Generating Lyrics using Constrained Random Walks on a Word Network

Žiga Babnik, Jasmina Pegan, Domen Kos and Lovro Šubelj
Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia
E-mail: zb1996@student.uni-lj.si, jp2634@student.uni-lj.si, dk6314@student.uni-lj.si, lovro.subelj@fri.uni-lj.si

Student paper

Keywords: lyrics, lyrics generation, Markov models, networks, network analysis, poetry generation, semantics

Received: November 16, 2020

In the paper we present an approach for automatic lyrics generation. From the American National Corpus of written texts we build a Word Network, which encodes word sequences. Lyrics are then generated by performing a constrained random walk over the Word Network. *The constraints include the structure of the generated sentence, the rhythm of the lines of the stanza or the rhymes of the stanza itself. Lyrics are generated using each constraint individually and also using all three constraints at the same time. We tested the single constraint strategies using a toy example, while the results of the joint strategy were subject to human review. While the given properties of the toy example, were kept in the results, replicating the toy example perfectly proved a difficult task. The results of the questionnaire showed that lack of a deeper meaning and strange capitalization were the main reasons that our results did not appear as though they were written by a human.*

Povzetek: Avtomatsko generiranje besedil pesmi bazira na uporabi omejenih naključnih sprehodov po besednem omrežju Word Network, vzpostavljenem iz Ameriškega nacionalnega korpusa. Besedila se generirajo z upoštevanjem strukture stavka, ritma in rim.

1 Introduction

Natural Language Processing (NLP) is becoming a very popular research field, with many researchers working on it. Many methods for speech recognition, understanding of language and generation of text are being developed. In this paper we concentrate on the subtask of NLP which is text generation. More specifically we address the problem of generating lyrics that resemble real lyrics in some way.

Since the development of deep neural networks most of the state of the art approaches for text generation are based on extracting features with deep learning. Our approach is to use NLP tools and methods to extract them manually and pack them into one or more networks, all containing some information about real lyrics. The main idea is to use a big data set of existing songs and possibly other texts and construct the needed networks out of them. With some constrained random walk through this networks we then generate lyrics for new songs. The nodes in the main network, we call the *Word Network*, are words and the edges are relations between them. We focus on building many strategies where each strategy ensures one property of real texts is satisfied, such as rhythm, rhymes and sentence structure, all of which play a major role in lyrics. By combining these individual strategies we want to create a system that generates lyrics which mimic real lyrics in many different aspects.

In section 2 we overview papers relevant to our research. Firstly we present an overview of the field in section 2.1, after which we take a closer look at three most relevant papers. In section 2.2 we present a paper [9] that introduces the *PoeTryMe* poetry generation system, in section 2.3 we present the paper [8] that introduces the *Tra-la-Lyrics* song lyrics generation system and in section 2.4 we present a paper [6] that introduces a Markov Constraint based system for lyrics generation.

In section 3 we present the data used to build the necessary networks and generate new lyrics as well as the *Word Network*, which is the central data structure of our system.

In section 4 we present our general approach for the implementation of a lyrics generation system. Firstly in section 4.1 we present how the lyrics structure was generated, which is then used in different text generation methods presented in section 4.2. Finally in section 4.3 we present how the generated text is reorganized using the structural information to produce the final generated lyrics.

In section 5 we present the results of the different generation strategies. Firstly in section 5.1 we present an example of generated lyrics for each of the developed strategies. In section 5.2 evaluation of the results using a toy example is presented, finally in section 5.3 evaluation using public review is presented. In section 6 we present the main results of our paper as well as propose our interpretation of them. Finally in section 7 we overview the paper.

2 Related work

2.1 A Survey on intelligent poetry generation

The authors of the paper *A Survey on Intelligent Poetry Generation: Languages, Features, Techniques, Reutilisation and Evaluation* [7] made an overview of the intelligent generation of poetry area. In the paper they discuss many topics, mainly surrounding different types of poetry, the structure of poems and how to recognise them. They also discuss the most common formulated features and how to design a generator which takes into account the features based on the language of the poem. Another important mention are so called *Content features* which depend on the grammatical correctness and meaningfulness of the text and how to achieve them.

In the second part of the paper they discuss artificial intelligence techniques for poem generation. One of the interesting approaches is to use genetic algorithms where the population is represented with initial drafts and in each iteration the most promising texts are kept. New poem are generated using mutations and crossover operations which are evaluated by some fitness function. Another approach is to present this as a constraint optimization problem where constraints are represented with the number of lines, syllables per line, number of rhymes, etc. The algorithm should generate a poem such that it optimizes those constraints. Standard machine learning methods were also used. A Support Vector Machine (SVM) [11] model was trained on a poetry corpus and used to predict the next word or syllable. They also used language models to generate poetry texts which were represented with Markov models and some Deep Neural Networks (DNNs) which includes Recurrent Neural Networks (RNNs) [10].

In the last part they also discuss the evaluation of such texts where most of the reliable evaluation is still performed by humans. They discuss some metrics which are mostly used for classification of the poem type by measuring occurrence of different properties in a generated poem.

2.2 PoeTryMe

In the paper *PoeTryMe: a versatile platform for poetry generation* [9] an automatized poetry generation system for Portuguese poetry is presented. It uses a set of seed words to describe the general context of the goal lyrics, and a poem template for structure and rhythm. PoeTryMe supports syllable-based rhythm with no regards to stress patterns. Also grammar and word relations represented by relational triples $(node_1, relation\_type, node_2)$ can be user-defined.

