WeirdAnalogyMatic: Experimenting with Analogy for Lyrics Transformation

Hugo Gonçalo Oliveira
CISUC, Department of Informatics Engineering
University of Coimbra, Portugal
hroliv@dei.uc.pt

Abstract

This paper is on the transformation of text relying mostly on a common analogy vector operation, computed in a distributional model, i.e., static word embeddings. Given a theme, original song lyrics have their words replaced by new ones, related to the new theme as the original words are to the original theme. As this is not enough for producing good lyrics, towards more coherent and singable text, constraints are gradually applied to the replacements. Human opinions confirmed that more balanced lyrics are obtained when there is a one-to-one mapping between original words and their replacement; only content words are replaced; and the replacement has the same part-of-speech and rhythm as the original word.

Introduction

Materialisations of human creativity rarely start from scratch. Consciously or not, artists are inspired by what they experience, including other artists and their creations. This is also true in the scope of Computational Creativity, where many systems rely on an inspiration set. When it comes to linguistic creativity, poetry generation systems may rely on a corpus of human-created poems where templates (Toivanen et al. 2012; Colton, Goodwin, and Veale 2012) or language models (Potash, Romanov, and Rumshisky 2015; Yan 2016) are acquired from; the initial population of a genetic algorithm is derived from (Hämäläinen and Alnajjar 2019b); or a poem is selected and transformed to meet desired constraints (Bay, Bodily, and Ventura 2017).

We also follow a transformation approach, mostly supported by analogy, for producing new text. Briefly, given an original text, in this case, lyrics of a known song, and a word representing a new theme, we compute analogies on a distributional semantics space and shift the theme of the lyrics by replacing some of its words according to computed analogies. For this, we apply a common method for solving analogies of the kind '*what is to b as a is to a'?*', known as the vector offset or 3CosAdd, and popularised for assessing traditional models of word embeddings, like word2vec (Mikolov et al. 2013) or GloVe (Pennington, Socher, and Manning 2014), both known for keeping syntactic and semantic regularities. Analogies are solved with the following operation on the vectors of the involved words $\vec{a} - \vec{a'} + \vec{b} \approx \vec{b'}$ (e.g., a common example is $\vec{king} - \vec{man} + \vec{woman} \approx \vec{queen}$).

Our main goal is thus to explore to what extent we can rely on word embeddings for transforming the semantics of a poem, in such a way that its theme shifts according to the seed, while text remains syntactically and semantically coherent. Transforming text, rather than generating it from scratch, should help to maintain the latter. For this, we make a rough approximation that the song title summarises its theme and every word in the lyrics is related to this theme. Relying on this assumption and recalling how analogies can be computed, shifting the theme is a matter of computing analogies of the kind '*what is to the new theme as the original title is to a word used?*'.

However, we soon noticed that text resulting from the exclusive application of the analogy operation had a series of issues. Therefore, we describe some constraints introduced towards better lyrics, e.g, to guarantee that functional words are not changed, syntax is coherent, or the original metre is kept. Yet, although more constraints lead to better structure and singability, they lower the chance of selecting related words, with a negative impact on the theme shift. To analyse the impact of different constraints on aspects like grammar, semantics, novelty or singability, a selection of results with different constraints was subjected to a human evaluation, which suggested that there should be a one-to-one mapping between original words and their replacement, only content words should be replaced, and the replacement must have the same part-of-speech and rhythm as the original word. Although our experiments were performed in song lyrics, this would work similarly in any kind of poetry, or other textual genres.

The proposed approach constitutes the engine of a system for lyrics transformation, which we baptised as WeirdAnalogyMatic (WAM) because the obtained results could potentially be followed in the creation of parodies from known songs, popularised by artists such as Weird Al Yankovic – e.g., with hits like *Eat it* (transformation of Michael Jackson's *Beat it*), *Smells Like Nirvana* (based on Nirvana's *Smells Like Teen Spirit*), or *Like a Surgeon* (based on Madonna's *Like a Virgin*). This kind of parody has also featured several comedy TV shows (e.g., Saturday Night Live) and advertising campaigns (e.g., *These Bites are Made for Poppin'* a 2006 Pizza Hut ad by Jessica Simpson for Super Bowl, which is a transformation of *These Boots are Made for Walkin'*; or the 2000 TV ad for Mountain Dew, a trans-

Proceedings of the 11th International Conference on Computational Creativity (ICCC'20)
ISBN: 978-989-54160-2-8

formation of Queen's *Bohemian Rhapsody*). All of those examples suggest that attempting at the automation of this creation procedure may be worth.

The remainder of the paper briefly overviews different approaches for poetry and song lyrics generation, with a focus on those that, along the way, exploit word embeddings. We then describe our approach and illustrate with the result of adding more constraints, step-by-step. Before concluding, we present the results of the evaluation survey, together with examples of the most and least appreciated lyrics.

