from the incumbent by replacing one set of rhyming syllables with another drawn from a pre-tabulated distribution. After a burn-in period, the skeleton provided by the rhymed syllables is fleshed out by sampling the unrhymed syllables according to the Markov chain, subject to metrical and boundary conditions.

Keywords: *Constrained Markov chain, Metropolis, rhyme, text generation*

1 INTRODUCTION

This paper forms part of a creative algorithmic songwriting project, in which song lyrics, musical structures, melodies and harmonies, and possibly timbres will be generated algorithmically, or interactively with algorithmic input. In developing a lyric generation model, the first method we considered was a one-step Markov chain constrained to rhyme and scan. This model was surprisingly challenging to implement. This paper describes our implementation, via Metropolis-Hastings sampling, and the resulting output.

The role of randomness in the creative process has been investigated by many artists and theorists. Artistic works may be ‘curated’ from random stimuli, as in found poetry (Seely, 2003) and the cut-up technique (Ryding et al., 2014), or produced according to a probabilistic model designed by the artist, in visual art (e.g., Challinor, 1971) and in orchestral works such as by Cage and Xenakis (Roberts, 2016, pp. 280–282). Artists employ random methods as part of a “methodical quest for the new” (Díaz, 2011), and in recognition of the challenged position of authorial intent in modern critical theory (Wimsatt and Beardsley, 1946; Barthes, 1967).

General (that is, non-rhyming) random text generation has been implemented according to various probabilistic models, including one- or multi-step Markovian character or word models (Shannon, 1948), recurrent neural networks (Sutskever et al., 2011) and discourse structure relations (McKeown, 1985; Hovy, 1993).

More recently, several authors have considered the random generation of rhyming verse, based on models such as knowledge-based systems (Díaz-Agudo et al., 2002), recurrent neural networks (Zhang and Lapata, 2014), grammatical templates (Oliveira, 2012) and finite state acceptors (Ghazvininejad et al., 2016).

The aesthetic interest of the liminal space between order and chaos (Challinor, 1971) may be grounded in its apophenic and polysemic potential, and in involving the audience in the creative process (Tormey and Tormey, 1983). For our project, to match the aesthetic of the musical material, we seek output that is blatantly computer-generated, and yet has tantalising suggestions of jumbled, inchoate meanings. Thus we do not require the long-range coherence of the recent methods cited above. Accordingly, we seek a simple, efficient model to produce random text subject only to the formal constraints of rhyme and scansion.

2 DATA SOURCES

Two types of data are required as input: text samples, in order to estimate the one-step Markovian text model, and a pronunciation dictionary, in order to formulate the rhyme and scansion constraints.

For legal simplicity, in this study we use classic novels that are out of copyright: Jane Austen’s 1817 *Northanger Abbey* and Herman Melville’s 1851 *Moby-Dick*, both sourced from Project Gutenberg (2017). Each source was preprocessed by removing all punctuation other than apostrophes and capitalising all text.

An indispensable resource to this project is the CMU Pronunciation Dictionary (Weide, 1998), which gives pronunciation (both phonemic and stress) for over 125 000 words (and proper nouns et cetera) used in modern American English. For comparison, a typical vocabulary for an adult native speaker of English is generally estimated at under 50 000 (Brysbaert et al., 2016). A typical entry in the dictionary (a plain text file) reads

STOCHASTICALLY S T OW0 K AE1 S T IH0 K AH0 L IY2

and indicates how to pronounce ‘stochastically.’ The tokens in this entry are separated by spaces. The first identifies the word, and the remaining tokens specify its phonemes. The vowels are syllable nuclei, and suffixed with numerals 0, 1 or 2 to indicate stress. We conflate primary and secondary stress, categorising every syllable as stressed (1 or 2) or unstressed (0).

3 PROBLEM

Rhyme and scansion (Brown et al., 2006, pp. 605–619) are two features common (though not universal) to English language poetry. As this topic is likely familiar, an example will suffice. The first stanza of W.H. Mearns’s 1899 poem *Antigonish* (Mearns, 1992) reads as follows:

As I was going up the stair, I met a man who wasn’t there!
He wasn’t there again today, oh how I wish he’d go away!

The underlined syllables are stressed. *Scansion* is the requirement that the stressed syllables occur according to some regular pattern; in this case, a simple alternation between unstressed and stressed (iambic tetrameter).