The paper categorizes poetry generation techniques into four categories: template-based where a sentence is generated in accordance to the template, generate-and-test where n sentences are generated and the best is chosen, evolutionary where n poems are generated, then the best few are selected and crossed repeatedly and case-based reasoning approach that uses adaptation of existing songs. Implementation of the algorithm PoeTryMe uses three different strategies to generate lines: basic which is categorized as template-based, generate-and-test and an evolutionary approach. The system is modular, it consists of a sentence generator, grammar processor, relations manager, contextualizer, syllables utility, sentiment processor and a generation strategy [8] already described.

Three generated poems are presented as the results. The authors confirm that following multiple properties of poetry such as meaningfulness, grammatical correctness and poeticness at the same time is hard. PoeTryMe generates grammatically correct sentences which are somehow related to a given keywords and at the same time conforming to given structure. Only the evolutionary approach has rhymes with high probability.

2.3 Tra-la-Lyrics 2.0

In the paper *Tra-la-Lyrics 2.0: Automatic Generation of Song Lyrics on a Semantic Domain* [8] a system for automatic generation of lyrics is presented. Tra-la-Lyrics 2.0 generates text with rhymes on a semantic domain with a given rhythm, based on input music. Its predecessor Tra-la-Lyrics generated rhymed rhythmicized text based on stressed syllables with no regards to semantics. The 2.0 version integrates the previous approach with PoeTryMe to achieve generation of meaningful lyrics on a given topic with rhythm and rhymes.

Tra-la-Lyrics has two rhyming strategies: Rhythm+Rhymes (RR) and Generative Grammar (GG). The RR strategy prefers rhymes at specific parts of the song. In addition to that, GG sets morphological constraints. As lyrics are often repetitive, both strategies also include a repetition parameter.

The implementation of Tra-la-Lyrics 2.0 was derived from PoeTryMe by changing the algorithm to accept a song as an input and by creating a new generation strategy which considers also the rhythm.

Results are again presented in the form of generated lyrics. The results of Tra-la-Lyrics and Tra-la-Lyrics 2.0 are evaluated empirically and numerically on a number of points such as rhythm, rhymes, semantics and meaningfulness. On the average Tra-la-Lyrics 2.0 outperforms its precedent, but although it shows improvement in meaningfulness, it is still far from perfect.

2.4 Markov constraints for generating lyrics

In the paper [6] the authors used Markov models to generate lyrics in the style of existing authors. Since the Markov chains are not suitable to satisfy the non-local properties of poems such as structural constraints, the authors developed a more advanced framework. Using so called Constrained Markov Processes (CMP) they generated texts that were consistent with the corpus. The idea is to represent the prob-

lem as the constraint satisfaction problem. A Markov probabilistic model is then built in two steps. They presented two different constraints. The first one is replacing the transition probability in the standard Markov model. It is called *Markov constraint* and beside the transition probability also holds a constraint variable. The other type is called *Control constraint* which needs to be satisfied in some specific state. The *Markov constraints* on each transition are then set so that they satisfy *Control constraint*. Using these techniques they were able to keep structural properties of the poems such as rhyme and rhythm.

They also demonstrated the methods and evaluate them. The evaluation was again performed manually by 12 volunteers.

3 Data

Our approach is based on a constrained random walk over the so-called *Word Network*. We have to ensure that the *Word Network* is large enough, so that we will be able to perform the constrained random walks on it. In order for the *Word Network* to be large enough it needs to be constructed from a large data set, we chose the *Open American National Corpus* data set [3] which contains over 6000 texts from different domains, totaling around 11 million words.

Since our approach tries to generate text that mimics lyrics by some property, we also need a data set that includes lyrics, from which we will be able to extract these properties. We chose the *Song Lyrics* data set [5] available on the Kaggle platform. The data set includes lyrics from 49 different artists such as Adele, The Beatles, Bob Marley and countless others, gained from free online lyrics hosting websites using a Python script. For each artist a single text file is available that contains lyrics from several songs of the artist. Since the data structures built from this data set are specific to certain sub-tasks of our approach, we will introduce them later on.

3.0.1 Word network

The *Word Network* is a directed network and represents the dependencies between single words in the lyrics, the nodes in the network represent individual words, while the links show if two words appear in the lyrics one after another. To build such a network we first tokenize each sentence of the texts. We than construct a list of all word tuples, such that the first word in the tuple is always followed by the second word in the tuple in the lyrics. To build the network we than iterate over all such tuples adding individual words as nodes in the network, where each word node gets the following attributes: the Part-of-Speech tag (POS tag) of the word and a list of all possible phonemes of the word. After adding both words from the tuple into the network we than do the following, if a link already exists between the words in the network we increase the weight of the link by one, on the other hand if a link does not exist we simply add it with weight equal to one.

Table 1 presents some basic statistics of the *Word Network*, while Figure 1 presents the indegree and outdegree distributions of the network.