Related Work

Poetry generation has long been a research topic in Computational Creativity, with much work during the last 20 years (Gonçalo Oliveira 2017). A prominent approach is the generation based on templates, instantiated by similes (Colton, Goodwin, and Veale 2012), instances of other relations (Gonçalo Oliveira and Oliveira Alves 2016), or by replacing certain words with others, with the same part-of-speech (PoS) (Agirrezabal et al. 2013), or associated to a target subject (Toivanen et al. 2012). While templates generally guarantee that syntactic rules are met, towards semantic coherence, poetry generators often have to rely on a model of semantics. For this, semantically-related words can be acquired from semantic networks (Agirrezabal et al. 2013; Gonçalo Oliveira and Oliveira Alves 2016), models of word associations (Toivanen et al. 2012), or of distributional semantics, such as word embeddings (Ghazvininejad et al. 2016; Hämäläinen and Alnajjar 2019a).

Alternative approaches to text generation, including creative text, are based on language models, which can be learned from large corpora with recurrent neural networks (Yan 2016), often with LSTM layer(s) (Potash, Romanov, and Rumshisky 2015). Yet, recently, the generation of different kinds of text has been attempted with larger transformer-based language models, like GPT-2 (Radford et al. 2019), fine-tuned for a specific domain. In any of the previous, the first step is to learn word embeddings from a corpus on the target domain.

Not so different from template-based, one last alternative for producing new text is to start with a single original text and replace some of its words towards the desired intent. Such an approach was used for generating lyrics for parodies inspired by daily news (Gatti et al. 2017), achieved by expanding content words of a headline with WordNet (Fellbaum 1998) and WikiData, then used for replacing words in original lyrics, having in mind syntactic (PoS) and metric constraints. Distributional semantics was not considered. Another common application is in the generation of shorter texts, like headlines (van Stegeren and Theune 2019) or slogans (Repar et al. 2018), where domain vocabulary can be expanded with word embeddings.

Also in the context of creative systems, the operations of similarity, neighbours, theme, and analogy in a word embedding space were formalised and used for producing song lyrics, with the selection of replacement words constrained by the given intentions (e.g., form, theme, sentiment) (Bay, Bodily, and Ventura 2017). Out of those operations, we focus exclusively on analogy and assume that all words in the lyrics are somehow related to a theme, which we approximate by the song title. The paper is focused on experiments and necessary workarounds for taking advantage of analogy and still have a result that is not only syntactically and semantically coherent, but also singable.

Step-by-Step Approach

Our goal is to transform a given text, so that it is still meaningful, but its semantics shifts to a new theme t_n, given by a single word. For this, we assume that every word w_o in the original text is somehow related to a fixed meaning in a distributional space, seen as the original theme t_o. We then rely on analogy for computing new words w_n for replacing each w_o. In our experiments, we use song lyrics and make the rough approximation that t_o can be obtained from the song title[1], i.e., we use a model of distributional semantics for computing t_o as the weighted average of the vector of all content words in the title. Since, at least in the tested models, words are ordered according to their frequency in the training corpus, we used their index in the model as their weight. This can be seen as a cheap approximation to word relevance, because more frequent words (i.e., less relevant) will have a lower index, thus lower weight, while less frequent ones (i.e., more relevant) will have a higher index.

To wrap it up, we assume that every word w_o in the original lyrics is to the theme t_o as a new word w_n is to a new theme t_n. So, once a new theme t_n is selected, we can, for every w_o, apply the 3CosAdd analogy solving method to the vectors of the involved words, and compute w_n as follows: $w_n = w_o - t_o + t_n$.

Yet, we soon realised that following this with no additional constraints resulted in text that was both hard to sing and ungrammatical. For minimising those issues, some constraints were added to the process of lyrics transformation. Such constraints are thoroughly described in this section, with their impact illustrated by results obtained. Different models of word embeddings were tested, but all results reported were obtained with the Stanford GloVe word vectors[2] (Pennington, Socher, and Manning 2014), with 300 dimensions, pre-trained in a corpus of 6B tokens from Wikipedia and Gigaword 5. Though originally applied to word2vec models, 3CosAdd is also applicable to GloVe and generally achieves better performance in semantic analogies (Pennington, Socher, and Manning 2014).

Examples presented here used as input the lyrics of *Smells Like Teen Spirit* (hereafter, SLTS), by Nirvana[3], with original lyrics in figure 1. The original theme is given by $t_o \approx \alpha.smells + \beta.like + \gamma.teen + \delta.spirit$, where α, β, γ and δ are the index-based weights. In this case, t_o is a vector close to *smells*, the word with higher index value.

[1]Even if this does not hold for many lyrics, it was our experimentation setting. Alternative theme approximations may consider the average embedding of the first line, the chorus, or the full song.

[2]`https://nlp.stanford.edu/projects/glove/`

[3]Given the controversy around the actual meaning of these lyrics and their connection with the title, this was, arguably, not the best choice, but it suits the purpose of illustrating the procedure.

