Shiraz: Generating Poetry Based on Narrative Schemas

Stephen Ngo and **Jack Ceverha**

1 Introduction

Poetry is a complex phenomenon, as the auditory and semantic qualities of words and human notions of aesthetic are all considerations when writing a poem. The problem of poetry generation has garnered academic interest among researchers.

(Ghazvininejad et al., 2016) describes Hafez, a poetry generation system using an input topic to generate a sonnet related to the input. However, the Hafez system lacks in the ability to write a narrative. We describe Shiraz, an extension of the Hafez system that generates poetry given a narrative schema.

2 Problem Definition and Algorithm

2.1 Task Definition and Algorithm

Current English-language poetry generation systems can capture the structure and rhyme scheme of a poem, as well as simulate literary style to a limited degree. However, these systems are not able to capture broader narrative or structural goals in the poetry generation process. As such, computer-generated poetry lacks many of the technical considerations that poets use in writing poetry.

We want to create a system that does impose further structural restrictions on poem generation: rather than generating a poem based on one topic, we want to generate poems that include multiple related topics.

We approach poetry generation as a narrative task to explore how using narrative schemata to generate poetry changes the coherence of a poem. The Hafez system that we are extending takes in a user-specified topic as an input and produces a poem as an output. Rather than creating sonnets that's topical to a single word or phrase, Shiraz uses a narrative chain as described by (Chambers and Jurafsky, 2008).

The core architecture of the poem generation project is identical to the Hafez system. However, rather than using user input, we use narrative chains. While we use a pretrained database of narrative trains, in a fully end-to-end implementation of Shiraz, the sonnet generation process would be completed unsupervised.

2.2 Poem Structure Constraints

Shakespearean sonnets are 14-line poems following an ABAB CDCD EFEF GG rhyme scheme, with each line composed of 10 syllables of alternating stress, and the fourteen lines separated into four stanzas: three four-line stanzas and a matching couplet.

Then, with 0 representing an unstressed syllable and a 1 representing a stressed syllable, a Shakespearean sonnet can be characterized as being of the form $((01)^5)^{14}$.

The Hafez system imposes the constraint that poems should use words related to the input topic: relatedness of words is calculated using a Word2Vec model trained on song lyrics, which we also use for our project. When constructing the final poem, Hafez tries to maximize the number of topical words included in the output sonnet.

For the purposes of generating narrative poetry, we add an addition constraint to the sonnet structure: sonnets produced by the system must have a different topic per stanza, i.e. four stanzas per sonnet.

2.3 Selecting Poem Topics

A narrative chain is ”a set of verbs representing events” expressing the role and actions of the protagonist of a story (Chambers and Jurafsky, 2008). The verbs (links) in the chain are ordered temporally, so each consecutive link is a significant plot point occurring one after the other with respect to the protagonist.

Narrative chains are drawn from narrative schemata. A narrative schema is a set of verbs that often share the same subjects and objects in succession within the same document (Chambers and Jurafsky, 2009).

For the purposes of our project, we use pre-trained narrative schema data created in (Chambers and Jurafsky, 2010). The data comes in two forms: a list of counts of how often verbs follow each other in a narrative chain, as well as a list of schema words as well as other highly correlated verbs.

By using the list of counts to reorder schema verbs chronologically, we form narrative chains.

2.4 Algorithm Overview

We find constructing narrative chains of length 4 given schema data. After generating narrative chains, we use the Hafez architecture to generate four distinct stanzas, one each using the which we concatenate in order to form the output sonnet.

We chose to generate stanzas independent of one another so that each sonnet would be focus on its given topic. First, we will discuss the narrative chain creation pipeline, then move into discussion of the Hafez pipeline.

2.5 Narrative Chain Generation

Given a narrative schema database file, we first extract all narrative chains, taking only the chain of verbs, leaving behind role information. Afterwards, we load in verb temporal ordering data (for events A and B, $count(A, B)$ is the number of times A happens before B in real time). This data is produced by the Timebank-trained temporal ordering system of Chambers et al. For each four-word schema, we observe contiguous words w_0 and w_1. If $count(B, A)$ is larger than $count(A, B)$, then swap the two verbs. This applies a temporal ordering to the words that, while not perfect, will hopefully improve the appearance of natural ordering that we hope to achieve to improve the system.