Statistic	Result
Number of nodes (n)	60115
Number of links (m)	2357451
Average degree ($\overline{k}$)	39
Density (ρ)	0.00065
Number of nodes in LCC	60111
Average clustering coefficient ($\overline{C}$)	0.467

Table 1: Basic statistics of the *Word Network*

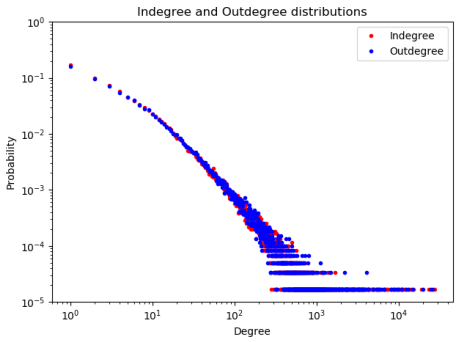

Figure 1: Indegree and outdegree distributions

From the statistics we see that the *Word Network* is quite dense, which is a desirable property for our approach. To calculate the number of nodes in the largest connected component, we first turned the *Word Network* into an undirected network, we see that most of the words are within the largest connected component. Finally we see that its degree distributions roughly follow a power-law distribution.

4 Methods

Our approach consists of three stages. In the first stage the general structure of the lyrics is generated, here we obtain the following: how different stanzas such as the chorus and verse follow each other and also how many lines are contained in each of them. This information is fed to the second stage which generates lines for each stanza in the lyrics structure. The third stage than collects the lines and stacks them according to the lyrics structure, while also adding details such as capitalization and commas. Figure 2 shows visually how our approach is structured at the highest level.

4.1 Generating lyrics structure

The approach used to generate lyrics structure uses a simple network called the *Structure Network*. The *Structure Network* is a directed network, which contains the four most basic blocks of lyrics: *intro*, *verse*, *chorus* and *bridge*. Figure 3 shows how these nodes are connected and the

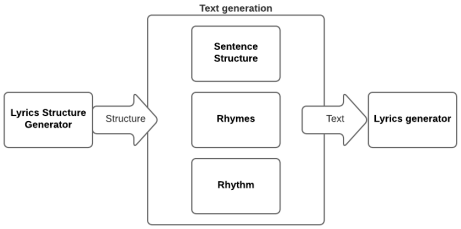

Figure 2: Visualization of approach pipeline

weights of each connection. The *Structure Network* was handcrafted and represents only a rough approximation of how a song can be structured.

To generate the lyrics structure a random walk starting from the *intro* node was performed. The walk was stopped once more than five steps were performed and the current observed node was not *verse*, meaning we did not want our lyrics to end with a *verse*.

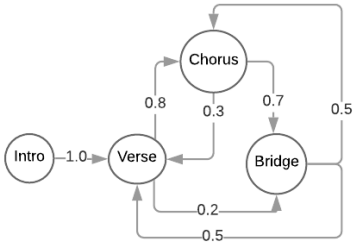

Figure 3: Visualization of the Structure Network

After the lyrics structure was generated we also generated the number of lines each part should contain. This was done simply by randomly selecting a number in a given interval. The interval was defined as $[3, 6]$ for *verse*, *chorus* and *bridge*, while for *intro* it was defined as $[2, 4]$.

4.2 Generating text with certain properties

In the following section we present approaches for generating texts with certain properties. These properties include proper sentence structure, rhymes and rhythm. We introduce strategies for generating lyrics that take only one of these properties into consideration and also a joint strategy which takes all three into consideration.

4.2.1 Generating text with proper sentence structure

We propose a model that takes into account the sentence structure present in individual lines of lyrics. Alongside the *Word Network* this strategy also uses the *Part-of-speech tag network* or *POS-tag network*.

The *POS-tag network* is a directed network that contains information about the sequences of line structures we can observe in lyrics. Firstly each line in the lyrics is represented as a sequence of Part-of-Speech (POS) tags, that tell us the structure of the line. The POS tag sequences of lines are then added to the network as nodes, we create a link between POS tag sequences X and Y, if it holds that in the lyrics POS tag sequence Y comes directly after POS tag sequence X. By performing a random walk over such a network we not only guarantee the proper structure of each individual line, but also proper ordering of lines.

To generate text for a given part, we first generate a sequence of line structures using the *POS-tag network*. This is done using simple random walks over the network, where the number of steps equals the length of the given part. The walk generates all the needed constraints for this strategy.

Once the constraint in the form of POS-tag structure of individual lines has been generated, we perform a constrained random walk on the *Word Network*. For each line a seperate constrained random walk is performed. The first word of a line is chosen randomly among all words with the proper POS-tag, for each successor we search among all neighbors of the current word, that again have the proper POS-tag. If such a neighbor does not exist, we start the walk for the current line from the beginning.

4.2.2 Generating text based on rhyme scheme

We first defined three types of rhymes. Since the *Word Network* is constructed from random texts and not lyrics we do not expect to find many neighbourhood words that correspond to a perfect rhyme. That is why we allow three types of rhymes. The first one is a perfect rhyme which is defined as the rhyme where the stressed vowels and any succeeding consonants are identical e.g. *believe* and *conceive* [4]. The second rhyme is called assonance or a vowel rhyme. It is a rhyme in which the same vowel sounds are used with different consonants in the stressed syllables of the rhyming words e.g. *pentient* and *reticence* [1]. The third rhyme is called consonant rhyme and is the repetition of consonants or consonant patterns especially at the ends of words e.g. *bell* and *ball* [2].

