A Ballad of the Mexicas: Automated Lyrical Narrative Writing

Divya Singh
San Jose State University
San Jose, CA
divya.singh@sjsu.edu

Margareta Ackerman
San Jose State University
San Jose, CA
margareta.ackerman@sjsu.edu

Rafael Pérez y Pérez
Universidad Autónoma Metropolitana
Mexico City, Mexico
rperez@correo.cua.uam.mx

But she fell in love with him
Girl when they feel the same
The princess was in love with the priest
Can't let go and it never goes out

She also abominated what he did
Be the things they said
The princess was shocked by the priest's actions
And though her heart cant take it all happens[1]

Abstract

Recently, computational systems began approaching challenges that were previously considered to lay exclusively in the human creative domain, such as the art of storytelling and lyrics writing. In this paper, we explore combining these two art forms through the automated creation of ballads. We introduce MABLE (MexicA's BaLlad machinE), based on the plot generation system, MEXICA. Integrating both cognitive and statistical models, MABLE is the first computational system to write narrative-based lyrics. User studies demonstrate MABLE's success at creating emotionally-engaging lyrics with coherent plot.

Introduction

The making of a meaningful and engaging plot extends beyond grammatical structure, requiring an understanding of the underlying context as well as complex interactions amongst agents. As such, it is perhaps unsurprising that the automated creation of coherent stories continues to pose a substantial challenge. This is evident in previous work on lyrics and poetry generation, where the selection of related and powerful words can lead to emotional engagement, yet without the integration of a coherent plot. On the other hand, lyrical works written by humans across diverse genres often center around a consistent story.

In this paper, we tackle the integration of automated storytelling with lyrics writing. As such, we concurrently address challenges encountered in both of these domains, aiming for emotionally engaging lyrics endowed with poetic structure, all the while telling a coherent story.

[1]Excerpts from a ballad created by MABLE.

The art of storytelling through song is found across many different musical genres and styles, yet this tradition is perhaps best embodied in the ballad. Originating from medieval French dance songs and later characteristic of poetry and lyrics from the British Isles, the *ballad* came to be known today as a song narrating a story, typically centered around a romantic, sentimental scheme. As such, we have chosen to initiate our exploration into narrative-based lyrics through the automated creation of ballads, creating lyrics that tell stories of love.

Our efforts in automating the writing of ballads lead to a new approach to their creation, which, to the best of our knowledge, has not been previously utilized by either humans or machines. Our method starts off with a complete plot-line, capturing the main elements of the story. The next step is used to endow the story with rhyme and rhythm. Specifically, every other line in our lyrics is devoted to the progression of the narrative, while the remaining phrases aim to transform the narrative into a ballad.

For the first phase, our system MABLE relies on MEXICA (Pérez y Pérez and Sharples 2001), a plot generation machine based on the engagement-reflection model for creative writing. MEXICA produces novel, coherent plots of stories about the Mexicas, an indigenous people of what is today Mexico City.[2]

After generating a plot using MEXICA, the second phase utilizes a statistical model to expand the plot into lyrics. For each line in the plot, the statistical model is used to create a new, poetic phrase that, by rhyming with the original sentence and following its metric structure, leads to seamless integration of the narrative into the ballad. Specially, we utilize a second-order Markov model trained on a corpus of love songs. Then, we select amongst sentences generated using this model through a set of constraints that allow MEXICA's sentences to fit with the new phrases. We believe that the integration of MEXICA's cognitive-model with the statistical model used in the second phrase is one of the most interesting aspects of our approach, and is perhaps central

[2]The first part of this paper's title comes from the common practice of naming ballads in the form "A ballad of the ... "

to our system's ability to convey narratives in a poetic form.

To evaluate MABLE's artifacts, we compare them against poems and lyrics created by previous systems, as well as human-made lyrics. Results point to MABLE's success in creating emotionally engaging lyrics with a coherent storyline.

We begin with a summary of related previous work and a discussion of MEXICA, which is integral to MABLE. Next, we proceed with an in-depth description of our model, following which we present our user study and discuss findings and implications. We also briefly explore the potential of utilization MABLE for the creation of musical pieces by utilizing ALYSIA (Ackerman and Loker 2017), a songwriting system that creates melodies for user-provided lyrics. The paper concludes with avenues of investigation for future work.