Load up on guns, bring your friends
It's fun to lose and to pretend
She's over-bored and self-assured
Oh no, I know a dirty word
Hello, hello, hello, how low
Hello, hello, hello, how low
Hello, hello, hello, how low
Hello, hello, hello
With the lights out, it's less dangerous
Here we are now, entertain us
I feel stupid and contagious
Here we are now, entertain us
A mulatto, an albino, a mosquito, my libido

Figure 1: Original lyrics of *Smells Like Teen Spirit*.

Only analogy

The first attempt to test our hypothesis was to rely exclusively on analogy. Each line of the lyrics was first tokenized with the Stanford CoreNLP toolkit[4] (Manning et al. 2014), and then every single word w_o in the original lyrics was replaced by a new word w_n, that would be to the new theme t_n as w_o was to the original theme t_o. Results immediately confirmed that this would not be enough for our purpose. This is illustrated in figure 2, where the title and first two lines of SLTS are presented for two different values of t_n.

t_n=*computational*, **Title:** *computation mathematical mathematical theoretical*
computation theoretical theoretical theoretical, *theoretical theoretical mathematical* *theoretical's mathematical mathematical mathematical* *mathematical mathematical mathematical*
t_n=*art*, **Title:** *arts works arts arts*
loads work arts gun, work work arts *part's arts work losing arts work do*

Figure 2: Lyrics for STLT with only analogy.

Keep stopwords

Although we could expect that computed analogous words would have the same PoS, thus ensuring that the grammatical structure was kept, this was not always the case. A possible cause is that, because functional words are used in many contexts, they end up having a limited contribution to the analogy. Thus, when w_o is a functional word, the computed w_n tends to be similar to t_o, often with the same PoS as t_o. To fix this, we used the list of English stopwords included in CoreNLP[5] and only replaced words not in this list. Once we keep stopwords, with the previous values for t_n, we get, for the first five and last line of SLTS, the text in figure 3.

There are still issues, but keeping stopwords is a step towards more grammatical text. We further note that the adjective 'over-bored' remains unchanged, which happens because the tokenizer considered it a single word for which there is not a vector in the embedding model. But now the main problem is that, at least for some themes t_n, the same word w_n is used many times. As analogy computed this way is not a one-to-one relationship, the most similar word to the resulting vector is sometimes the same, thus resulting in replacing different words w_n by the same w_o. A final issue occurs in the last lines, where articles are incorrect, but this is fixed with simple replacements, such that 'an' precedes words starting with *a*, *e*, *i*, *o*, and 'a' the others.

[4] https://stanfordnlp.github.io/CoreNLP/

[5] https://github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/stopwords.txt

t_n=*computational*, **Title:** *computation mathematical mathematical theoretical*
computation up on theoretical, theoretical your mathematical *it's mathematical to mathematical and to mathematical* *she's over-bored and computation* *mathematical no, i mathematical a mathematical mathematical* *computation, computation, computation, how theoretical* *...* *a computation, an computation, a computation, my cognitive*
t_n=*art*, **Title:** *arts works arts arts*
loads up on gun, work your arts *it's arts to losing and to do* *she 's over-bored and arts* *arts no , i what a painting arts* *gallery, gallery, gallery, how high* *...* *a artist, an curator , a museum , my museum*

Figure 3: Lyrics for STLT with the Stopwords constraint.

Replacements history

To avoid repetition and guarantee a one-to-one correspondence between w_o and w_n, a map can be used for the history of replacements made. Such that, when selecting a word w_n for replacing w_o, the history may be looked up and, if w_n was already used as a replacement for a different word than w_o, it is not used. Instead, out of the words not previously used in the lyrics, the most similar to the analogy is selected. This is tested for each word in the model, ranked according to its similarity to the computed analogy vector, until a usable word is found or a predefined rank is reached. For the reported experiments, the maximum rank was 2,500, meaning that, at most, the 2,500 most similar words were tested.

We stress that, even if lower-ranked words will probably not be exactly an analogy, they should be similar to t_n, or on the same topic. Once the map is integrated, with the previous values for t_n, the first five lines of the lyrics of SLTS become those in figure 4.