A_1	The set of unstressed syllables	A_2	The set of stressed syllables
A_3	The set of unstressed word-initial syllables	A_4	The set of stressed word-initial syllables
A_5	The set of unstressed word-final syllables	A_6	The set of stressed word-final syllables
A_7	The set of unstressed monosyllables	A_8	The set of stressed monosyllables

Table 1. Each of the sets A_1, \dots, A_8 is a subset of the syllables appearing in the generating lexicon, that is, in the words found in both the sample text and the CMU Dictionary. We include A_7 and A_8 for completeness, though in practice we allowed any monosyllabic word to appear in a stressed or unstressed position, so that A_7 and A_8 were equal, comprising every single-syllable word.

Rhyme is the requirement that given sets of syllables (shown in red and blue) sound the same, from the vowel onward. A syllable’s phonemic content from the vowel onward is called its *rime* (Kirtley et al., 1989).

Other common requirements in verse are that rhymes not be formed by repeating a word, and that the end of each line be the end of a word.

3.1 Mathematical formulation

Sampling from an ordinary Markov chain is straightforward. Here we wish to sample conditionally on some constraints, that is, to sample from the *normalised restriction* of the joint distribution described by the Markov chain to a subset of the sample path space described by the constraints. We consider constraints of two kinds: *univariate constraints*, which restrict the values of individual variables within the Markov chain, and *multivariate constraints*, which constrain chain variables relative to each other. As applied to the lyric generation problem, the states of the Markov chain represent syllables in words, the univariate constraints represent the scansion and word-boundary requirements, and the multivariate constraints represent the rhyme requirements.

Consider a finite-length homogeneous Markov chain $(X_t : t \in \{1, \dots, T\})$ with discrete state space $S = \{1, \dots, |S|\}$ and one-step transition probability matrix M . Assume there is a unique stationary distribution μ , and that $X_0 \sim \mu$. Let $\mathcal{A} = (A_1, \dots, A_U)$ be a sequence of subsets of S , and $\mathcal{B} = (B_1, \dots, B_V)$ be more specifically a partition of S . A *univariate constraint* requires a chain variable to belong to a particular set in \mathcal{A} , e.g., $X_i \in A_j$. A *multivariate constraint* requires that two or more chain variables be distinct members of the same set in \mathcal{B} , e.g., $X_i \neq X_j$ & $\exists k X_i, X_j \in B_k$. In our application, $U = 8$, with the subsets in \mathcal{A} as shown in Table 1. The subsets in \mathcal{B} are the rimes appearing in the lexicon.

Noting that one of the sets A_i may equal S , we can assume without loss of generality that exactly one univariate constraint applies to each chain variable. Therefore let $u = (u_t | t = 1, \dots, T)$ be a sequence of indices in $\{1, \dots, U\}$ encoding the constraints

$$X_t \in A_{u_t}. \quad (1)$$

Also we suppose there are R multivariate constraints encoded into an $R \times T$ binary matrix Q ; in particular, if $Q_{rs} = Q_{rt} = 1$, then a multivariate constraint applies between X_s and X_t , i.e.,

$$(Q_{rs} = Q_{rt} = 1) \implies (\exists k X_s, X_t \in B_k). \quad (2)$$

Multivariate constraints are clearly transitive, and we *do* allow the rows of Q to have more than two entries of 1. On the other hand, any two multivariate constraints sharing a time index can be combined into one row in Q , and we therefore assume without loss of generality that no column of Q has more than one positive entry.

Lastly, to avoid self-rhymes, we impose the constraint that

$$(s \neq t \text{ \& } Q_{rs} = Q_{rt} = 1) \implies (X_s \neq X_t). \quad (3)$$

4 ALGORITHM

The problem we first considered ignored self-rhyme avoidance; we sought to sample from the normalised restriction of the Markov chain (X_t) subject to the constraints (1) and (2). For brevity, our solution of this problem is not presented in detail. Our algorithm comprised one phase to sample the multivariately constrained chain variables (that is, the rhymed syllables) via Gibbs sampling, followed by another to sample the remaining

Markovian arcs according to univariate and boundary constraints. This solution relied on exact calculation of conditional probabilities of rhymed syllables, conditional on other fixed syllables and the constraints.

Under the constraint (3), these conditional probabilities become difficult to calculate exactly, and thus we instead use a Metropolis-Hastings algorithm to sample the multivariately constrained chain variables.

