A^* Romantic Poetry Generation

Nathanael Fillmore
nathanae@cs.wisc.edu

Introduction

Poetry publication in the United States is a multi-hundred dollar industry. Yet current methods of production are inefficient—they've hardly changed since before the Industrial Revolution. In this paper we present novel methods for training a computer to generate poetry using a corpus. (In all seriousness, it is interesting to see how well we can make the computer create meaning and form when we remove the constraints on content and ordering present in machine translation and typical natural language generation.)

Previous attempts at using computers to automatically generate poetry tend to rely on hand-coded rules. For example, (Gervas 2001) uses a rule-based system to generate Spanish poetry. The rules were manually created by reviewing academic literature on poetry. (Manurung, Ritchie, and Thompson 2000) and (Manurung 2003) use stochastic hill-climing search to create poems. But evaluation and mutation of candidates rely on a hand-crafted grammar and lexicon. (Levy 2001) proposes a similar evolutionary algorithm, but again using a hand-crafted lexicon, conceptual knowledge base, and grammar. Other examples, going back at least to the 1970s, use hand-crafted template poems and fill in the blanks to create new poems. (See §2.3.2 in (Manurung 2003) for an overview.)

On the other hand, several techniques we present here are similar to corpus-based approaches used in machine translation. These are referenced below.

Methods

We propose five novel methods for poetry generation. One is a baseline, three improve separately on the baseline, and the last is a hybrid. All require a tokenized, sentence-segmented corpus of poetry and a trigram (or other n-gram) language model trained on the corpus.

Unconstrained A^* search

Our baseline method is simple. We use A^* search with a cost function based on a trigram language model to find high-probability poems. A^* search (along with similar algorithms like beam search) is frequently used in machine translation systems to order candidate translations, e.g. (Tillman and Ney 2003; Och, Ueffing, and Ney 2001). Our procedure:

1. Sample a subset of the vocabulary from the unigram MLE multinomial based on the corpus.
2. Use memory-bounded A^* search to find the most likely sequence of length l of words from that subset.

The first step, taking a sample of the vocabulary, was needed to fit the problem in memory. The sample size was about 30 times the number of words in each poem. During the second step, A^* search, each candidate partial poem $\mathbf{w} = w_1, \ldots, w_k$ was assigned (inverse) path cost g as follows:

$$g(\mathbf{w}) = \log p(w_1) + \log p(w_2|w_1) + \sum_{i=3}^{k} \log p(w_i|w_{i-2}, w_{i-1})$$

Each candidate was assigned a heuristic value h, the estimated (inverse) path cost to the goal, as follows. Before the search, precompute

$$S(u) = \min_{v_1,v_2 \in V} \log p(u|v_1, v_2), \ \ \forall u \in V$$

where V is the sampled subset of the vocabulary. Then during the search h can be computed efficiently:

$$h(\mathbf{w}) = \min_{\mathbf{u}=\{u_{k+1},u_{k+2},\ldots,u_l\} \subseteq V \setminus \mathbf{w}} \sum_{u_i \in \mathbf{u}} S(u_i)$$

where l is the desired length of the complete poem. Candidates were popped and expanded based on $g + h$, but pruned based on $-g - h - \alpha k$, where α is a small value aimed at discouraging long candidates from being pruned.

IDF templates

Our next method aims to learn templates from the corpus. (Bilingual) template extraction from a corpus, a form of example-based machine translation, has also been used by some machine translation systems, e.g. (Brown 2000; Carl 1999; Lu et al. 2001). Their specific approachs are different than ours; for one thing, parallel texts are involved. Our procedure:

1. First, as a preprocessing step, compute the IDF of each word in the corpus. Replace each word whose IDF is above a threshold (4.0) by a placeholder. These are content words; only stopwords remain. For example,

<S> the budding twigs spread out their fan , to catch the breezy air ; </S> <S> and i must think , do all i can , that there was pleasure there . </S>	→	<S> the X X X X their X , to X the X X ; </S> <S> and i X X , X all i X , that X was X X . </S>

The threshold can be changed, of course; a lower threshold will lead to more generalized templates.

2. Next, before generating a poem, uniformly sample k contiguous lines from the preprocessed corpus, starting at an <S> . This is our template. Note that sampling from a uniform distribution over line tokens is equivalent to sampling from a multinomial distribution over line types. More common patterns will be chosen more frequently.
3. Run A* search as above. But if a stopword occurs at position i in the template, force that word to occur at position i in the generated poem.