Part-of-Speech tagging

Controlling repetition lead to better lyrics, but also made still existent grammatical issues more clear. Examples include *losing*, in the second line of both lyrics shown, or lines like *physics no, i understand a methods translation* or *gallery no, i what a painting literature*. Therefore, to select words w_n that match the PoS of a target w_o, we did the following: (i) PoS-tag each line in the original lyrics with the CoreNLP PoS tagger; (ii) For each open w_o (nouns, verbs, adjectives, adverbs) and candidate replacement w_c, create a new line where w_o is replaced by w_c; (iii) PoS-tag the new line; (iv) If the sequence of PoS tags is the same as the original, use w_c

t_n=*computational*
Title: *computational mathematical theoretical creativity*
computation up on rifles , solve your colleagues *it's mathematics to losing and to algorithms* *she's over-bored and probabilistic* *physics no, i understand a methods translation* *bioinformatics, bioinformatics, bioinformatics, how molecular*
t_n=*art*, **Title:** *art works arts museums*
loads up on gun, work your friend *it's exhibition to losing and to do* *she's over-bored and exhibited* *gallery no, i what a painting literature* *photography, photography, photography, how high*

Figure 4: Lyrics for STLT with the History constraint.

as w_o, otherwise, test the following most similar candidate. For SLTS, this results in the lyrics of figure 5.

t_n=*computational*, **Title:** *utilizes like modeling creativity*
computation up on rifles, solve your colleagues *it's mathematics to need and to underestimate* *she's over-bored and self-assured* *oh no, i understand a computational translation* *hello, hello, hello, how theoretical ...* *with the algorithms out, it's less mathematical* *here we are still, enlighten us* *i work conceptual and probabilistic* *here we are still, enlighten us* *a dynamical, a neuroscience, an epidemiology, my optimization*
t_n=*art*, **Title:** *teaches like art museum*
design up on arts, work your artists *it's exhibition to take and to teach* *she's over-bored and self-assured* *oh no, i think a contemporary literature* *hello, hello, hello, how high ...* *with the paintings out, it's less known* *here we are still, participate us* *i want naive and infectious* *here we are still, participate us* *a conceptual, a curator, an exhibit, my photography*

Figure 5: Lyrics for STLT with the PoS constraint.

Although this works for most cases, a limitation arises from the fact that the PoS tagger was trained in sentences of the Wall Street Journal: besides being very different from the style of lyrics, the lines of the latter rarely correspond to complete sentences. Therefore, a minority of grammatical issues is still expected.

Considering the Metre

With previous fixes, new lyrics can be produced on the new theme, also meeting grammatical constraints. Yet, several results are hard to sing in the rhythm of the original song melodies. To improve this, in addition to the previous constraints, selected words w_n have to match the metre of the original words w_o. More precisely, each w_n must have the same number of syllables and the position of its primary stress must coincide with the target w_o. This information can be acquired from the CMU Pronouncing Dictionary[6]. As the lyrics in figure 6 show, with this constrain, text is easier to sing.

t_n=*computational*, **Title:** *solves like math theory*
weight up on tools, solve your skills *it's skill to need and to assume* *she's over-bored and self-assured* *oh no, i work a finite phrase* *hello, hello, hello, how high ...* *with the lamps out, it's less difficult* *here we are still, interrupt us* *i sense greedy and infectious* *here we are still, interrupt us* *a stochastic, a regression, a prevention, my cognition*
t_n=*art*, **Title:** *writes like art culture*
weight up on arts, work your works *it's dance to take and to afford* *she's over-bored and self-assured* *oh no, i think a public name* *hello, hello, hello, how high ...* *with the shows out, it's less serious* *here we are still, introduce us* *i want crazy and infectious* *here we are still, introduce us* *an artistic, a curator, a museum, my exhibit*

Figure 6: Lyrics for STLT with the Metre constraints.

Considering Rhymes

Beyond metre, a final constraint concerned rhymes. One possibility, perhaps the most natural, would be to guarantee that pairs of words that rhymed in the original lyrics still rhyme in the new. However, in order to better resemble the sound of the original song, also because it was easier, we opted to constrain w_n such that it rhymes with the target w_o. This was achieved by selecting w_n only if its termination has the same sound as w_o's, again according to the CMU Dictionary. For SLTS, this results in the lyrics of figure 7.

The first impression is that, although the new words rhyme, many end up not being changed, because no word with the required termination is found in the most similar 2,500. This has a negative impact on novelty (i.e., many words are the same as in the original lyrics) and relatedness with the new theme t_n is low. To analyse the impact of testing more words, the maximum number of similar words was set to 100,000, with results in figure 8.

Although more words are indeed replaced, the topic is still too distant from t_n. With 2,500 similar words, few w_o are replaced, thus not shifting the theme enough to t_n, but with 100,000, many words are replaced by others that are not that clearly related to t_n, and definitely not an analogy of the desired kind. In fact, this does not happen only when the rhymes constraint is added. The analysis of these and other results confirmed that adding constraints has a positive impact on coherence of text and metre, but the relation of words w_n to t_n is also gradually weaker. Observation also suggested that a good equilibrium between coherence and

[6] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

t_n=*computational*, **Title:** *smells like teen spirit*
mode up on guns, bring your friends
it's fun to choose and to intend
she's over-bored and self-assured
oh no, i go a dirty word
hello, hello, hello, how slow ...
with the lights out, it's less dangerous
here we are now, entertain us
i feel stupid and contagious
here we are now, entertain us
a mulatto, an albino, a mosquito, my libido

t_n=*art*, **Title:** *sells like scene spirit*
mode up on sons, bring your friends
it's fun to choose and to intend
she's over-bored and self-assured
oh no, i go a dirty word
hello, hello, hello, how slow ...
with the nights out, it's less dangerous
here we are now, entertain us
i deal stupid and contagious
here we are now, entertain us
a mulatto, an albino, a mosquito, my libido