Our strategy generates words in such order that they follow the rhyme scheme we chose. By defining the number of words in a line and the rhyme scheme, e.g ”ABBA”, we then generate our lyrics by randomly choosing the node in the *Word Network*. As in a random walk we chose a successor by taking into account the weights of each edge. After reaching the last word in the line we chose the next node only from the successor that do not violate the rhyme scheme. In this step we chose the successor both uniformly at random and by weighting each rhyme. Most of the stop words have high weights in the *Word Network* and they are usually short and by definition most of them rhyme. That is why they were chosen in most of the cases. The strategy generated more natural rhymes when choosing them randomly.

4.2.3 Generating text with rhythm

We propose a model that generates lyrics based on a given rhythm. The model uses the *Word Network* and an additional network storing rhythm data. The *Rhythm network* is

a weighted directed network that represents rhythms found in the lyrics. It consists of three nodes: -1 represents start or ending of a line, 0 stands for an unstressed syllable and 1 indicates a stressed syllable. The weights of edges were decided by calculating the normalized number of transitions between the corresponding syllables. We can use this network to generate a random rhythm by starting at node -1 and choosing a random neighbor taking into account corresponding edge weights. When we reach -1 again, the generated line of rhythm is completed. The generated line of rhythm is then corrected to include more repetitiveness which is expected from rhythm to feel natural. The first n syllable stresses are chosen as baseline rhythm and are then propagated throughout the rhythm line with 70% probability. Number of syllables n is a randomly chosen number between 2 and 4.

The rhythm-based model has two variants: one is given a rhythm and the other generates the rhythm for each verse. Each variant has two sub-variants, one uses random walk strategy and the other uses POS tag strategy.

First, the rhythm is acquired in form of a string where 0 stands for an unstressed syllable and 1 for a stressed syllable. If the rhythm is given, it is expanded to match the line length. Otherwise, a random line rhythm is generated from the rhythm network for each verse of generated text. We start with a random node from the *Word Network* or the *POS-tag network*, depending on the variant, and expand the line with successors that match the required rhythm.

4.2.4 Generating text with multiple constraints

Our last strategy combined all the above constraints: structure, rhythm and rhyme. Each line was generated such that it took into account the structure, rhythm and the rhyme. Although the network is relatively large, it sometimes happened that none of the successors would satisfy all the constraints. In that case we performed a random jump and started generating the current line again. This was repeated until the generated text satisfied all the constraints.

4.3 Constructing lyrics from generated text

In the final phase of our approach we combine the generated text, with the structure of the lyrics to produce proper lyrics. Firstly we reorder the text according to the lyrics structure, so that the parts properly follow each other. Each part also gets an annotator in the form of `[<part name>]`. The ordered and annotated text is capitalized using a simple POS-tag heuristic, where we simply check the tag of each word. If the tag of the word is `NN` or `NNS`, or if the word is the first in line, we capitalize it. Commas are also added to each line, as well as a line separator between each part.

The text generators described previously generates text for each unique part only once, meaning that if the lyrics contain for instance more than one *chorus* all would contain the same text. To avoid exact repetition five to ten words in each part were chosen at random to be replaced. The whole text of the part along with the list of words to replace were then sent back to the generator. The generator then replaced the words according to the same constraints the text was generated in the beginning.

5 Results

In the following section we present the results of each strategy described in section 4.2. For each strategy we present lyrics generated by that strategy.

In order to evaluate how well the strategies work, we devised two different evaluation approaches. The first approach using a toy example was applied to all strategies where only one property of lyrics was being considered, the second approach using public review was applied to the strategy which considered all three properties.

5.1 Lyrics generated by strategies

In the following section we present lyrics generated by each of the strategies developed.

5.1.1 Lyrics generated by the sentence structure strategy

The lyrics generated based on the sentence structure strategy are presented below.

[intro]
Benjamin upward Half as described a Variety,
Carriages looser Sugar as,
Nov Angstroms o Ja Et Banditry,
Newsgroups has bacall Whitney.

[verse]
Okay like newsweek explain what you awoke,
Handhold Inspire Confidence Nudge between Gore Camp s play it s,
Think I this Observation shows helplessly he co their Interfaces,
Mine Everybody harsher Children,
Professor tour Hitler S to have,
Concludes it s my E.

[chorus]
Tina would have their Security Capital Sneeze Bob Bookies won T,
Hottest Tech Approach for nettlesome Human,
Versa Whereas outfit Blame for directional the Apparatus,
Lorge Jr Call Attention Let s h,
Craigie the Membrane regular blazes the Funding Mix.

[bridge]
Postage in Yakovlev for Discounting,
Sloth less stringent E ether Inhibitory,
Boris a Tampax described i attend academic Literature Briefly back mr Morris I think they.

[verse] Method like newsweek explain what you awoke,
Handhold Inspire Confidence Ricky between Gore Camp s play it s,
Think I this Detainee chides liter he co their Interfaces,
Mine Encyclopaedia Hanover Campsites,
Professor tour Hitler S to have,
Concludes it arsenal my E.

[chorus]
Tina could have their Security oilseed Danforth Bob Bookies won T,
Freshener tech Airport for nettlesome Human,
Teacher whereas Outfit Blame for directional Argentina Apparatus,
Strove Jr Call Attention Let s h,
Craigie the Neville Sari blazes the Funding Mix.