POS templates

Our third method learns templates from the corpus in a different way. Instead of fixing stopwords, we force the ith word in the generated poem to have the same part of speech as the ith word in the template. The previous template becomes:

<S> TD VBG NNS VBN RP PRP$ NN ,
TO VB DT JJ NN ; </S>
<S> CC PRP MD VB , VB DT PRP MD ,
DT EX VBD NN RB . </S>

This kind of template requires us to change how we sample the vocabulary. On the one hand, what if our template requires (for example) a PRP but our sample doesn't contain any PRPs? We won't be able to generate a poem! Conversely, although less important, if our sample contains a PRP but the template doesn't call for it, we will waste space, since the PRP will certainly not occur in any generated poem. To solve both problems, we sample the vocabulary from a mixture of unigram MLE multinomials, each restricted to words of a single POS:

$$V = \bigcup_{p \in P} V_p \sim \text{Mult}\{v \in \text{corpus} : \text{pos}(v) = p\}$$

where P is the POS template.

Topic

Our fourth method aims to improve the meaning, rather than the form, of the generated poems. We sample the vocabulary from the unigram MLE based on the subset of sentences $\mathbf{s}$ in the corpus that contain one of a set of given keywords $\mathbf{u}$:

$$V \sim \text{Mult}\{v \in \mathbf{s} : \mathbf{s} \in \bigcup_{u \in \mathbf{u}} \{\mathbf{s} : u \in \mathbf{s}\}\}$$

We assume that sentences in the corpus which contain one of our keywords are related to that keyword, so sampling the vocabulary from only those sentences should encourage the generated poem to have a similar topic. After sampling a vocabulary we use unconstrained A* search as in the baseline.

Combination

Our last method is a hybrid, combining all three refinements. We sample the vocabulary from a mixture of multinomials, each limited to words of a particular POS and drawn from the subset of sentences that contain one of the keywords:

$$V = \bigcup_{p \in P} V_p \sim \text{Mult}\{v \in \mathbf{s} : \text{pos}(v) = p, \mathbf{s} \in \bigcup_{u \in \mathbf{u}} \{\mathbf{s} : u \in \mathbf{s}\}\}$$

We use a template like

<S> the VBG NNS VBN RP their NN ,
to VB the JJ NN : </S>
<S> and i MD VB , VB all i MD ,
that EX was NN RB . </S>

and constrain search by both stopwords and POS tags as before.

Experiments

Evaluating a poetry generator is difficult. (Popescu-Belis 2007) distinguishes two metrics for general NLG systems: distance-based metrics and task-based ones. Distance-based metrics such as BLEU (Papineni et al. 2001) or ROUGE (Lin 2004) are quite unsuitable for evaluating a poetry generator. BLEU, for example, is computed by having a human and a machine translate the same test set; the BLEU score is proportional to the number of shared n-grams. But our generator is *trying* to produce something new—there's no reasonable reference point to measure the distance from.[1]

A task-based metric, based on a user study, is more promising. (a) Generate poems using each of our proposed methods. (Optionally select human-created poems as well, for reference.) (b) Have subjects rate each poem for grammaticality, thematic unity, poetic plausibility, etc. Then (c) compare how significantly ratings vary among different approaches. We have not had time to conduct such a study.

Instead, we present example poems produced using each of our methods. We collected a corpus of 19th-century poetry from Project Gutenberg and built a smoothed trigram language model based on the corpus using the CMU-Cambridge SLM toolkit (Clarkson and Rosenfeld 1997). We wrote a program, based partly on code from (Zhu et al. 2008), to run the A* search.

Figure 1 shows the top 15 poems produced by unconstrained A* search. Fragments of some poems make sense, but overall each poem is ungrammatical and nonsensical. Figure 2 shows the top 15 poems produced by A* search constrained by an IDF template, and figure 3 shows the top 15 poems when a POS template is used. Both results are substantially more plausible than the results from the unconstrained case. Figure 4 shows the top 15 poems when the search is unconstrained but the vocabulary is sampled from a subset of sentences that contain "love" or "tears". The topic does seem to show up—the actual word "tears" occurs in several of the generated poems. Grammatical cohesiveness is worse than in the template-based examples—this is expected—but still better than in the baseline, a bit of a surprise, probably because the sampled vocabulary is more cohesive than in the baseline. Figure 5 shows the top 15 poems using a combination of all our strategies.