Previous work

A wide range of approaches to the creation of lyrics and poetry have been explored, ranging from templates-based to statistical methods. Tra-la lyrics 2.0 (Gonçalo Oliveira 2015) offers a template-based approach for lyrics generation, combining two previous systems, PoeTryMe and Tra-la-Lyrics. In the former system, user-provided keyword are fed into a sentence generator that utilizes templates and a semantic network to create sentences. These are then used to fill in a poem template. Tra-la-Lyrics creates lyrics based on melodies by constructing text where the stresses match the rhythm of the melody. Tra-la lyrics 2.0 combines these two systems by integrating rhythm information from the melodies into PoeTryMe's architecture.

Full-face poetry generation (Colton, Goodwin, and Veale 2012) constructs poems by extracting key phrases from newspaper articles, which are then combined with database of similes to create template-based sentences. These sentences are in turn combined on the basis of considerations such as their rhyming patterns, word similarly, and sentiment. Other work that creates poetry based on user provided text includes (Tobing and Manurung 2015) and (Misztal and Indurkhya 2014).

SMUG (scientific music generator) (Scirea et al. 2015) explores a related direction by creating songs based on academic papers. The idea is to extract key words from the paper and to utilize them by filling one of several pre-defined song structures, which are subsequently embellished with elements such as onomatopoeias.

Systems have also been designed particularly for the creation of rap lyrics, which have a distinct style. (Hieu Nguyen 2009) relies on a data set of hip-hop lyrics, separated into choruses and verses. These corpusi are used to construct two quadgram models, from which sentences are subsequently generated. Finally, the sentences are combined based on common themes and matching number of syllables in rhyming words. Another system for Rap lyrics, DopeLearning by (Malmi et al. 2015), combines sentences from existing human-made lyrics. DopeLearning's lyrics string together phrases selected from a data base of rap lyrics from 104 different artists, relying on a deep neural net and RankSVM to select each subsequent line on the basis of considerations such as semantic similarity and rhyming.

Other related work considers a co-creative approach to lyrics writing. For example, LyriSys (Watanabe et al. 2017) uses a topic transition model, and allows the user to select topics and specify parameters such as the number of lines and syllables per line. The user can then select and edit lines from amongst those suggested by the system.

Closely related to lyrics generation is work on automated poetry creation. For example, (Toivanen et al. 2013) use constraint programming with standard constraint solvers. A static library of constraints is utilized, such as syllables, number of lines, and rhymes. (Ghazvininejad et al. 2016) create topical poetry starting from a user-specified topic word, which is then used to identify 1000 similar words. These words are subsequently grouped in rhyming classes, and rhyming pairs are selected. Next, a final state acceptor is created, where each poem that it could generate satisfies sonnet constraints and the pairs of rhyming words are used to create a rhyming pattern. The paths are then extracted using a Recurrent Neural Network.

Despite extensive work in this area, none of the previous systems aim to automatically create lyrics (or poems) that convey coherent stories, which is our focus here.

MEXICA

MEXICA (Pérez y Pérez and Sharples 2001) (Pérez y Pérez 2015) (Pérez y Pérez 2007) is a system that characterizes a model of creativity in writing that develops short stories about the old inhabitant of what today is México City. It is inspired by the engagement-reflection account of writing as creative design (Sharples 1999).

In MEXICA, a story is defined as a sequence of actions. Each action has an associated set of preconditions and post conditions in terms of emotional links and tensions between characters. For instance, the precondition of the action "the hunter murdered the jaguar knight" might be the emotional link "hunter hates the knight"; the post condition of the action "the princess awarded the eagle knight" might be the emotional link "the knight is very grateful towards the princess".

In MEXICA tensions represent conflicts, e.g. when the life of a character is at risk, when the health of a character is at risk (i.e. when a character is hurt or ill), or when a character is imprisoned. Each time an action is performed, new conflicts may arise. The model also includes what we refer to as inferred tensions, i.e. tensions that are activated automatically when the system detects that: 1) two different characters are in love with a third one (tension due to love competition); 2) when a character has two opposite emotions towards another one (tension due to clashing emotions); 3) and when a character hates another character and both are located in the same position (tension due to potential danger).