Figure 7: Lyrics for STLT with the Rhymes constraint.

t_n=*computational*, **Title:** *smells like jean spirit*
mode up on tons, wring your ends
it's sun to choose and to intend
she's over-bored and self-assured
oh no, i go a dirty nerd
hello, hello, hello, how slow ...
with the bytes out, it 's less dangerous
here we are now, entertain us
i deal stupid and courageous
here we are now, entertain us
a mulatto, an albino, a mosquito, my libido

t_n=*art*, **Title:** *sells like scene spirit*
mode up on sons, string your ends
it's sun to choose and to intend
she's over-bored and self-assured
oh no, i go a dirty bird
hello, hello, hello, how slow ...
with the nights out, it's less dangerous
here we are now, entertain us
i deal stupid and courageous
here we are now, entertain us
a mulatto, an albino, a burrito, my tuxedo

Figure 8: Lyrics for STLT with the Rhymes constraint tested in 100,000 similar words.

relatedness is achieved with all constraints but the rhymes, which is further analysed in the following section.

Due to space limitations, we cannot show many resulting lyrics, but resulting titles give a good idea of what happens for different values of t_n. See table 1 for a selection of titles obtained with the Metre constraints.

Evaluation

Given the underlying subjectivity, in order to confirm our initial conclusions, we relied on the opinion of humans. For this purpose, we prepared a survey for the assessment of relevant aspects of the resulting lyrics, namely novelty, grammar, semantics, singability and overall appreciation. The questions of the survey were uploaded to Amazon Mechanical Turk[7] with the following instructions:

- Summary: Answer the following questions regarding the proposed new lyrics, considering popular songs and their original lyrics. Thank you very much for your help!
- Detailed Instructions: (i) Recall the following song and, if you need, listen to it, e.g., on Youtube (URL); (ii) Read the new proposed lyrics and answer the following questions on different aspects; (iii) The meaning of the slider values is: 1–Strongly disagree, 2–Disagree, 3–Neutral, 4–Agree, 5–Strongly agree; (iv) All answers are mandatory.

Figure 9 is an example of an assignment, which comprised seven questions, aiming to assess selected aspects. Six of those questions were to be answered with a 5-point Likert scale, namely asking: (i) Whether the judge was **familiar** with the song, which would later enable to ignore answers by unfamiliar users; (ii) How different the new lyrics were to the original (roughly, **novelty** towards the original song); (iii) How grammatical were the new lyrics (**grammaticality**); (iv) How semantically coherent the new lyrics were (**semantics**); (v) How easy it was to sing the new lyrics with the melody of the original lyrics (**singability**); (vi) What was the overall appreciation of the new lyrics (**overall**). The fifth question asked the judge to select the best **topic** for the song. Given a list of eight words, they had to pick one, or none. This included the six themes used for producing the lyrics in this evaluation – *art*, *computational*, *eat*, *elections*, *sick*, *sing* – plus two additional words – *love*, *war*. Our hypothesis is that selecting t_n as the topic is a strong indicator that the new lyrics are indeed about t_n.