[1] Even when there is a point of reference, poetry presents a special challenge. Compare, for example, the English translations of the first stanza of Horace's Ode 1.38 by Gerald Hopkins and William Cowper (in (Carne-Ross and Haynes 1996)). Both were distinguished poets, and their translations were separated by only 50 years—yet they share no words: the unigram BLEU score would be 0!

Acknowledgement

Thanks to Jerry Zhu, Tony Anderson, and Nina Mukherji for helpful discussions.

References

Brown, R. D. 2000. Automated generalization of translation examples. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING-2000)*, 125–131.

Carl, M. 1999. Inducing translation templates for example-based machine translation.

Carne-Ross, D. S., and Haynes, K., eds. 1996. *Horace in English*. Penguin Books.

Clarkson, P., and Rosenfeld, R. 1997. Statistical language modeling using the CMU–cambridge toolkit. In *Proc. Eurospeech '97*, 2707–2710.

Gervas, P. 2001. An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems* 14(3-4):181–188.

Levy, R. P. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the Fourth International Conference on Case Based Reasoning Workshop on Creative Systems: Approaches to Creativity in AI and Cognitive Science*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., ed., *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Lu, Y.; Zhou, M.; Li, S.; Huang, C.; and Zhao, T. 2001. Automatic translation template acquisition based on bilingual structure alignment. *Computational Linguistics* 6(1):83–108.

Manurung, H.; Ritchie, G.; and Thompson, H. 2000. Towards a computational model of poetry generation.

Manurung, H. M. 2003. An evolutionary algorithm approach to poetry generation.

Och, F. J.; Ueffing, N.; and Ney, H. 2001. An efficient A* search algorithm for statistical machine translation. In *Data-Driven Machine Translation Workshop*, 55–62.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2001. Bleu: a method for automatic evaluation of machine translation.

Popescu-Belis, A. 2007. Evaluation of nlg: Some analogies and differences with mt and reference resolution. In *Proceedings of MT Summit XI Workshop on Using Corpora for NLG and MT (UCNLG+MT)*, 66–68.

Tillman, C., and Ney, H. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics* 29(1):97–133.

Zhu, X.; Goldberg, A. B.; Rabbat, M.; and Nowak, R. 2008. Learning bigrams from unigrams. In *The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*.

her . you yet may they
not strong ? and ,
there 's not a soul as strong as could then
be it so ; and
$\log p = -236.7064$

her . " turn where we
will see or no ;
idon . but from thee which ' her
' be ! and ,
$\log p = -236.7294$

her . where is he not
strong ? and , in
idon . – be our ' hand '
of man ; and ,
$\log p = -236.9815$

– o care ! where is
he not strong ? and
there 's not a soul as strong as could then
be it so ! and
$\log p = -237.0587$

her . and we will see
or no ; and ,
her . – be our ' hand '
of man ; and ,
$\log p = -237.1798$

her . " turn where we
will see or no ?
there 's not a soul as strong as could then
be it so ; –
$\log p = -237.2298$

this is noble ! where is
he not strong ? and
and , with a soul as strong as could then
be it so ; and
$\log p = -237.3875$

her . we will see or
no ; and , in
there 's not a soul as strong as could then
be it so ! –
$\log p = -237.4169$

her . – we will see
or no ; and ,
there 's not a soul as strong as could then
be it so . '
$\log p = -237.4769$

her . where is he not
strong ? and , as
idon . but we will see or no
; and , in the
$\log p = -237.5105$

her . – o be thou
mute ! and , in
– idon . and who but this endured
not ; and , in
$\log p = -237.5225$

– where – where – where is he
not strong ? and ,
idon . and there will see or no
; and , in the
$\log p = -237.5322$

" o death ! where is
he not strong ? and
there 's not a soul as strong as could then
be ' there ' on which the
$\log p = -237.5633$

her . where is he not
strong ? and , with
– idon . – be our ' hand
' of man ; and
$\log p = -237.6242$

– o care ! where is
he not strong ? –
idon . but from thee which ' her
' be true ? and
$\log p = -237.6372$

Figure 1: Top 15 poems using unconstrained A* search and a vocabulary sampled from the unigram MLE.