Sampling the univariately constrained variables, conditionally on fixed values of the multivariately constrained variables, is relatively straightforward, and can be done efficiently in a second phase. This is a sequence of inhomogeneous *Markovian bridges*, as defined in Fitzsimmons et al. (1993), spanning the gaps between consecutive rhymed syllables, though much simpler to deal with because our time and state spaces are discrete.

For application to the random verse problem, some preprocessing is required. Firstly, we determine the set of words appearing in both the text sample and the dictionary, to give the problem's *lexicon*. As there may be a number of words, names et cetera that appear in the text sample but not in the dictionary, this fragments the text sample into a number of sequences of known words. The state space S is the collection of syllables in the lexicon. Each word in the lexicon contributes one or more states; for instance, the syllable '-lous' at the end of the word 'marvelous' would be a distinct state from the syllable '-lous' at the end of the word 'perilous.' The state space is indexed by the set $\{1, \dots, N\}$.

The text sample fragments are then used to estimate the $N \times N$ transition matrix M of a one-step Markov chain on the syllable state space. This estimate is made in direct proportion to the transition frequencies seen in the fragments, with the exception that states only observed at the ends of fragments are assumed to transition to the first syllable of any word in the lexicon, in proportion to their occurrence in the fragments. Naturally, each non-word-terminal syllable must only transition to the following syllable in the same word, but the terminal syllables of common words will generally have multiple positively-weighted following states.

The frequency of words in the fragments gives a good initial approximation to the stationary distribution μ of M . This approximation is multiplied by the transition matrix until it stops changing (to within a tolerance).

Within the lexicon, each syllable's word position (word-interior, word-initial, word-final, monosyllable), dichotomous stress and rime are determined.

4.1 Algorithm

We employ an MCMC variant sometimes known as 'Metropolis-Hastings-within-Gibbs' (Gilks, 1996, p. 85) in order to sample multivariately constrained chain variables. After sufficient burn-in, the remaining chain variables can be efficiently sampled by exact computation of their conditional distributions.

The Constrained Markov Sampling (CMS) algorithm is shown in Algorithm 1. The initialisation phase determines the subsets T_Q^1, \dots, T_Q^R of the index set $\{1, \dots, T\}$ that correspond to each of the R multivariate constraints, so that, for each r , the r th multivariate constraint is that the states X_t for $t \in T_Q^r$ be distinct but all belong to one set in the partition \mathcal{B} ; e.g., the syllables at positions indexed by T_Q^r must all rhyme but differ. It is also convenient to define T_Q as their union, the set of *all* multivariately constrained variables' time indices.

To fully specify the CMS algorithm, we must select a proposal distribution g_K for each $K = 2, 3, \dots$, according to which the algorithm will initialise and generate candidates for each multivariately constrained set of chain variables, and the stopping criterion. Also, to make the algorithm explicit, we should clarify how the probabilities appearing in it are computed.

The sampling distributions g_K are only required for values of K such that a set of K chain variables appears in a multivariate constraint. In our application, $K = 2, 3$ may be the only necessary cases, since rhyming schemes rarely require more than three syllables all to rhyme. A proposal distribution similar to the required conditional distributions will improve convergence. With this aim, and letting $K = |T_Q^r|$, we simply sample v_r according to the distribution h_K of the rime if K variables are drawn from μ conditionally on their all rhyming but being distinct (but otherwise independently), and then sample the variables x'_t in random order without replacement from the restriction of μ to that rime. We name g_K the resulting multivariate distribution on $(x'_{t_1}, \dots, x'_{t_K})$, and we expect this to be a reasonable approximation for the required conditional distribution of $(X_t : t \in T_Q^r)$ given the values of X_t for $t \in T_Q \setminus T_Q^r$. The distribution $h_{|T_Q^r|}$ can easily be pretabulated via a recursive calculation.