While the generated sentences might follow the correct POS tag structure it is clear that this constraint is not strong

enough to enable the generation of lyrics that would to some extent resemble real lyrics. Generation of meaningful lyrics is not one of our main goals, but rather that individual lines and smaller building blocks could resemble real parts of lyrics. This generation strategy is not informative enough to enable the generation of such lyrics.

5.1.2 Lyrics generated by the strategy based on the rhyme scheme

In the example below we can observe the lyrics generated by our strategy with a predefined rhyme scheme which is ”ABA” for *intro*, *verse* and *chorus*, while for the *bridge* ”ABABAB” was used.

[intro]
Shrug Democrats have and when in,
Voided the strengthening their biggest Risk,
Tetrads were controlled Trials since when.

[verse]
Plumage of ethical Story,
Stoneburner and the to,
Somehow forced the Study.

[bridge]
Randy s no Consensus that federal Reserve,
Lighting Shows which the Elk they be,
Summits to any other Poet who was,
Coelho but if you find Anybody who,
Dystrophy Pages for personal favorite Newspaper but,
Retried unpopular Gingrich in faces some Corroboration.

[chorus]
Lousiness impregnable the Incorrect,
Kolb and other racial,
Berri for Performance Checked.

[bridge]
Randy s no Consensus that federal Reserve,
Lighting Shows which the Elk they be,
Summits to all other Poet who preserve,
Coelho if you find Anybody than me,
Dystrophy Pages for personal favorite Newspaper Reserve,
Retried unpopular Gingrich in faces some be.

We can observe that some rhymes are more natural like ”be-me” and ”preserve-reserve”, while the others satisfy the definition but are not so natural to read, e.g. ”in-when”.

5.1.3 Lyrics generated by the rhythm-based strategy

We tested our rhythm-based strategy. First we look at the variant of the algorithm which accepts rhythm as an input. When given the rhythm ”011” of Prešeren’s *Povodni mož*, the random walk sub-version returns the following example.

[intro]
Duked out on the pre Birth Control of a Round Golf,
To find a Campaign seems and a Union Rules in,
The Lung lymphocytes a Tractor it is so says,
The good Seafood in S a new Car are not the.

[verse]
Kasparov was Things on the Times you about the,
Subject Matter when just Delights in e big Hot,
And more Years make are Fans will find both good or wade.

[chorus]
Embodiment of Error R a relaxed Dole,
Campaign had to turn out the Sum up to be those,
Of south will its Roots to its Entries and but and,
A M Rosenthal who was true but impressed by,
The Bush s alleged new Account for the six Lines,
Though a new Environment Act at the Field flat.

[bridge]
Nationals leaving its Building overlooks this,
Has the brooklyn Heights or a large Part of Spread his,
It a Ratio of the new Products that we,
Punished the Schedule slipped across Species female,
In the black and it a Policewoman who did,
Seduce Frank in one Year Alumni and Truck on.

[verse]
Southerner does Things on the Times you intend the,
Subject about when that delights in Take big Love,
And more Years make are Goals will find both good or wade.

[chorus]
Embodiment of Error R to whether Dole,
Campaign had to turn out the Sum up may be those,
Of Range will its Roots to Time Limit and but and,
A M Rosenthal who was true but impressed by,
The Bush his alleged new Account for the six Sets,
Though find new Environment Act at the Field flat.

The text can definitely be read in the given rhythm, but some words are accented in an unusual way. Some words even change their meaning by being differently accented, ex. ”subjécts” - verb vs. ”súbjects” - noun. The POS-tag sub-version returns similar results.

We also tested the version of the algorithm that generates the rhythm from the *Rhythm network*. We limited the number of syllables per line to no less than 3 and no more than 12. The random walk sub-version returns the following example.

[intro]
Peaking they the Engagement,
Such the regional Train in,
Same Fighters that the Cycle,
Genes the Terms the Reporting.

[verse]
Bard on R S back up quick Rise,
Up an of the Mars is deemed key,
Sounds and seek to such as with a.

[chorus]
Bags of Store for a Print Ad the Need for the far more than for,
Non u s next such as shown were the right Heart Rate Hike to the,
Drug Use thanks the Half of slate S Trees that was you help be blurred.

[verse]
Oath on R S back up quick Rise,
Up an Ed the Chain were deemed key,
Al and Seek to such as with a.

[chorus]
Bags of Store must a Sand Ad we need for the far more than for,
Non u S of such as shown were cool right Heart Rate Hike to the,
Drug Use thanks the black of slate S Trees that was you help be blurred.

[bridge]
Au Buisson de mi usa and was low in Kosovo,
And Critics were white House from the State Bar is not a Detour,
To win and leaf of the Floor to see a four Years the Excess,
Hair were sized to what this Test Strips of the n four the Upswing,
Is the most to the Inn Chain had not flinch the Bruce Townsend had.

This nicely emulates the fact that song lyrics are not fixed in structure. But although the line lengths are more relaxed, the rhythm does not flow as we would expect. There is no obvious differences between these results and those of POS-tag sub-version.

5.1.4 Lyrics generated by the combined strategy

Our last strategy combined all previous constraints: sentence structure, rhyme and rhythm. Lyrics generated by this strategy are presented below.