In this way, each time MEXICA generates a new story, the system produces a sequence of actions, and information about the emotional relations and conflicts between the characters in the tale. This information is used by MABLE to extract the sentiments in story.

Description of the model

Lyrics rely on structural elements to create motion, the sense of being pushed forward towards the climax. Pat Pattison, College of Music Professor and lyrics writing and poetry expert (Pattison 2009), describes how to write poetry using sense-bound imagery to enhance a song's emotional impact on its listeners. He tells how a lyric structure can create motion, which, in turn, creates emotion. The main elements used to achieve this effect are rhyme, rhythm, the number of lines, rhyming scheme and rhyme type.

We utilize some of Pattison's guidelines with the aim of creating lyrics that carry an emotional affect on our audience. MABLE begins with a story narrative, provided by the narrative system MEXICA. The rest of the system is broken down into three main modules, which together turn the storyline into lyrics. In particular, this is achieved by extending the original narrative sentence through the addition of a new figurative line of lyrics after each story line. The final ballad consists of alternating lines, where each narrative sentence is followed by a figurative one.

The following are the three main modules of MABLE. The *Sentence Evaluator* creates a set of sentences that fit with a given narrative line using parameters such as rhyme quality, number of syllables, rhyming scheme, and rhyme type. The second module, the *Sentiment Analyzer*, selects from amongst this set of sentences ones that match with the story's sentiments. Finally, the *Integrator* connects the lines and, if needed, changes the figurative sentences into third person narrative. An overview of MABLE's architecture is illustrated in Figure 1.

A second-order Markov model, created using Markovify (https://github.com/jsvine/markovify), has been trained on a corpus consisting of lyrics of 129 love songs by various artists, totally 4697 lines. Titles of top 100 lyrics are taken from an online music catalogue (https://www.last.fm/) and the remaining titles from a NME music blog (http://www.nme.com/blogs) (that contains songs voted by twitter followers). Lyrics to these titles are crawled from 70-80s rock and pop songs available on azlyrics.com. We then cleaned the lyrics corpus by removing punctuation, extra white spaces and repeated text, such as multiple instances of a chorus.

After receiving a sentence from MEXICA's story, a batch of phrases created using the Markov Model is fed to the Sentence Evaluator. The Markov Model is repeatedly called until a set of at least 50 high quality candidate lines are created. The quality of lines is measured on the basis of rhyming score and the number of syllables in the lines.

After candidate figurative lines are attained, they are passed to the Sentiment Analyzer to get emotionally connected with the story. We use linguistic resources to generate the lyrics. Phonetics transcription are taken from the CMU Pronouncing Dictionary, which contains over 134,000 words and their pronunciations. A sentiment analyzer, Twinword API (https://www.twinword.com/api/) is used for extracting sentiments of phrases and categorizing them into positive, negative and neutral. We now discuss MABLE's components in greater detail.

1. **Sentence Evaluator:** This component finds candidate figurative lines that rhyme with a given narrative sentence. It uses the Markov model to generate a batch of 100 new sentences having no more than 60 characters. It then evaluates them based on rhyming quality and number of syllables. If not sufficiently many high quality lines are found, new batches are generated, until we have at least 50 high quality sentences.

   To evaluate rhyming quality, we start with a pool of words that rhyme (to various degrees) with the last word in the story line. To this end, we use NLTK (http://www.nltk.org/) to create the pool of words which is then used to find words in the batch that rhyme with any word in the bag of words. More specifically, we use nltk.tokenize for sentence tokenization, nltk.corpus.cmudict.entries() to discover rhyming words, and nltk.corpus.cmudict.dict() to get the pronunciation of end words.

   For example, if the input narrative sentence is *the priest was ambitious*, then the bag of rhyming words may include the words *auspicious, dishes, discus, judicious,. . . , officious, riches, suspicious*, sorted by their rhyme score. Note that the rhyming words rhyme with the original to various degrees.

   Then, we search our batch (created using the second order Markov model) for sentences ending with words that rhyme with any word in our bag of words. This leads to a greater number of acceptable options than had we required perfect rhyming with the word *ambitious*. Moreover, this flexibility allows our ballads to more closely resemble human-made songs, which often include imperfect rhymes.