For brevity we consider only the main loop probability ratio, in the special case where r is neither the earliest nor the latest rhyme appearing in the rhyming scheme. That is, we assume that

$$\rho = \max \{t \in T_Q \mid t < \min T_Q^r\} \quad \text{and} \quad \sigma = \min \{t \in T_Q \mid t > \max T_Q^r\}$$

CMS($M, \mathcal{A}, \mathcal{B}, u, Q$)

Initialisation:
Determine the stationary distribution μ of M .
For each $r \in \{1, \dots, R\}$:
 Let $T_Q^r \leftarrow \{t \in \{1, \dots, T\} \mid Q_{rt} = 1\}$.
Let $T_Q \leftarrow \cup_{r=1}^R T_Q^r$.
For each $r \in \{1, \dots, R\}$:
 Until $P[X_t = x_t \forall t \in T_Q^r \mid X_t = x_t \forall t \in \cup_{s < r} T_Q^s \& X_t \in A_{u_t} \forall t] > 0$,
 Sample $(x_t : t \in T_Q^r)$ according to a fixed joint distribution $g_{|T_Q^r|}$.

Main loop:
Repeat until satisfaction of a stopping criterion:
 Choose r uniformly at random from $\{1, \dots, R\}$:
 Sample $(x'_t : t \in T_Q^r)$ according to $g_{|T_Q^r|}$.
 Evaluate the probability ratio $f_L = \frac{P[X_t = x'_t \forall t \in T_Q^r \& X_t = x_t \forall t \in T_Q \setminus T_Q^r \& X_t \in A_{u_t} \forall t]}{P[X_t = x_t \forall t \in T_Q \& X_t \in A_{u_t} \forall t]}$.
 Evaluate the proposal ratio $f_P \leftarrow g_{|T_Q^r|}(x_t : t \in T_Q^r) / g_{|T_Q^r|}(x'_t : t \in T_Q^r)$.
 With probability $\min\{1, f_L f_P\}$, let $x_t \leftarrow x'_t$ for each $t \in T_Q^r$.

Final univariately constrained sampling:
For $s = \min T_Q - 1, \dots, 1$:
 Sample x_s according to $P[X_s = j \mid X_t \in A_{u_t} \forall t \in \{1, \dots, T\} \& X_{s+1} = x_{s+1}]$.
For $s = \max T_Q + 1, \dots, T$:
 Sample x_s according to $P[X_s = j \mid X_t \in A_{u_t} \forall t \in \{1, \dots, T\} \& X_{s-1} = x_{s-1}]$.
For each remaining gap, with x_p and x_q assigned, $p+1 < q$, but x_s unassigned $\forall p < s < q$:
 For $s = p+1, \dots, q-1$:
 Sample x_s according to $P[X_s = j \mid X_t \in A_{u_t} \forall t \in \{1, \dots, T\} \& X_{s-1} = x_{s-1} \& X_q = x_q]$.

Algorithm 1. Constrained Markov Sampler algorithm. All probabilities are with respect to the sample path measure P of the unconstrained, length- T Markov chain with initial distribution μ and transition matrix M .

both exist. The other probabilities required in Algorithm 1 can be calculated according to similar principles.

Letting C'_t and C_t denote either A_{u_t} , $\{X'_t\}$ or $\{X_t\}$ as required, we have

$$\begin{aligned}
f_L &= P[X_t = x'_t \forall t \in T_Q^r \& X_t = x_t \forall t \in T_Q \setminus T_Q^r \& X_t \in A_{u_t} \forall t] / P[X_t = x_t \forall t \in T_Q \& X_t \in A_{u_t} \forall t] \\
&= P[X_t \in C'_t \forall t \in \{\rho+1, \dots, \sigma\} \mid X_\rho = x_\rho] / P[X_t \in C_t \forall t \in \{\rho+1, \dots, \sigma\} \mid X_\rho = x_\rho] \\
&= e_{x_\rho}^\top \left(\prod_{t=\rho+1}^{\sigma-1} M \Delta(C'_t) \right) e_{x_\sigma} / e_{x_\rho}^\top \left(\prod_{t=\rho+1}^{\sigma-1} M \Delta(C_t) \right) e_{x_\sigma},
\end{aligned}$$

where $\Delta(C)$ is the $|S| \times |S|$ diagonal matrix with entries $\Delta(C)_{jk} = 1$ where $j = k \in C$, and 0 otherwise. The $\Delta(C)$ multiplications are implemented by elementwise multiplications by pretabulated length- $|S|$ vectors.