the grassy the greatness of their birth ,
to which the mind 's ;
and i to thee , and all i thought ,
that it was in her .
$\log p = -263.280135$

the most the greatness of their birth ,
to see the glory of ;
and i , who , with all i thought ,
that it was a time .
$\log p = -263.292844$

the was the greatness of their birth ,
to see the glory of ;
and i , who , with all i thought ,
that there was a time .
$\log p = -263.308526$

the potent the greatness of their birth ,
to which the blast , ;
and i , who , with all i thought ,
that it was in her .
$\log p = -263.352116$

the man the greatness of their birth ,
to see the glory of ;
and i to thee , and all i thought ,
that it was a time .
$\log p = -263.519144$

the was the greatness of their birth ,
to which the mind 's ;
and i to thee , and all i see ,
that it was a time .
$\log p = -263.538126$

the or the greatness of their birth ,
to see the glory of ;
and i to thee , and all i thought ,
that there was a time .
$\log p = -263.581126$

the most the greatness of their birth ,
to which the blast , ;
and i , who , with all i see ,
that it was a time .
$\log p = -263.600326$

the was the greatness of their birth ,
to which the blast , ;
and i , who , with all i see ,
that there was a time .
$\log p = -263.616007$

the was the greatness of their birth ,
to which the shepherd 's ;
and i , who , with all i was ,
that there was a time .
$\log p = -263.633826$

the sail the greatness of their birth ,
to see the glory of ;
and i be silent , and all i thought ,
that it was a time .
$\log p = -263.691835$

the a the greatness of their birth ,
to see the glory of ;
and i be silent , and all i thought ,
that there was a time .
$\log p = -263.708216$

the noble the greatness of their birth ,
to see the glory of ;
and i , who , with all i be ,
that it was a time .
$\log p = -263.724335$

the queen the greatness of their birth ,
to see the glory of ;
and i , who , with all i be ,
that there was a time .
$\log p = -263.730235$

the be the greatness of their birth ,
to see the glory of ;
and i to thee , and all i be ,
that it was a time .
$\log p = -263.800726$

Figure 2: Top 15 poems using A* search constrained by an IDF template, with a vocabulary sampled from the unigram MLE.

the yielding they been in his turn ,
to have a noble mind ;
and it may be , hope that we might ,
that there was she there .
$\log p = -266.247216$

the yielding was left in his turn ,
to have a noble mind ;
and it may be , like that we might ,
that there was she there .
$\log p = -266.293316$

the yielding they been in his death ,
to have a noble mind ;
and it may be , hope that we might ,
that there was need there .
$\log p = -266.469226$

the yielding was left in his death ,
to have a noble mind ;
and it may be , like that we might ,
that there was need there .
$\log p = -266.515326$

the yielding was made in her turn ,
to have a noble mind ;
and it may be , will that we might ,
that there was need there .
$\log p = -266.828016$

the yielding they been in her turn ,
to have a noble mind ;
but it will be , hope that we might ,
that there was need there .
$\log p = -267.272916$

the yielding was left in her turn ,
to have a noble mind ;
and it may be , like this we might ,
that there was need there .
$\log p = -267.318816$

a yielding they been in his turn ,
to have a noble mind ;
but it will be , like that we might ,
that there was need there .
$\log p = -267.440416$

a yielding was left in his turn ,
to have a noble mind ;
but it will be , like that she may ,
that there was need there .
$\log p = -267.567107$

the ascending they been in his turn ,
to have a noble mind ;
but it will be , hope that we may ,
that there was need there .
$\log p = -267.763907$

the ascending was left in his turn ,
to have a noble mind ;
and it will be , hope that we might ,
that there was need there .
$\log p = -267.788916$

a yielding they been in his death ,
to have a noble mind ;
but it will be , like this we might ,
that there was need there .
$\log p = -267.825226$

a yielding was left in his death ,
to have a noble mind ;
but it will be , like that we may ,
that there was need there .
$\log p = -267.835717$

the ascending they been in his death ,
to have a noble mind ;
but it will be , hope that we might ,
that there was need no !
$\log p = -267.854226$

the ascending was left in his death ,
to have a noble mind ;
and it will be , like that we might ,
that there was need there .
$\log p = -267.864226$

Figure 3: Top 15 poems using A* search constrained by a POS template, with a vocabulary sampled from a mixture of multinomials.