   In order to figure out whether two words rhyme, we tokenize the words. To this end, we use the Carnegie Mellon University Pronunciation Dictionary[3] to get phonetic transcription of words. In case a word is not found in the dictionary, we take the last letter of the word. All pronunciations are considered,

[3]The dictionary consists of over 127069 words and their pronunciations. Out of these, 119400 are assigned a unique pronunciation, 6830 have two and 839 have three or more pronunciations.

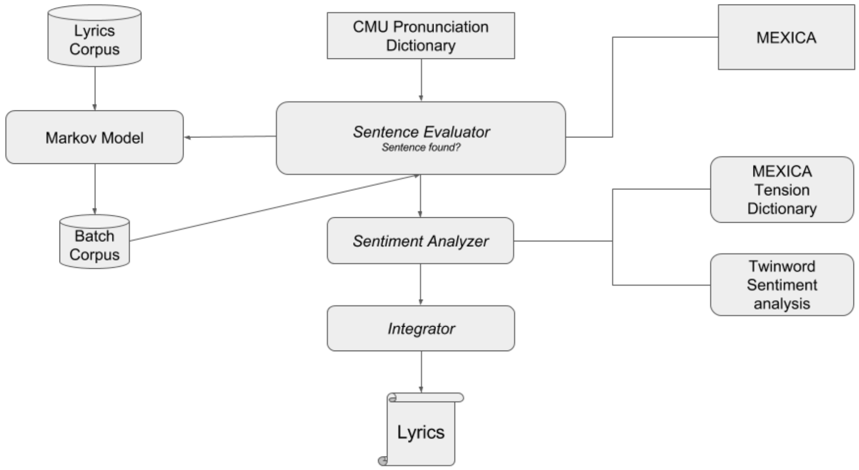

Figure 1: ***An overview of the lyrics generation workflow***. *The workflow depicted here creates a figurative sentence for MEXICA's narrative line. The Sentence Evaluator repeatedly calls a Markov model to attain a list of candidate figurative sentences, which will follow the narrative line in the ballad. After a set of good candidate sentences is obtained, the Sentiment Analyzer finds the ones that express the same sentiments as those found in the narrative sentence. Finally, the Integrator adjusts the selected sentences so that they are told in third person narrative. The algorithm is repeated for each line in the narrative.*

as the dictionary often provides multiple pronunciation options. For each of the pronunciations, we form a tuple list of candidate rhyming words from the CMUdict (http://www.speech.cs.cmu.edu/cgi-bin/cmudict) lexicon and their rhyme qualities (how well they rhyme with the end word in the narrative line).

To calculate the quality of a rhyme, we take into account the degree of phonetic similarities such as end rhymes, slant rhymes, and assonance rhymes.[4] We give high scores to words that end on the same vowel. An alternate rhyming scheme is used, which is a classic, frequently used rhyming scheme, also known as AABB.

We sort the candidate sentences based on their rhyme quality score, selecting the top 50 to go to the next step, which is to count the number of syllables.

CMUdict allows us to count the number of syllables in words, which lets us count the total number of syllables in each sentence. If no valid candidates are found, we return lines constructed using common interjections, while matching the number of syllables, such as "Oho oh oh oh oh." This adds to the poetic feel of our lyrics.

[4] *End rhyme* is the most common type used in poetry, which occurs when the last syllables of words are matched. Its regular use marks off the ending of lines. It makes the lyrics sound appealing while making it easier for the audience to remember. *Slant rhyme* is formed by words with similar consonant sounds. *Assonance rhymes* do not have to rhyme in the traditional sense, but have matching vowels.

2. **Sentiment Analyzer**: Although following all the stylistic constraints of poems gives us well-structured results, we found the lines to be somewhat disconnected from the original story without additional processing. To further improve the lyrics, we incorporate sentiment analysis.

   We connect the story lines with the newly generated ones by using the Twinword API (https://www.twinword.com/api/). This API returns a sentiment score for each word. The sentiment of each sentence is obtained by taking the average of its sentiment scores over all words. The score indicates how positive or negative is the overall line. If the score is below -0.05, it is tagged as negative, and anything above 0.05 is positive. Any line that falls within this range is neutral. We also utilize MEXICA's tensions, such as *love competition* and *clash in emotions*. Whenever these tensions occur in a narrative sentence, MABLE selects candidate lines having opposite sentiments to that of the story line.