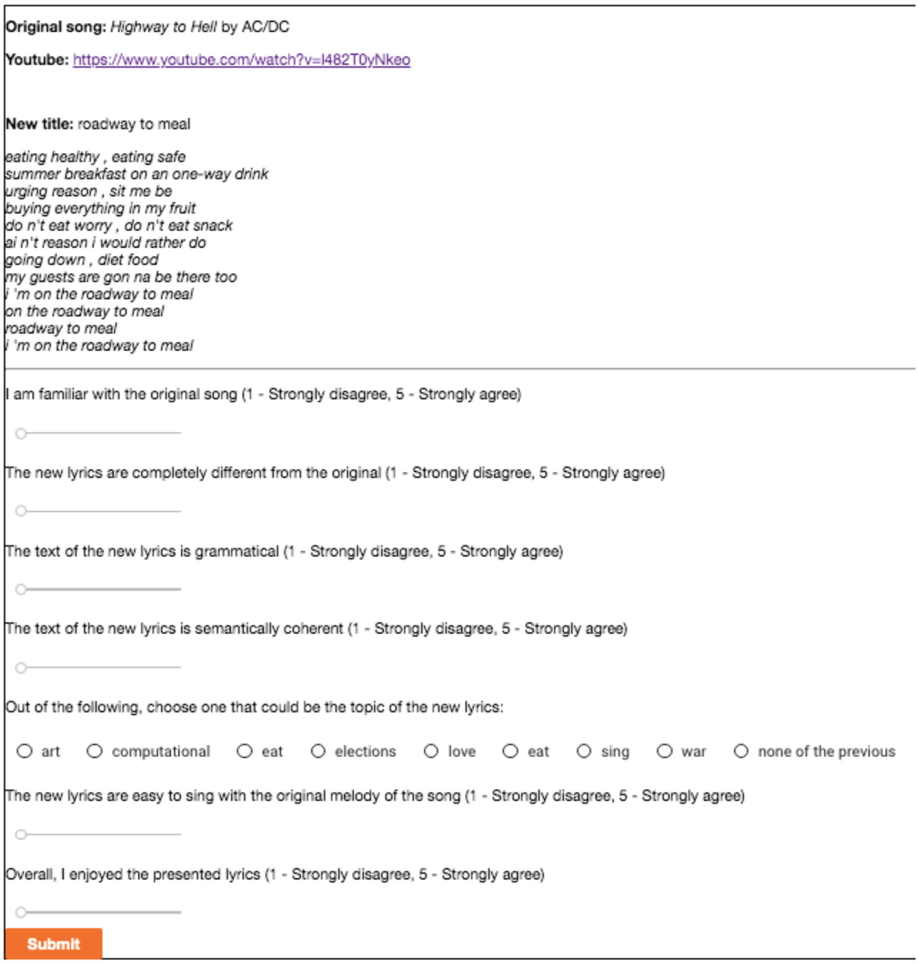
The aforementioned themes were used for producing lyrics with four different configurations: (i) Keep stopwords + Replacements history (History); (ii) Previous + PoS tagging (PoS); (iii) Previous + Metre (Metre); (iv) Previous + Rhymes (Rhymes). For producing the lyrics with each configuration, the 2,500 most similar words were always tested for replacement. The following original songs were used: (i) *Beat it*, by Michael Jackson; (ii) *Enjoy the Silence*, by Depeche Mode; (iii) *Heroes*, by David Bowie; (iv) *Highway to Hell*, by AC/DC; (v) *Smells Like Teen Spirit*, by Nirvana.

Combining the six themes and the five original lyrics, 30 different lyrics were generated for each configuration. As four different configurations were tested, the evaluation set had 120 different lyrics. For each of those, three different judges answered the previous survey, resulting in 360 completed assignments, on which we can rely for comparing the results of each configuration, on each targeted aspect. For each configuration and assessed aspect, figure 2 shows the Mode (Mo) and the Median ($\tilde{x}$) of all aspects, according to the judges. An exception is the topic aspect, which could have multiple answers, but only one was correct. In this case, the table shows the proportion of assignments for which the selected topic was Correct ($= w_n$) or None.

[7] https://www.mturk.com/

t_n	Original					
	smells like teen spirit	*highway to hell*	*heroes*	*enjoy the silence*	*beat it*	*born to be wild*
art	*writes like art culture*	*sculpture to art*	*paintings*	*include the painting*	*work it*	*worked to be fine*
computational	*solves like math theory*	*roadway to math*	*methods*	*derive the quantum*	*solve it*	*coined to be brute*
eat	*eats like child eating*	*roadway to meal*	*diners*	*consume the eating*	*eat it*	*learned to be fat*
elections	*wants like vote party*	*ballot to poll*	*parties*	*expect the ballot*	*vote it*	*held to be next*
sick	*cares like child caring*	*roadway to flu*	*patients*	*afford the caring*	*care it*	*cared to be sick*
sing	*sings like girl singing*	*freeway to song*	*singers*	*perform the singing*	*sing it*	*sung to be loud*

Table 1: Titles produced with the Metre constraints.


Original song: *Highway to Hell* by AC/DC

Youtube: https://www.youtube.com/watch?v=l482T0yNkeo

New title: roadway to meal

eating healthy , eating safe
summer breakfast on an one-way drink
urging reason , sit me be
buying everything in my fruit
do n't eat worry , do n't eat snack
ai n't reason i would rather do
going down , diet food
my guests are gon na be there too
i 'm on the roadway to meal
on the roadway to meal
roadway to meal
i 'm on the roadway to meal

I am familiar with the original song (1 - Strongly disagree, 5 - Strongly agree)

The new lyrics are completely different from the original (1 - Strongly disagree, 5 - Strongly agree)

The text of the new lyrics is grammatical (1 - Strongly disagree, 5 - Strongly agree)

The text of the new lyrics is semantically coherent (1 - Strongly disagree, 5 - Strongly agree)

Out of the following, choose one that could be the topic of the new lyrics:

art computational eat elections love eat sing war none of the previous

The new lyrics are easy to sing with the original melody of the song (1 - Strongly disagree, 5 - Strongly agree)

Overall, I enjoyed the presented lyrics (1 - Strongly disagree, 5 - Strongly agree)

Submit

Figure 9: Example assignment of the evaluation survey.