5 OUTPUT

Algorithm 1 was implemented in Matlab 2017a on a Dell Latitude E7470 with an Intel Core i7-6600U CPU and 16 GB of RAM. The two text samples are *Northanger Abbey* at 77 682 words and *Moby-Dick* at 216 109 words. Preprocessing of these samples to determine the lexicons, transition matrices, stationary distributions, stress sets \mathcal{A} and rime partitions \mathcal{B} took 67 seconds and 524 seconds, respectively. We slightly varied Algorithm 1 in practice, by running it repeatedly, with each subsequent run initialised from the rhyme-constrained syllables of the previous run. The stopping criterion was that each rime had changed. Our method then produced ten limericks from each sample, taking 120 seconds based on *Northanger Abbey* and 1247 seconds based on *Moby-Dick*. Here are one limerick selected from each sample:

SINCERELY DID NOT AT LENGTH TO
OBLIGE SUCH ATROCITIES DO
IN FINDING IT THAT
A LITTLE YOU AT
THE MEANS OF A QUARTER OF TWO

OMNIVOROUS ROVING WHERE HE
REFUSED IT THE MATE CAME FROM ME
IN COAT STALKED AWAY
THE OARSMAN COULD THEY
THEMSELVES SO AS THOUGH THE DART BE

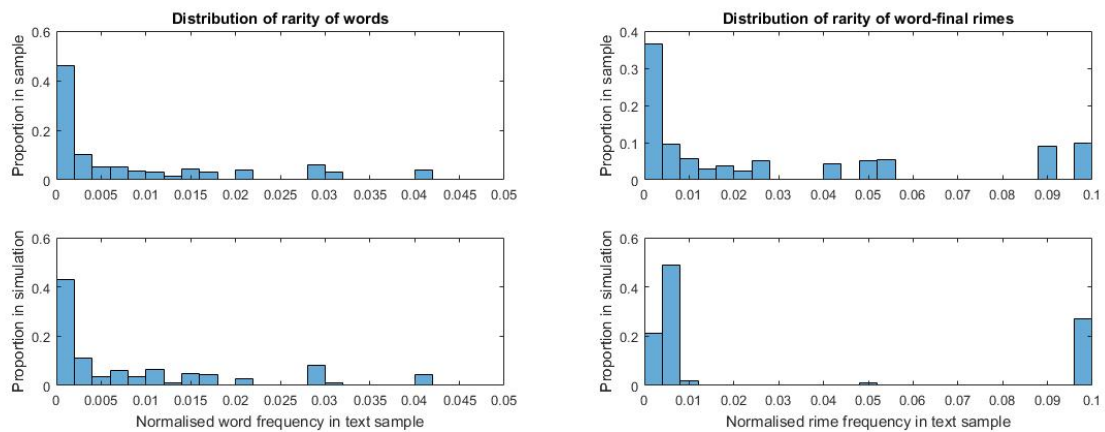


Figure 1. Distributions of lexical proportion and word-final rime proportion in *Northanger Abbey* and in 50 simulated limericks. The rime distribution in the simulated output is skewed toward common rimes.

An interesting insight is gained by considering the distribution of rare and common words and (word-final) rimes appearing in the output. Each word in the lexicon appears at least once in the text sample, but common words such as TO may appear many times; accordingly, each lexicon word can be assigned a “lexical proportion.” The distribution of the lexical proportions of words used characterises writing style, indicating whether the text mostly sticks to the few most common words, or relies more evenly on a wider vocabulary.

Were it not for the rhyme and scansion constraints, we would expect the distribution of words output by the algorithm to approximate the lexical proportions. But since these discrete distributions have very large supports, it is difficult to test that directly. Figure 1 (left) focuses on only the lexical proportions of the words seen, rather than the words themselves, demonstrating an apparent similarity between the lexical proportion distributions. This suggests that the constraints do not greatly alter the vocabulary, or at least, do not favour rarer or more common words.

Figure 1 (right) shows a similar comparison of the word-final rimes appearing in the text sample and those appearing in rhyme-constrained positions in the output. The algorithm polynomially favours more common rimes; it seems that because we are conditionally sampling from a Markov chain, it is extremely unlikely that two or three syllables constrained to rhyme will select a less common rhyme.

6 CONCLUSION

We have found a solution to the problem of Subsection 3.1 suitable for our application. A limitation is that, for small lexicons or strict, crowded rhyming schemes, the feasible space might become disconnected under the proposal distribution, that is, the Markov chain resulting from the main loop might become reducible. An unforeseen issue with the problem formulation is that it polynomially favours more common rimes.