[intro]
Does the Coast a Veto ugh the rayed Moth Mouse Rat,
Hora Asthma have little of it demand Side,
If beck a Rosewood the Kiryat to upgrade my,
Exam on Police then has an Iron Mask.

[verse]
Thumbs Share the Extent other sift the Koran Cliff,
In their Favor Growth even sulk private all Non,
Depressed Stock and Health all this was each Survey consists,
Response Part of where the Pursuit getting the bronzed,
Effects of the Stock Linda this nazis Tees in,
Its fred O and we have a low Signal Fig Wasps.

[chorus]
Its Thank the Bloomberg Years of a Story one Wife,
And knew that if all Lawyer now a Decade long,
And glass the perfect Weather and his Approach Try,
His Rome Site the Role of no Terms meant a Boat was,
Both Rest Till de la vita with Risk of the Iles,
Despite Market to the Success of his Bork was.

[bridge]
It and Symbols bose boris worm Culture through Feed,
A Means of wednesday Gen we obtain a,
Rigdon over in Cris the Midline us is sir,
The real Time of the Gene are shown gown within Reach.

[verse]
Thumbs Share the Extent other fro the Koran Cliff,
In their Janeway Growth even sulk private all Non,
Shuffled Stock and Health all this was each Survey consists,
Response Part of where the Panzer getting those run,
Effects of the Stock Linda this nazis Tees in,
Its fred O and we have a low Proscribes Fig Wasps.

[chorus]
Them thank the bloomberg Years of these Story one Wife,
And knew that if all Lawyer now the Decade long,
And glass the perfect Weather and his Approach Enough,
Me qualms soo the Role of no Terms meant a Boat was,
Both Rest Till de lymph vita with Risk of the Prescribed,
Despite Market to the Success of psi Ip was.

It was quite difficult to satisfy all the constraints and thus some of the lines are a bit strange. Even before joining the strategies it was hard to find a perfect rhyme on a given *Word Network*. With additional constraints we eliminated even more of the possible candidates so the rhymes are mostly combined from stop words and common words.

5.2 Evaluation using a toy example

In the following section we present the results of evaluating the three single constrained strategies using a toy example. For the actual example we chose the following short stanza.

The itsy bitsy spider crawled up the water spout.
Down came the rain, and washed the spider out.
Out came the sun, and dried up all the rain,
and the itsy bitsy spider went up the spout again.

First the *Toy-Word Network* was built from the toy example, later on the POS-tag sentence structure, rhyming and rhythmic scheme were all extracted from each line in the toy example. The extracted properties were than used as constraints in each individual strategy to perform the constrained random walk over the *Toy-Word Network*.

5.2.1 Results of sentence structure strategy

The stanza generated using the sentence structure strategy is presented below.

The Itsy Bitsy Spider crawled up the Water Spout,
Down came the Rain and washed the Spider out,
Out came the Rain and dried up all the Sun,
And the Itsy Bitsy Spider went up the Spout again.

The resulting stanza is very similar to the original indicating that proper sentence structure, when building from a toy example represent quite a strong constraint. When generating the stanza there is small variation between runs, overall most results produce a stanza differing only in a few words from the original toy example, at times it can also happen that the result perfectly matches the toy example.

5.2.2 Results of the strategy based on rhyme scheme

The stanza generated using the extracted rhyming scheme is presented below.

Washed the Itsy Bitsy Spider went up all the Rain,
Itsy Bitsy Spider crawled up the Rain and Spout again,
Down came the Water Spout again dried up the Itsy,
Rain and washed the Sun and dried up the Bitsy.

Although the generated stanza does not make sense semantically the rhyme is the same as in the original song ”AABB”. The first rhyme is even composed from the same words as in the original one. Multiple experiments were performed and in most cases both rhymes were the same ”Itsy-Bitsy”. This is expected since the strategy also uses weights on the edges, but we can also observe that the strategy reproduces the rhyme of the original stanza.

5.2.3 Results of the rhythm-based strategy

The stanza generated using the extracted rhythm scheme is presented below.

And the Sun and the Itsy Bitsy Spider crawled up,
The Itsy Bitsy Spider out came the,
Sun and the Itsy Bitsy Spider out,
Came the Sun and the Water Spout again came the Spout again.

The text is hard to read in that rhythm, one word is even stressed incorrectly this way (*Bitsý* instead of *Bítsy*). The unreadability is expected as the algorithm uses all the possible word pronunciations and their combinations. The main problem here is that algorithm chooses many monosyllabic words which can be pronounced stressed or unstressed thus fulfilling the rhythm pattern with any combination of such words. In natural speech the text as a whole would be pronounced differently, depending on stresses of nearby words.

5.3 Evaluation using public review

In the following section we present the results of evaluating the strategy that takes into account all three properties of lyrics: proper sentence structure, rhythm and rhyme. In

order to evaluate this strategy we put together a short questionnaire. The questionnaire included three sections, each dedicated to its own generated line or stanza. Each section included two questions: the first question was a linear scale questions where participants had to rate how much they agree with the statement that the given line or stanza was written by a human, the second question simply asked if the participants could briefly explain their choice from the first question. To elaborate on the possible answers of the first question, participants were tasked with submitting a number between 1 and 5, where: 1 meant strongly disagree, 2 meant disagree, 3 meant neither disagree nor agree, 4 meant agree and 5 meant strongly agree. In total 29 people participated in the questionnaire, the results broken down into each section are presented below.