– no – hopes spun in truth a
human voice hath said ;
and , as he may do all i can not
but be thou blest !
$\log p = -237.780191$

no – hopes spun in simple truth a
human voice hath said ;
and there he may do all i can not but
be thou blest ! and
$\log p = -237.849291$

to have a heart that had found a
human voice hath said ;
but , as he may do all i can not
but be thou blest !
$\log p = -238.1624$

– no – hopes spun in her –
she hath eyes : –
her . but i can not but be
thou blest ! and ,
$\log p = -238.2489$

– no – hopes spun in her –
she hath eyes : the
her . no turn she can not but
be thou blest ! and
$\log p = -238.429$

– no – hopes spun in her light
to thy heart ; –
her . i can not but be thou
blest ! and , as
$\log p = -238.7468$

no – hopes spun in truth a human
voice hath said ; –
and , as one whose own country , far less
could we but have thou there the
$\log p = -238.808691$

thou whose happy hand had found a human
voice hath said ; –
and all that he may do all i can not
but be thou blest !
$\log p = -238.841091$

– o ye , in simple truth a
human voice hath said ;
and he may do all i can not but be
thou blest ! and ,
$\log p = -238.8789$

– no – hopes spun in her light
to thy heart ; the
her . and do all i can not
but be thou blest !
$\log p = -238.8972$

no – hopes spun in truth a human
voice hath said ; the
her . and i could not but be
thou blest ! and ,
$\log p = -238.953291$

thou whose happy hand had found a human
voice hath said ; the
her . and , far less could we
but have they not ?
$\log p = -238.985191$

– her tears to flow , thou hast
thou thy own life ;
and that he may do all i can not but
be thou blest ! and
$\log p = -239.065191$

was happy that she had found a human
voice hath said ; –
her hope she won , they can not but be
thou blest ! and ,
$\log p = -239.1032$

a passing cloud , in simple truth a
human voice hath said ;
there he may do all i can not but be
thou blest ! and ,
$\log p = -239.1829$

Figure 4: Top 15 poems using unconstrained A* with a vocabulary sampled from a subset of the corpus matching `love,tears`.

the shedding tears shed on their side ,
to note the old time ;
and i will praise , that all i can ,
that there was aught ever .
$\log p = -264.550417$

the living eyes had in their weakness ,
to note the old time ;
and i will dare , where all i can ,
that there was aught ever .
$\log p = -264.566107$

the shedding tears shed on their side ,
to preserve the old time ;
and i may say , we all i can ,
that there was aught ever .
$\log p = -264.693526$

the living eyes had in their weakness ,
to preserve the old time ;
and i will die , that all i can ,
that there was aught ever .
$\log p = -264.738716$

the shedding tears shed on their side ,
to find the old time ;
and i would give , that all i can ,
that there was aught ever .
$\log p = -264.894126$

the living eyes had in their weakness ,
to find the old time ;
and i must die , that all i can ,
that there was aught ever .
$\log p = -264.978816$

the living eyes had on their side ,
to find the old time ;
and i may say , tis all i can ,
that there was my heart .
$\log p = -265.544198$

the living eyes had in their light ,
to preserve the old time ;
and i may dare , where all i can ,
that there was my heart .
$\log p = -265.590407$

the shedding tears shed on their side ,
to please the gentle heart ;
and i will dare , where all i can ,
that there was my heart .
$\log p = -265.621107$

the living eyes had in their weakness ,
to love the living nature ;
and i must be , that all i can ,
that there was aught ever .
$\log p = -265.781016$

the living eyes had in their birth ,
to note the old time ;
and i may say , we all i can ,
that there was sorrow ever .
$\log p = -265.815216$

the shedding tears shed on their side ,
to love the living god ;
and i may say , where all i can ,
that there was aught ever .
$\log p = -265.819426$

the living eyes had in their weakness ,
to love the living god ;
and i will go , we all i can ,
that there was aught ever .
$\log p = -265.859816$

the living eyes had in their weakness ,
to love the end i ;
and i may say , that all i can ,
that there was my heart .
$\log p = -265.867816$

the shedding tears shed on their side ,
to find the old men ;
and i will die , that all i can ,
that there was my heart .
$\log p = -265.999716$

Figure 5: Top 15 poems using our hybrid method.