2.6 Hafez Pipeline

The system's pipeline is given an input topic word is, in summary:

1. Identify top k similar words to a given topic using a word2vec model. When the topic is multi-word, Hafez tries to find a phrase for it. If it cannot find a phrase, it treats the topic as a bag of words and just tries to find the words most similar to that bag of words.

2. Select rhyme pairs within the 1000 words using word2vec, favoring words with high cosine similarity. These words will serve as the right-most words within each line.

   The word2vec model was trained on Wikipedia, song lyrics, and the Gigaword. We used the resulting model for our project.

3. For each line-ending word, generate all legal candidate sentences ending in the word from right to left (from the end of a line to the start). Construct a Finite State Acceptor (FSA) that encodes all formally legal poems (that conform to the syllabic constraint).

4. Identify "good" poems using an RNN encoder-decoder language model by running a beam search algorithm. This acts to prune the dense FSA in the process.

Then. the system can be characterized as a two-step process: creating the rhymes and an FSA encoding all legal poems (fitting within the form of a sonnet), then finding the "best" poem by pruning the FSA using an encoder-decoder model.

For the purposes of the beam search algorithm, we set the beam size to be 30 words for computational tractability.

2.6.1 FSA Representation

The FSA has nodes labeled by Ln-Sk, representing the poem at syllable k at line n, forming a syllable-by-syllable mesh. Words are encoded by arcs that connect syllables that it consumes, (e.g.: "Asparagus" beginning at syllable 0 of line 1 would connect L1-S0 to L1-S3).

3 Experimental Evaluation

3.1 Methodology

The evaluation metrics of the original Hafez paper, while sound in methodology and large in scope, were too limiting to demonstrate the excellence of the system's performance. Preference was the only metric used to compare their system to less-advanced versions of their system. They did demonstrate the improvement that their topical-words-encouragement and encoder-decoder LSTM have on human preference, but they did not go beyond simple percent preference. Our evaluation approach uses this same pairwise preference metric, but it also includes two other pairwise metrics, namely coherence and grammaticality. We also include single poem non-binary metrics (integer values from 1 to 5), namely coherence, grammaticality, and topicality. Finally, we include a simple question ”Was this poem written by a human?” for each poem in every trial. We hoped to gain a more insight into the subtleties of the Hafez system. In order to get a sufficiently large amount of data, we utilized Amazon Mechanical Turk.

Coherence, as we define it and as we explain to the HIT workers, is a metric to describe how connected each line and stanza of a poem is to the next. This is the main metric that our main experimental system (modification of Hafez we call Shiraz) was built to improve. We hypothesized that given a natural ordering of verbs (narrative chains), the coherence between stanzas would improve.

Grammaticality is simple but subjective. This metric is meant to represent the poem’s closeness to proper English grammar. Some would argue that this is not important in poetry, but our system is a small subset of poetry in which that is essential.

Topicality is a little more vague. Preceding every poem in the test is a string of four words separated by underscores. For the Hafez variations, these are the four narrative chain verbs used to source the poems. For the human poems, they are one-word summaries of each stanza. Topicality represents the extent to which the poem stays on topic to these words.

3.1.1 Setup

Our experiment utilized four different sets of twenty poems. Three of these sets were Hafez variations, the fourth was a human corpus.

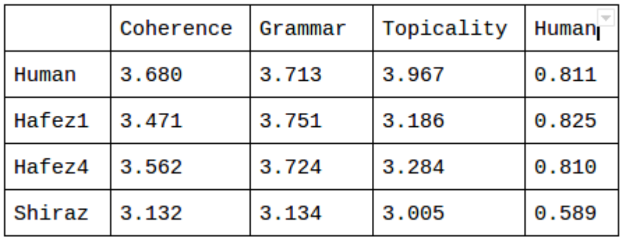
The first of the Hafez variations (Hafez1 or HF1), sources the poem with only the first verb of the ordered narrative chain. The second (Hafez4 or HF4) sources the poem with all four words of the chain. The third (Shiraz or SH) is our main experimental system, described previously (separating poem generation into stanzas). Hafez4 can be thought of as the bag-of-words approach to incorporating the narrative information, while Shiraz brings in word ordering information.