   After sentiment analysis is done on the narrative sentence and the candidate figurative lines, we reduce our set of candidates lines to those that carry the same sentiment (positive, negative, or neutral) as the narrative line.

3. **Integrator**:

   While MEXICA tells narratives in third person, most of the lyrics on which we train our Markov model are written in first person. As such, integrating the narrative lines with the figurative sentences often requires a change from a first to a third person point of view.

The point of view decides the distance between the lyrics and the audience. It tells to what extent we know the storyteller and the actors in the story. So, what should we prefer, a first person narrative or a third person narrative?

In third person narrative, the singer acts as a storyteller who directs the audience's attention to an objective world that neither the singer nor the audience is a part of. It is similar to watching a movie while sitting in the audience (Pattison 2009). Neither we, the audience, nor the singer participate in the song's world. In this case, the singer acts as a storyteller. The third person narrative may be cleaner and more focused (Pattison 2009).

After we obtain the emotionally connected lines to our story, we neutralize the point of view using the Integrator. We substitute the first and second person pronouns with third person pronouns to connect the figurative lines with the narrative.

A Step-by-Step Example

To help illustrate our model, we present a step-by-step example creating a figurative sentence for a narrative line. Consider, for example, the following story line, given by MEXICA: *"The princess was in love with the priest."* Our goal is to create a new sentence that will follow it in our lyrics. As such, we would like to create a sentence that rhymes with it, carries a similar sentiment, and has the same number of syllables. This process is subsequently repeated for each sentence in the narrative to form a complete ballad.

1. **Sentence Evaluator**

   The sentence is tokenized into a list of words - ['The', 'princess', 'was', 'in', 'love', 'with', 'the', 'priest']. Using CMUdict, we attain the pronunciation of the last word. In this case, *priest* being the last word, we get [u'P', u'R', u'IY1', u'S', u'T']. Then, again using CMUdict, we get a bag of words that rhyme with *priest*: [beast, east, feast, increased, least...]. The numerical quality of rhyming words is calculated by the degree of phonetic transcription similarity of words. In particular, the rhyme quality is given by the number of consecutive matching pronunciation elements, starting from the end of the words. In case of ties, we give preference to words whose pronunciations end on the same vowels or words having the same vowels at the same places (for example, *moon* and *loop*.)

   Then we repeatedly call *getMarkovBatch()* to get candidate sentences whose last words rhyme with the above bag of words, excluding sentences that end with exactly the same word.

   Please note that the candidate lines given below do not perfectly rhyme with the word *priest*. In case perfect rhymes are not found, our model picks lower quality rhyming lines. Since imperfect rhymes are common in human-made lyrics, variety in rhyme quality also improves our lyrics.

   Here are examples of some candidate sentences:

   - Now that I have lost my light
   - wondering where would I have got
   - You can tell the lovers to part
   - Down the highway of regret
   - Mornings where blue and clouds of white
   - So I stop and think about it

   Next, we trim our list of candidate sentences to those whose number of syllables matches those in MEXICA's sentence. As such, our list is reduced to the following:

   - You can tell the lovers to part
   - wondering where would I have got
   - The leader of the holy ghost
   - Mornings where blue and clouds of white
   - So I stop and think about it

2. **Sentiment Analyzer**

   This step further reduces our set of candidates by performing sentiment analysis, and eliminating sentences that carry a different sentiment from MEXICA's sentence. Sentiment analysis rates each sentence into three categories: positive, negative, and neural score range.

   The following are some sentences from our list of candidate figurative lines along with their emotional score, where 1 represents positive, 0 represents neutral, and -1 corresponds to negative sentiment:

   - You can tell the lovers to part, 1
   - wondering where would I have got, 0
   - The leader of the holy ghost, 1
   - Mornings where blue and clouds of white, -1
   - So I stop and think about it, 0

   This gives us a final list of candidate sentences. Since the narrative sentence that we are working from has neural sentiment, our final set of figurative sentence candidates consists of the following two:

   - wondering where would I have got
   - So I stop and think about it

   Then, we pick a random sentence from the remaining figurative lines and pass it to the Integrator.

3. **Integrator**

   After we select a figurative sentence, we pass it to the Integrator to substitute first person with third person pronouns.