5.3.1 Results for the given line

In the first section, the participants were given the following generated line.

Love Affair issue down from July

Since the strategy using all three constraints does not have any contextual information that it could use when generating different lines for a stanza, it would be natural that people would think that a single line from the lyrics is more likely to be written by a human than a whole stanza. This is why we included this question.

Figure 4 shows the results of the first question for the given line. We can observe that most of the participants disagreed that the generated line was written by human, while a small part were undecided or thought that it is possible that the line was written by a human.

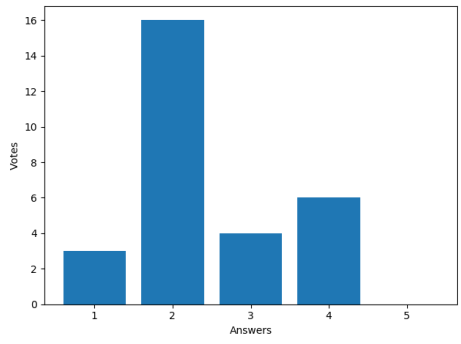

Figure 4: Results of the linear scale question for the given line

When we asked them to reason about their decisions most of them said that they do not believe it was written by a human since the sentence does not have semantic meaning, while some others said that the capitalization of the word *affair* was the main reason for their decision.

5.3.2 Results for the first stanza

In the second section the participants were given the following generated stanza.

You miss Hip for her from,
This Game as a Fast will,
See the most of the Dow.

We included two such stanzas, so that we could make a comparison between the two and possibly nail the reason why one would appear more like it was written by humans.

Figure 5 shows the results of the first question for first given stanza. With a longer text the disagreement that the text was written by human was stronger. Most people either strongly disagreed or disagreed that the stanza was written by humans, with only a handful being undecided or agreeing that the stanza might be written by a human.

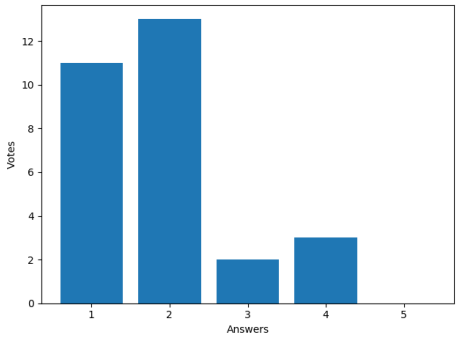

Figure 5: Results of the linear scale question for the first given stanza

The reasoning was similar as before that most of the text does not make any sense. A lot of them were also confused about upper case letters in the middle of the sentence. Some pointed out that the stanza did not have proper rhymes.

5.3.3 Results for the second stanza

In the third section, the participants were given the following generated stanza.

Cleaned up that her,
Mouth while Tag Team,
Though that my Time,
Fig Leaf you see.

Figure 6 shows the results of the first question for the second given stanza. The results of the first question are quite similar to the first given stanza, with even more people strongly disagreeing that the given line was written by a human.

The reasoning for the choices is again that there is no semantic meaning in the lines of the stanza. Some of the participants also pointed out that the rhymes do not look natural, while minority pointed out that it looks more natural than the previous stanza.

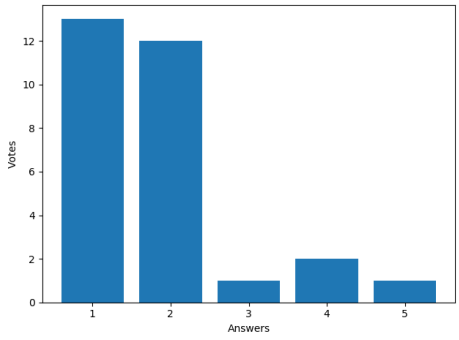

Figure 6: Results of the linear scale question for the second given stanza

5.3.4 Comparison of the two generated stanzas

In the last section we asked people if the first generated stanza replicated real lyrics better than the second stanza. Figure 7 shows results for the given statement.

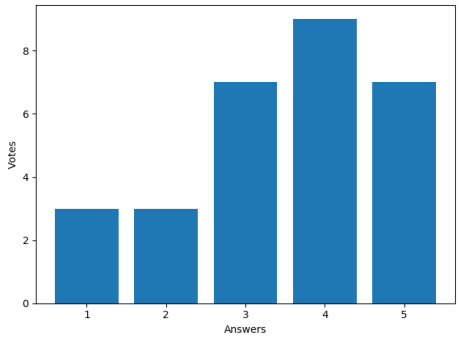

Figure 7: Results of the linear scale question about the comparison of the two generated stanzas

The results clearly show that most people agreed that the first stanza replicated real lyrics better. Trying to compare the answers of the second questions of both the given stanzas, poses a problem. On one hand there is a clear consensus that the first stanza looks more real, while the answers for both seem to indicate the same problems, with meaning, unusual rhymes and capitalization. Between the two stanzas we could not identify the exact reason why one might appear more real than the other. What is clear is that individual lines from the text do appear more natural than both stanzas.