The human corpus consists of twenty human-written poems, eight taken from professional poets, and twelve taken from amateur writers and from English major undergraduates’ publicized works. The professional poems tended to be at least several decades old, whereas the amateur poems were at the latest ten years old. This variance in age was to add variety to the corpus, as well as to represent human poetry at varying skill levels.

All poems sourced by the same chain are paired with themselves. In this group is also a random human poem. In total, this means that there are 120 unique experiment pairs. For robustness, we uploaded 3 duplicates of each experiment to the testing system, for a total of 360. Each HIT worker is randomly assigned one of these experiment pairs. After being assigned, the pair is removed from the list of experiments to run.

3.2 Results

After leaving the batch on MTurk for 1 day, we received 158 validated responses. Below is the single poem data.



	Coherence	Grammar	Topicality	Human
Human	3.680	3.713	3.967	0.811
Hafez1	3.471	3.751	3.186	0.825
Hafez4	3.562	3.724	3.284	0.810
Shiraz	3.132	3.134	3.005	0.589

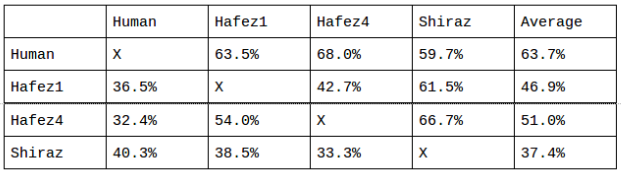
The human poems performed the best across all of these metrics except, surprisingly, humanness. The variance from the other human scores is so low that this does not indicate much about the human corpus. However, it does indicate that the Hafez poems appear just as human as human poems, which is a wonderfully interesting result.

Of the generated poems, Hafez1 was the most human and grammatical, while Hafez4 was the most topical and coherent. This is the first of many negative results for the Shiraz approach - it was significantly lower across the board.

The pairwise data introduced a little more insight into these numbers (Figure 1 below). Each value in the table represents the percentage of poems in that row that were evaluated as more coherent, grammatical, or preferable to the poems in the column.

The general trends started in the single poem metrics continue here, with human poems significantly outperforming the generated poems. Hafez1 and Hafez4, however, show different trends. Hafez1 has the highest pairwise coherence percentage, while Hafez4 has the highest pairwise preference and grammaticality scores. It is interesting to point out the head-to-heads that don't match the average: Hafez1 is on average more pairwise coherent than Hafez4, but in their head-to-head, Hafez4 had the edge with a 51.7 percent coherence.

The next table averages across all three metrics:



	Human	Hafez1	Hafez4	Shiraz	Average
Human	X	63.5%	68.0%	59.7%	63.7%
Hafez1	36.5%	X	42.7%	61.5%	46.9%
Hafez4	32.4%	54.0%	X	66.7%	51.0%
Shiraz	40.3%	38.5%	33.3%	X	37.4%

Each cell now shows the average performance of the row poems versus the column poems. The final column can be used as the most authoritative ordering of performance (Human, Hafez4, Hafez1, Shiraz). However, Shiraz is the best at one thing: competing against the human poems. Its average performance against the human corpus of 40.3 percent is significantly higher than that of Hafez1 and Hafez4. This is puzzling, considering the overwhelming evidence against the success of our approach. However, we think that it indicates some unique, subtle success of incorporating narrative information.

3.3 Discussion

This experiment is limited, in part by the small (138 responses) MTurk survey size, but also because of the variety of training data. However, clear trends emerged through the noise of this system, which we think means the data is sound and can be reasoned upon. Therefore, we can draw the conclusion that our current approach to incorporating narrative information into poetry generation is not successful.

There are many reasons that this happened. One of the biggest is the large neural network so crucial to the Hafez system. A neural approach to modeling generally implies that features are to be learned by the system, and not imposed or classified by the implementer. Perhaps forcing this narrative constraint restricted the well-trained model too much, causing our results to suffer.

Another reason is that we do not incorporate enough narrative information. We do not touch the roles information supplied alongside the verbs in the Chambers schema database. These roles (eg. Congress) are often more unique words than the schema verbs, and thus could lead to more unique and accurate poem generation. Then again, this could also further restrict the neural model. We could also incorporate the narrative information more precisely (see Future Work).

We believe that the success of the Shiraz system directly against human poetry shows that our idea can lead to some success, and that our implementation is at fault.