   Here are the resulting lyrics:

   The princess was in love with the priest
   Wondering where would they have got

   The process is repeated for each of MEXICA's lines, giving complete lyrics. An example is shown in Figure 2. Every odd number sentence is created by

The priest was born under grace of the great god
And evolving from the shadows lifted
The lady was an inhabitant of the great city
But just remember there's a sign of intensity

The lady wanted him from the start
The friends that they are
The lady hid her love for the priest
They just don't think they have got a secret

But she fell in love with him
They don't know of the time
The princess was in love with the priest
Wondering where would they have got

The princess admired the lady
Movies only make them crazy
She felt much affection for her
They thought the world were on fire

The priest was ambitious
He will be out of place
He wanted power easily
As time goes by so slowly

Figure 2: *An example of a ballad made by MABLE.*

MEXICA, whereas every even-numbered lyric sentence is created using the above process, by creating a suitable figurative line for each of MEXICA's narrative sentences.

Evaluation

Evaluation was performed through a user study comparing MABLE's lyrics with those created by two other systems, as well as lyrics written by humans. The subjects consisted of 30 students, 26 of which were undergraduate and 4 of which were graduate students, with an average age of 24 years old. Among these students, 26 were male and 4 were female. Subjects were not informed of how any of the lyrics had been made.

The survey asked them to rate lyrics along three parameters: coherence, emotional engagement, and the lyrics' overall quality. Specifically, the following three questions were asked for each of the lyrics provided: (1) Rate the storyline in these lyrics (is the plot coherent?) (2) How emotionally engaging are these lyrics (does it transmit feelings)? (3) Overall, how would you rate these lyrics?

Answers were provided via the standard 5 option Likert Scale, rating quality from *very poor* (1) to *very good* (5). For human-written lyrics, we selected professionally-written lyrics but which our subjects may not be familiar, as such avoiding the work of recently popular artists. Further, to align with MABLE's goal, we opted for lyrics that tell a story, choosing *Hall Of The Mountain King* by Denise Beadle, Peter Olliff, Peter Smith, and Edvard Grieg.

In addition to lyrics by MABLE, we have also included an artifact by a rap lyrics generator (Hieu Nguyen 2009). To give a sense for the style, we include the first four lines here:
my nasty new street slugger my heat seeks suckers
now i'm a pimp you a player
i i'll rob boys ii men like i'm michael bivins
if i'm from southside jamaica queens nigga ya'heard me

Finally, we also compared with lyrics created with the full-face model (Colton, Goodwin, and Veale 2012). The first few lines of the lyrics we used are given here:
the repetitive attention of some traditional african chants
a heroic struggle, like the personality of a soldier
an unbearable symbolic timing, like a scream
blue overalls, each like a blueberry
some presidential many selfless leaders

Results are summarized in Figure 3.

Discussion

The development of a system that produces lyrics based on fables is an interesting challenge for computational creativity. One on hand it requires an agent capable of producing fables; on the other hand, it is necessary to modify such stories to satisfy lyrical requirements such as rhythm, rhyme, and so on. We are not aware of any other system that generates similar pieces.

Our results suggest that subjects clearly preferred the human lyrics. We believe that one of the reasons for it is the smooth integration of the entire text into a single coherent unit. MABLE takes a given story and then expands it, line by line, to generate lyrics. Our model not only adds rhetorical language that satisfies rhythm and rhyme constraints but which is also semantically related to the original phrase. The results presented in this work suggest that our approach is a good initial step, although much more work is required to integrate the new sentence in a more natural way.

Our study shows a correlation between the overall quality of a poem's and narrative's coherence. This suggests that subjects prefer lyrics with coherent narratives. As such, this may be an important characteristic for creative agents. This work is the first to tackle the challenge of developing lyrics with a coherent plotline, which may be essential for matching human quality lyrics.