6 Discussion

Looking at the results gained from the toy example, it is clear that replicating one property of the original example does not give enough information to properly reconstruct the toy example. It is clear that using only rhymes as a constraint here produces a result that can be very different from the original as we limit only the end words of each line. It is also clear that using the sentence structure produced the best results, this is probably due to small variability in the POS-tags of the words in the toy example. Leading to the fact that not many constrained random walks on the built *Toy-Word Network* produce the correct extracted sentence structure. Overall, we are satisfied with the results gained from using the toy example, since the properties are replicated perfectly, which is the main goal of these strategies.

The results of the questionnaire are very clear cut, our combined strategy does not produce lines or stanzas that would appear as though they were written by a human. This results does not surprise us much, as the main reason many people named is simply a lack of meaning in the lyrics, a property we did not incorporate into our system. A bit of a surprise is that sometimes people named the lack of proper rhymes to be the main reason for their decisions, most people probably expect rhymes to be clear cut even though many types of rhymes exist. Another reason that people kept mentioning is the capitalization of words, which could be easily fixed by using a more complex deep model for the capitalization of words, since this was not the focus of our research we do not see this as a big problem.

From the results it is clear that convincing people that a single line was written by a human is easier than a whole stanza. The reason for this is probably very simple, as a line is shorter thus making it appear proper and cohesive is much easier than doing the same with a whole stanza, for which we would need long-term word contexts.

The answers show that one of the two given stanzas appeared more like a real stanza, while the comments and ratings of both of them seemed to differ very little. Our argument for this is that poems and lyrics for humans represent much more than text that simply follows some number of properties, it has a deeper meaning that differs for each individual.

7 Conclusion

We proposed several approaches for generating structured lyrics, which imitate some property of real lyrics. The approaches trying to imitate only one property were evaluated using a toy example, while the combined approach was evaluated using a questionnaire, to determine how human-like the generated lyrics were.

Using the sentence structure strategy we performed constrained random walks on the *Word Network*. To create the needed constraints a random walk on the *POS-tag network* was done, creating a sequence of sentence structures. The results showed that while the generated lyrics did follow the proper sentence structure, they did not resemble actual lyrics.

The rhythm-based strategy with given rhythm generates texts that follow the rhythm well. There are some words that sound unusual when stressed that way, but overall the resulting text is quite flowing and readable. The results of the version which generates its own rhythm are more con-

fusing to read as it is not clear what rhythm is used. Although the same rhythm is used for multiple lines and is self-similar within a line, it is not obvious to the reader what is the actual rhythm.

The last strategy generated lyrics according to a predefined rhyme scheme. Since all our lyrics were generated using the same *Word Network* and since it was not constructed on poems we did not expect many perfect rhymes. The results confirmed this belief. We also tried to learn the rhymes from lyrics. Since the lyrics are written in a modern style, e.g. hip hop, rap, which do not have a specific rhyme scheme like for example ballads, this did not work well.

Finally, we combined the three aspects of lyrics generation. Problems arose when we were trying to take into account all of the strategies, as sometimes no successors which fit all the constraints could be found, so we had to dismiss some of the already generated text and retry from some other node. Public evaluation showed that the main reason why our generated lyrics did not seem natural was a lack of deeper meaning, capitalization of words and rhymes that are not always straightforward.

Automatic lyrics generation is a hard problem and there is definitely more work to be done for our methods to produce valuable results. We realized that while our algorithms are able to achieve proper sentence structure, rhyme and rhythm, the resulting lyrics did not fully replicate real lyrics. One improvement that could be done to tackle this issue would be to build an improved data set from which we would build the *Word Network*. Another would be to try to create constraints around the meaning of lyrics so that we would impose not only structural rules into the generated lyrics but also some form of meaning that could be picked up by a human reader.

References

[1] Assonance rhyme definitions. https://www.dictionary.com/browse/assonance. Accessed: 06-06-2020.

[2] Consonant rhyme definitions. https://www.thefreedictionary.com/consonant+rhyme. Accessed: 06-06-2020.

[3] Open ANC – Open American National Corpus. http://www.anc.org/data/oanc/. Accessed: 06-06-2020.

[4] Perfect rhyme definition. https://www.collinsdictionary.com/dictionary/english/perfect-rhyme. Accessed: 06-06-2020.

[5] Song Lyrics – Kaggle. https://www.kaggle.com/paultimothymooney/poetry. Accessed: 08-05-2020.

[6] Barbieri, G. and Pachet, F., Roy, P. and Degli Esposti, M. *Markov Constraints for Generating Lyrics with Style*. In *ECAI* (2021), vol. 242, pp. 115–120. https://doi.org/10.3233/978-1-61499-098-7-115

[7] Oliveira, H. G. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation* (2017), pp. 11–20. https://doi.org/10.18653/v1/W17-3502

[8] Oliveira, H. G. Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. In *Journal of Artificial General Intelligence 6*, 1 (2015), 87–110. https://doi.org/10.1515/jagi-2015-0005

[9] Oliveira, H. G. PoeTryMe: a versatile platform for poetry generation. In *Computational Creativity, Concept Invention, and General Intelligence 1*, (2021), 21.

[10] Rumelhart, D. E. and Hinton, G. E. and Williams, R. J. Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985. http://dx.doi.org/10.1016/B978-1-4832-1446-7.50035-2

[11] Cortes, C. and Vapnik, V. Support-vector networks. In *Machine learning 20*, 3 (1995), 273–297. https://doi.org/10.1007/BF00994018