3.4 Related Work

3.4.1 Human Interactive Hafez

Rather than going in an unsupervised direction, a more interactive version of Hafez (Interactive Hafez) has been created (Ghazvininejad et al.,). This version of Hafez tries to optimize a poem by including a human in the loop and iteratively improve a generated poem.

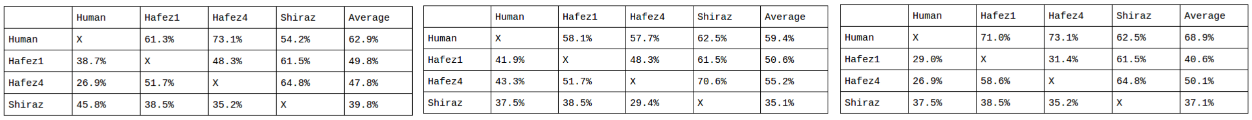
Interactive Hafez first prompts a human user for a topic, then generates a topical poem using the pipeline as described above. After the system outputs a poem, a user has the option to tune the poem, such as by encouraging the system to use certain words (thus reweighting certain words when doing the encoder-decoder step), requesting more/less repetition, a certain average word length, among other parameters.

The system can then learn to use these preferences on other runs. This differs from Shiraz, which creates topics in a completely unsupervised manner.

4 Future Work

4.0.1 Independent Stanza Generation

Shiraz creates poetry in a stanza-by-stanza basis, treating each stanza as independent of the next. We chose this approach because it ensures that each stanza is focused on a particular topic. We hypothesized that the relatedness of topic words would allow Shiraz to create relatively coherent poetry. However,



	Human	Hafez1	Hafez4	Shiraz	Average
Human	X	61.3%	73.1%	54.2%	62.9%
Hafez1	38.7%	X	48.3%	61.5%	49.8%
Hafez4	26.9%	51.7%	X	64.8%	47.8%
Shiraz	45.8%	38.5%	35.2%	X	39.8%

	Human	Hafez1	Hafez4	Shiraz	Average
Human	X	58.1%	57.7%	62.5%	59.4%
Hafez1	41.9%	X	48.3%	61.5%	50.6%
Hafez4	43.3%	51.7%	X	70.6%	55.2%
Shiraz	37.5%	38.5%	29.4%	X	35.1%

	Human	Hafez1	Hafez4	Shiraz	Average
Human	X	71.0%	73.1%	62.5%	68.9%
Hafez1	29.0%	X	31.4%	61.5%	40.6%
Hafez4	26.9%	58.6%	X	64.8%	50.1%
Shiraz	37.5%	38.5%	35.2%	X	37.1%

Figure 1: Pairwise experimental data for each metric

as Hafez1 and Hafez4 outperformed Shiraz in testing it may be worth generating the entire poem using one encoder-decoder model.

4.0.2 Choice of Model/Poem Type

Sonnets are short, compact poems, and are thus not optimized towards conveying narrative. In order to create more narratively-focused poems, we could explore training on epic poetry and narrative songs instead of the more general song corpus used for this project. We could also aim to generate epic poems in blank verse or a series of limericks or other shorter poems rather than trying to condense an entire narrative chain into a 14-line sonnet.

4.0.3 Encoding Narrative Information

At the end of the day, Shiraz chooses words to use based on similarity without a notion of narrative progression. Somehow encoding a notion of narrative (such as the narrative schema/chain models as described by the Jurafsky papers) directly into the FSA generation may lead to more robust poetry. Further, it would be possible to first generate a prose story, then translate said story into poetry. We could also train entirely different language models.

5 Conclusion

Shiraz is a poetry-generation system that generates its stanzas independent of one another, with each stanza relating to a separate topic. However, this approach did not effectively generate poetry, failing to outperform Hafez, the base system Shiraz was built from.

From our results, we can see that the Hafez baseline outperformed Shiraz despite how it handles multi-word topics as a bag of words. This may be indicative of the fact that Shiraz was ill-suited for sonnet generation.

References

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797. Citeseer.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 602–610. Association for Computational Linguistics.

Nathanael Chambers and Daniel Jurafsky. 2010. A database of narrative schemas. In *LREC*.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. Hafez: an interactive poetry generation system.

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191.