	Novelty		**Grammar**		**Semantics**		**Topic**		**Singability**		**Overall**	
	Mo	x̃	Mo	x̃	Mo	x̃	Correct	None	Mo	x̃	Mo	x̃
History	5	4	3	3	2	2	66%	13%	1	2	1	2
PoS	5	4	3	3	3	3	68%	12%	3	3	1	2
Metre	3	4	4	3	4	3	56%	13%	4	3	4	3
Rhymes	3	3	3	3	4	4	6%	40%	4	4	3	3

Table 2: Mode and Median of the rating different aspects in lyrics produced with different configurations.

To a large extent, the results of the survey confirm our initial conclusions. Using only the history of replacements leads to higher novelty, because more words are replaced. On the other hand, singability is clearly the lowest, as well as the overall appreciation. This is also the configuration with the least coherent semantics. Though surprising, because words selected this way should be the closest to an actual analogy, this aspect can be indirectly affected by the grammatical issues. When adding the PoS constraint, novelty is comparable, and so is grammaticality, which suggests that it is not clear that considering the PoS improves the syntax. But semantic coherence is better, which can be an indirect consequence of better syntax. Singability is also improved, but overall appreciation is the same as in the previous configuration. Both of these lead to the best proportion of assignments with the correct topic, respectively 66% and 68%.

As expected, introducing the metre constraint leads to lower novelty, lower proportion of assignments with the correct topic, though still positive, and improvements in all the other aspects, visible on the higher mode. Where this configuration stands out, is for having the best overall appreciation. Finally, when adding the rhymes constraint, novelty, grammar and the overall appreciation have a slight decrease, whereas semantics and singability improve. The main drawback of adding this constraint is that the proportion of correct topics is very low, even lower than the random chance (11%). Our interpretation is that this is due to the low number of replaced words, which results in lyrics very close to the original, thus easy to sing and with similar semantic cohesion, but in a very different topic than t_n.

On the familiarity with the songs, both mode and median were always 4 or 5. For the History and Rhymes configuration, six judges answered this question with 1 or 2, and nine for the other two configurations. Yet, if we ignore such answers, the only change is that the mode of the grammar aspect for the History configuration drops to 1.

For a broader idea of the results produced by WAM, figure 10 shows three lyrics for which the mean overall appreciation was 4 or higher, all generated with the Metre configuration. Figure 11 shows three lyrics with overall appreciation between 1 and 2, produced with different configurations. Curiously, although two judges rated the third with 1, another rated it with 4. Despite the rating differences, in the third example of each figure, an issue occurs when replacing the original token *don't*. The tokenizer splits it into *do+n't*, but only *do* is replaced, resulting in odd constructions like *wants n't* and *eats n't*. Lyrics with higher appreciation are slightly further from t_n, but still, to a great extent, semantically coherent. On the other hand, singability of lyrics by the PoS and History is confirmed to be low. For the latter, grammatical issues and strange words (e.g., *wo*) also contribute to the lower rating.

Conclusion

Interesting results were achieved, which confirmed that we can indeed rely on the analogy operation in word embeddings for automatically shifting the meaning of a poem, as long as some constraints are considered. We illustrated the impact of such constraints in WAM, a system that relies on analogy for transforming lyrics according to a new theme. Moreover, towards better appreciation, human opinions confirmed that replacement words should also have the same PoS and metre as the original. WAM can be seen as a fast way of generating parodies, or even advertising campaigns, based on original songs, poetry, or even other kinds of text as well (e.g., news headlines). To fix still existing issues, results may always be further curated.

Although we are generally happy with the results, in the future, additional experiments can still be performed at different levels. For instance, different priorities can be set for different constraints (e.g., if it is not possible to meet them, drop low-priority constraints); a language model can be used for considering the replacement given the previous or next word(s); or alternative analogy solving methods (e.g., (Levy and Goldberg 2014)) can be tested. To access future changes, we could possibly automatise the evaluation of some aspects (e.g., novelty with ROUGE) and follow alternative ways for evaluating others. On the latter, we noticed that, in some cases, the topic was incorrect, but somehow related to t_n (e.g., *sick* instead of *eat*). To minimise this, the topic might become an open answer and we may rely on its similarity with t_n for assessing its suitability.

We should add that, although we worked with English, a similar approach could be followed for transforming lyrics in other languages, as long as there is a model of word embeddings, a list of stopwords, a PoS tagger, and a method of splitting syllables and identifying the stress of words.

References

Agirrezabal, M.; Arrieta, B.; Astigarraga, A.; and Hulden, M. 2013. POS-tag based poetry generation with WordNet. In *Proc. 14th European Workshop on Natural Language Generation*, 162–166.

Bay, B.; Bodily, P.; and Ventura, D. 2017. Text transformation via constraints and word embedding. In *Proc. 8th International Conference on Computational Creativity*, ICCC 2017, 49–56.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full FACE poetry generation. In *Proc. 3rd International Conference on Computational Creativity*, ICCC 2012, 95–102.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Gatti, L.; Özbal, G.; Stock, O.; and Strapparava, C. 2017. Automatic generation of lyrics parodies. In *Proc. 25th ACM international conference on Multimedia*, 485–491.

Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In *Proc. 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2016, 1183–1191.

Gonçalo Oliveira, H., and Oliveira Alves, A. 2016. Poetry from concept maps – yet another adaptation of PoeTryMe's flexible architecture. In *Proc. 7th International Conference on Computational Creativity*, ICCC 2016, 246–253.

Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proc. 10th International Conference on Natural Language Generation*, INLG 2017, 11–20. ACL.

Hämäläinen, M., and Alnajjar, K. 2019a. Generating modern poetry automatically in Finnish. In *Proc. 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP, 6001–6006. ACL.

Hämäläinen, M., and Alnajjar, K. 2019b. Let's FACE it. Finnish poetry generation with aesthetics and framing. In

Title: *Expect the Ballot*

polls like balloting
vote the ballot
go coming in
into my even year
crucial to me
pierce back through me
can't you represent?
oh my even boy
all i ever promised
all i ever needed
is here in my votes
polls are very unnecessary
they can only do threat

Title: *Roadway to Flu*

caring healthy, caring sick
rookie patient on an one-way train
telling reason, need me be
treating everything in my gait
don't care worry, don't care treat
ain't reason i would rather do
going down, labour week
my pets are gonna be there too
i'm on the roadway to flu
on the roadway to flu
roadway to flu
i'm on the roadway to flu

Title: *Vote it*

they said him do n't you ever want around here
don't say to change your fear, you even reappear
the conflict's in their polls and their terms are really stressed
so vote it, now vote it
you even own, you even do what you can
don't say to change no rule, do n't be a midterm role
you say to be strict, even do what you can
so vote it, but you say to be due
now vote it, vote it
no part plans to be elected
showin' how quirky and weak is your war
it wants n't issue who's clear or left
now vote it, vote it

Figure 10: Lyrics with high overall appreciation, all with Metre configuration: *Enjoy the Silence* with $t_n = elections$; *Highway to Hell* with $t_n = sick$; *Beat it* with $t_n = elections$.

Title: *Eat the Eating*

eaten you food
meal the eating
go ate in
into your get consume
bite to really
fish want through really
do wo know learn?
yeah your get boy
all we better liked
all we better need
now here in your meat
eaten they very diets
them if only ca harming

Title: *Election to Referendum*

voting electoral, voting parliamentary
balloting vote on an one-way ballot
urging government, hold me be
following everything in my reshuffle
don't ensure opposition, don't ensure fraud
ain't government i would rather do
going down, candidate month
my elections are gonna be there too
i'm on the election to referendum
on the election to referendum
election to referendum
i'm on the election to referendum

Title: *Eat it*

they ate him don't you ever want around here
don't know to tell your food, you even reappear
the eating's in their lips and their meals are really sure
so eat it, else eat it
you even go, you even do what you can
don't know to tell no flesh, don't be a healthful meal
you know to be mean, even do what you can
so eat it, but you know to be sick
else eat it, eat it
no make does to be revolted
showin' how tasty and good is your feed
it eats n't happen who's whole or wrong
else eat it, eat it

Figure 11: Lyrics with low overall appreciation: *Enjoy the Silence* with $t_n = eat$ and History configuration; *Highway to Hell* with $t_n = election$ and PoS configuration; *Beat it* with $t_n = eat$ and Metre configuration.

Proc. 12th International Conference on Natural Language Generation, INLG 2019. ACL.

Levy, O., and Goldberg, Y. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of 18th Conference on Computational Natural Language Learning*, CoNLL 2014, 171–180. ACL.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The Stanford Corenlp natural language processing toolkit. In *Proc. 52nd annual meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60. ACL.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proc. Workshop track of the International Conference on Learning Representations*, ICLR 2013.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. ACL.

Potash, P.; Romanov, A.; and Rumshisky, A. 2015. Ghost-Writer: Using an LSTM for automatic rap lyric generation. In *Proc. 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2015, 1919–1924. Lisbon, Portugal: ACL.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8):9.

Repar, A.; Martinc, M.; Znidarsic, M.; and Pollak, S. 2018. Bislon: Bisociative slogan generation based on stylistic literary devices. In *Proc. 9th International Conference on Computational Creativity*, ICCC 2018, 248–255. ACC.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *Proc. 3rd International Conference on Computational Creativity*, ICCC 2012, 211–5.

van Stegeren, J., and Theune, M. 2019. Churnalist: Fictional headline generation for context-appropriate flavor text. In *Proc. 10th International Conference on Computational Creativity*, ICCC 2019, 65–73. ACC.

Yan, R. 2016. i, Poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proc. 25th International Joint Conference on Artificial Intelligence*, IJCAI 2016, 2238–2244. New York, NY, USA: IJCAI/AAAI Press.