Although MABLE was ranked highest amongst the computer-generated lyrics based on the coherence assessment, its advantage was reduced in the overall evaluation and it got second place in the emotional assessment. This result suggests that the use of strong language produced a clear impact in our subjects' evaluation of the lyrics. The poetry made by the rap lyrics generator (Hieu Nguyen 2009), which scored highest amongst computer generated systems on emotional engagement, contains words that evoke an emotional response (see above for an excerpt). Thus, we need to develop a mechanism that allows MABLE to analyse

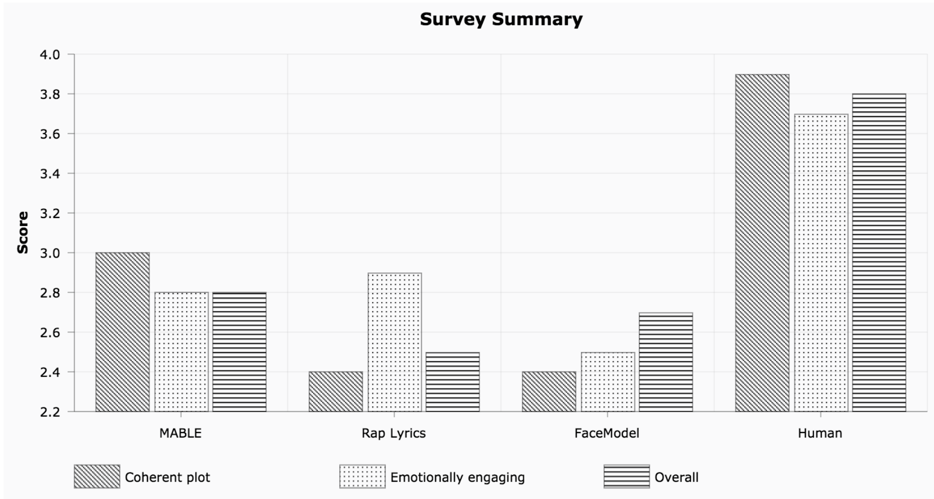

Figure 3: *Average scores for each of the lyrics in our survey, along the following parameters: Coherent plot, Emotionally engaging, and Overall. While human-made lyrics clearly perform best on all criteria, MABLE's lyrics score highest overall and on plot coherency amongst computer created poetry.*

and, when appropriate, substitute some of the words in the figurative sentences with more emotionally impactful words. It would be interesting to explore whether genres outside of rap lend themselves to emotionally engaging word choices, and how such choices could be identified and utilized in an automated lyrics writer.

Setting it to Music

Before concluding, we briefly investigate the sonification of MABLE's artifacts. Ultimately, lyrics are set to music, so that they could be expressed in their most natural form - through the singing voice. To explore the potential of MABLE's lyrics to fit within musical works, we utilize ALYSIA (Ackerman and Loker 2017), a system designed to create melodies for user-provided lyrics.

Combining MABLE with ALYSIA brings us one step closer to autonomous songwriters. To examine the potential of their integration, we run ALYSIA on two verses created in a poem by MABLE. The sentences were given to ALYSIA one at a time, for each of which ALYSIA provided melody options (vocal melodies to which the lyrics could be sang). The melodies were subsequently selected and combined by Ai Nakamura, a computer science undergraduate student with an associates degree in music theory and composition. Figure 4 shows the resulting song.

Conclusions and Future work

Songs are a universal channel for expressing human needs, desires, and goals. Unfortunately, not everybody has the chance to pursue this form of self-expression. Our models can be used to support the development of skills required for producing lyrics. A first step might be to allow a collaboration between MABLE and human learners. For instance, MABLE would present the user with examples of lyrics. Then, the system and the learner together would collaborate to generate new pieces; employing her routines for evaluation, MABLE could provide feedback to the user. Furthermore, a competition could be organized, allowing human judges to evaluate lyrics made by MABLE and the learners.

We believe that one of the most intriguing aspects of our work is the combination of different systems. Most notably, MABLE relies heavily on MEXICA's narrative writing ability. Yet, we can go further. For instance, rather than generating a full story and then modifying it, we are analysing the possibility of integrating both processes. In this way, the unraveling of the plot might be constrained by the figurative language employed for the lyrics and vice versa. How does this integration affect the ER-Model and the production of figurative language? In the same way, we are interested in identifying parameters that allow for automatically producing (e.g. from the web) a corpus that better suits the content of the stories used for the generation of lyrics.

We also briefly explored interaction between MABLE and melody-writing system ALYSIA. Future work will explore in-depth collaboration between these systems. The aim of this collaboration is twofold. Firstly, we wish to create complete works, exploring their combined creative potential. Secondly, we would like to investigate modes of collaboration between these two system, exploring automated systems' ability to collaborate on an artifact in a fully integrated fashion, perhaps resembling that of two human artists who may alter both the lyrics and melody in an iterative fashion.