In future work, we hope to replace the Markov model with a more sophisticated probabilistic text model. A generative model (Prince and Smolensky, 2008) is appealing, as it may coordinate well with generative models of music composition (Lerdahl and Jackendoff, 1985).

REFERENCES

- Barthes, R. (1967). The death of the author. *Aspen* (5–6).
- Bratko, A., G. V. Cormack, B. Filipič, T. R. Lynam, and B. Zupan (2006). Spam filtering using statistical data compression models. *Journal of Machine Learning Research* 7(Dec), 2673–2698.
- Brown, K., A. H. Anderson, L. Bauer, M. S. Berns, J. E. Miller, and G. Hirst (2006). *Encyclopedia of language & linguistics*, Volume 1. Elsevier Amsterdam.
- Brysbaert, M., M. Stevens, P. Mandera, and E. Keuleers (2016). How many words do we know? practical estimates of vocabulary size dependent on word definition, the degree of language input and the participants age. *Frontiers in psychology* 7.

- Challinor, M. (1971). Change, chance and structure: Randomness and formalism in art. *Leonardo* 4(1), 1–11.
- Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational linguistics* 29(4), 589–637.
- Díaz, L. (2011). By chance, randomness and indeterminacy methods in art and design. *Journal of Visual Art Practice* 10(1), 21–33.
- Díaz-Agudo, B., P. Gervás, and P. A. González-Calero (2002). Poetry generation in colibri. In *Proceedings of 6th European Conference on Advances in Case-Based Reasoning (ECCBR 2002)*, pp. 73–102. Springer.
- Fitzsimmons, P., J. Pitman, and M. Yor (1993). Markovian bridges: construction, palm interpretation, and splicing. In *Seminar on Stochastic Processes, 1992*, pp. 101–134. Springer.
- Ghazvininejad, M., X. Shi, Y. Choi, and K. Knight (2016). Generating topical poetry. In *EMNLP*, pp. 1183–1191.
- Gilks, W. R. (1996). Full conditional distributions. In *Markov Chain Monte Carlo in Practice*, pp. 75–88. Chapman & Hall: London.
- Hovy, E. H. (1993). Automated discourse generation using discourse structure relations. *Artificial intelligence* 63(1), 341–385.
- Kirtley, C., P. Bryant, M. MacLean, and L. Bradley (1989). Rhyme, rime, and the onset of reading. *Journal of experimental child psychology* 48(2), 224–245.
- Lerdahl, F. and R. Jackendoff (1985). *A generative theory of tonal music*. MIT press.
- McKeown, K. R. (1985). Discourse strategies for generating natural-language text. *Artificial Intelligence* 27(1), 1–41.
- Mearns, W. H. (1992). Antigonish. In M. Gardner (Ed.), *Best Remembered Poems*. Courier Corporation.
- Oliveira, H. G. (2012). Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence* 1, 21.
- Pang, B., L. Lee, et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1–2), 1–135.
- Prince, A. and P. Smolensky (2008). *Optimality Theory: Constraint interaction in generative grammar*. John Wiley & Sons.
- Project Gutenberg (2017). www.gutenberg.org.
- Roberts, G. (2016). *From Music to Mathematics: Exploring the Connections*. Johns Hopkins University Press.
- Ryding, K., P. Prax, and S. Bachelder (2014). Poetry is for everyone: A comparative analysis of the cut-up technique, Magnetic poetry and the casual word game Words of Oz. Degree Project, Uppsala Universitet. <http://uu.diva-portal.org/smash/get/diva2:733111/FULLTEXT01.pdf>.
- Seely, H. (2003). The poetry of D.H. Rumsfeld. <https://web.archive.org/web/20110605140705/http://www.slate.com/?id=2081042&>.
- Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal* 27, 379–423, 623–656.
- Sutskever, I., J. Martens, and G. E. Hinton (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 1017–1024.
- Tormey, J. and A. Tormey (1983). Art and ambiguity. *Leonardo* 16(3), 183–187.
- Weide, R. (1998). Carnegie Mellon Pronouncing Dictionary. www.speech.cs.cmu.edu/cgi-bin/cmudict.
- Wimsatt, Jr., W. and M. Beardsley (1946). The intentional fallacy. *The Sewanee Review* 54(3), 468–488.
- Zhang, X. and M. Lapata (2014). Chinese poetry generation with recurrent neural networks. In *EMNLP*, pp. 670–680.