References

[Ackerman and Loker 2017] Ackerman, M., and Loker, D. 2017. Algorithmic songwriting with alysia. *Inter-*

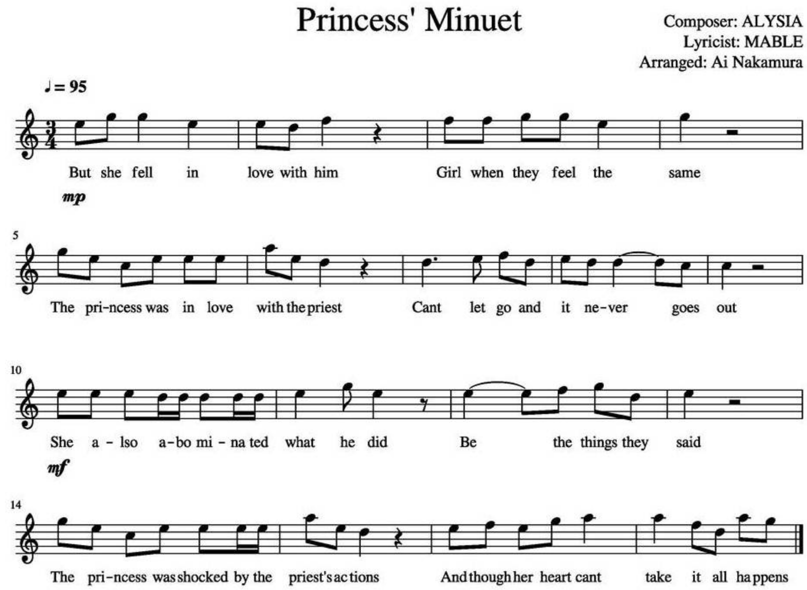

Figure 4: *Melodies for two verses created by MABLE. Each of MABLE's sentences were given to songwriting system ALYSIA, which created melodies for these sentences. The melodies where arranged by Ai Nakamura.*

national Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMUSART).

[Colton, Goodwin, and Veale 2012] Colton, S.; Goodwin, J.; and Veale, T. 2012. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

[Ghazvininejad et al. 2016] Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1183–1191.

[Gonçalo Oliveira 2015] Gonçalo Oliveira, H. 2015. Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. *Journal of Artificial General Intelligence* 6(1):87–110.

[Hieu Nguyen 2009] Hieu Nguyen, B. 2009. Rap lyric generator.

[Malmi et al. 2015] Malmi, E.; Takala, P.; Toivonen, H.; Raiko, T.; and Gionis, A. 2015. Dopelearning: A computational approach to rap lyrics generation. *arXiv preprint arXiv:1505.04771*.

[Misztal and Indurkhya 2014] Misztal, J., and Indurkhya, B. 2014. Poetry generation system with an emotional personality. In *Proceedings of the 5th International Conference on Computational Creativity.*

[Pattison 2009] Pattison, P. 2009. *Writing better lyrics.* Writer's Digest Books.

[Pérez y Pérez and Sharples 2001] Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

[Pérez y Pérez 2007] Pérez y Pérez, R. 2007. Employing emotions to drive plot generation in a computer-based storyteller. *Cognitive Systems Research* 8(2):89–109.

[Pérez y Pérez 2015] Pérez y Pérez, R. 2015. A computer-based model for collaborative narrative generation. *Cognitive Systems Research* 36:30–48.

[Scirea et al. 2015] Scirea, M.; Barros, G. A.; Shaker, N.; and Togelius, J. 2015. Smug: Scientific music generator. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 204.

[Sharples 1999] Sharples, M. 1999. How we write. *Writing as creative design. London and New York: Routledge.*

[Tobing and Manurung 2015] Tobing, B. C. L., and Manurung, R. 2015. A chart generation system for topical metrical poetry. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 308.

[Toivanen et al. 2013] Toivanen, J. M.; Järvisalo, M.; Toivonen, H.; et al. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the International Conference on Computational Creativity*, 160–167.

[Watanabe et al. 2017] Watanabe, K.; Matsubayashi, Y.; Inui, K.; Nakano, T.; Fukayama, S.; and Goto, M. 2017. Lyrisys: An interactive support system for writing lyrics based on topic transition. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, 559–563. ACM.