To Sing like a Mockingbird

Lorenzo Gatti and **Gözde Özbal** and **Oliviero Stock** and **Carlo Strapparava**
FBK-irst, Trento, Italy
l.gatti@fbk.eu, gozbalde@gmail.com, stock@fbk.eu, strappa@fbk.eu

Abstract

Musical parody, i.e. the act of changing the lyrics of an existing and very well-known song, is a commonly used technique for creating catchy advertising tunes and for mocking people or events. Here we describe a system for automatically producing a musical parody, starting from a corpus of songs. The system can automatically identify characterizing words and concepts related to a novel text, which are taken from the daily news. These concepts are then used as seeds to appropriately replace part of the original lyrics of a song, using metrical, rhyming and lexical constraints. Finally, the parody can be sung with a singing speech synthesizer, with no intervention from the user.

It ain't the melodies that're important, man, it's the words.
- Bob Dylan

1 Introduction

Musical parody, "the humorous application of new texts to preexistent vocal pieces" as defined by the Encyclopædia Britannica, is a creative act that is often used in advertising, for its comical results or even for achieving "détournement", i.e. reversing the meaning of a song and turning it against itself.

Take for example the song "Girls" by the Beastie Boys[1], which was used in a 2013 commercial[2] for the company GoldieBlox (that produces toys for girls). This parody modifies the lyrics of the song to promote less "gender-stereotypical" toys. As it often happens in these cases, the video quickly went viral (Fell, 2013). The same song was also covered by a Las Vegas artist[3], who changed just one word in the chorus to "defuse" its sexist lyrics while keeping it extremely recognizable ("Girls, all I really want is girls" becomes "Girls, all *they* really want is girls").

The effectiveness of creative modification, as postulated by the Optimal Innovation Hypothesis (Giora et al., 2004), can only be seen when the object to be modified is well-known to the listener, and for this reason musical parodies are usually based on very popular songs. However, this effect is not limited to lyrics or text, but it is also present when the music itself is modified (e.g. musical mashups, where two songs are combined by blending the music of a song with the vocal track of the other one) and even in the visual domain.

This paper will describe a system for automatically generating musical parodies, starting from a corpus of well-known songs and a novel text, which provides the context for the parody. We take novel, ever-changing texts from daily news feeds. From these, new concepts and words to be inserted in the parody are yielded. Words are replaced in the song according to musical and linguistic constraints, and the new lyrics and the original music are "reassembled". Finally, a singing synthesizer produces the musical realization of the parody.

2 Related Works

Much of lyric writing is technical and it certainly falls under the area of creative writing. Computational linguistics has recently advanced into the field of computational creativity.

Poetry generation systems face similar challenges to ours as they struggle to combine semantic, lexical and phonetic features in a unified framework. Greene et al. (2010) describe a model for poetry generation in which users can control

[1] http://youtu.be/0e8j3-TuzCs
[2] http://youtu.be/M0NoOtaFrEs
[3] http://youtu.be/bRqW4PxipG4

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 298–304,
Valencia, Spain, April 3-7, 2017.

meter and rhyme scheme. Generation is modeled as a cascade of weighted Finite State Transducers that only accept strings conforming to a user-provided desired rhyming and stress scheme. The model is applied to translation, making it possible to generate translations that conform to the desired meter. Toivanen et al. (2012) propose to generate novel poems by replacing words in existing poetry with morphologically compatible words that are semantically related to a target domain. Content control and the inclusion of phonetic features are left as future work and syntactic information is not taken into account.

Recently, some attempt has been made to generate creative sentences for educational and advertising applications. Özbal et al. (2013) propose an extensible framework called BRAINSUP for the generation of creative sentences in which users are able to force several words to appear in the sentences. BRAINSUP makes heavy use of syntactic information to enforce well-formed sentences and to constraint the search for a solution, and provides an extensible framework in which various forms of linguistic creativity can easily be incorporated. The authors evaluate the proposed model on automatic slogan generation.

As a study focusing on the modification of linguistic expressions, the system called Valentino (Guerini et al., 2011) slants existing textual expressions to obtain more positively or negatively valenced versions by using WordNet semantic relations and SentiWordNet (Esuli and Sebastiani, 2006). The slanting is carried out by modifying, adding or deleting single words from existing sentences. Insertion and deletion of words is performed by utilizing Google Web 1T 5-Grams Corpus to extract information about the modifiers of terms based on their part-of-speech. Valentino has also been used to spoof existing ads by exaggerating them, as described in (Gatti et al., 2014), which focuses on creating a graphic rendition of each parodied ad. Lexical substitution has also been commonly used by various studies focusing on humor generation. Stock and Strapparava (2006) generate acronyms based on lexical substitution via semantic field opposition, rhyme, rhythm and semantic relations provided by WordNet. The proposed model is limited to the generation of noun phrases. Valitutti et al. (2009) present an interactive system which generates humorous puns obtained by modifying familiar expressions with word substitution. The modification takes place considering the phonetic distance between the replaced and candidate words, and semantic constraints such as semantic similarity, domain opposition and affective polarity difference. Valitutti et al. (2013) propose an approach based on lexical substitution to introduce adult humor in SMS texts. A "taboo" word is injected in an existing sentence to make it humorous.

As another application of Optimal Innovation Hypotesis, (Gatti et al., 2015) present a system that produces catchy news headlines. The methodology takes existing well-known expressions and innovates them by inserting a novel concept coming from evolving news.

Finally, regarding our specific task of generating song parodies, we notice that in advertising, music is a widely used element to improve the recall of the advertised product, attract the attention of the consumers and aid to convey the message of the advertised product (Heaton and Paris, 2006). (North et al., 2004) demonstrated with their experiments that the recall of a product in a radio advertisement was enhanced by the musical fit, and the recall of the specific product claims could be promoted by the voice fit.

3 Corpus

For this work we used the corpus developed by Strapparava and Mihalcea (Mihalcea and Strapparava, 2012). The corpus contains 100 popular songs (e.g., *Dancing Queen* by ABBA, *Hotel California* by the Eagles, *Alejandro* by Lady Gaga), where the notes of the melody are strictly aligned with the corresponding syllables in the lyrics.

The genres of the songs fall mainly into pop, rock and evergreen. The corpus was built by aligning the melody contained within the MIDI tracks[4] of a song with its lyrics.

In the corpus, several features are present for each song. In the first place, the key of the song (e.g., G major, C minor). At the note level: the time code of the note with respect to the beginning of the song (`time` attribute); the note (`orig-note`) aligned with the corresponding syllable (the content of a `<token>` tag); the distance of the note from the key of the song (`tone`);

[4]The MIDI format does not encode an analog audio signal, but the musical notation of songs: pitch and note length, and other parameters such as volume, vibrato, panning and cues and clock signals to set the tempo.

```
<song filename="AHARDDAY.m2a">
  <key time="0">G major</key>
  <chorus>
    <verse pvers="1">
      <token time="5040" orig-note="B" tone="3" interval="210">IT</token>
      <token time="5050" orig-note="B" tone="3" interval="210">'S </token>
      <token time="5280" orig-note="C'" tone="4" interval="210">BEEN </token>
      <token time="5520" orig-note="B" tone="3" interval="210">A </token>
      <token time="5760" orig-note="D'" tone="5" interval="810">HARD </token>
      <token time="6720" orig-note="D'" tone="5" interval="570">DAY</token>
      <token time="6730" orig-note="D'" tone="5" interval="570">'S </token>
      <token time="7440" orig-note="D'" tone="5" interval="690">NIGHT</token>
    </verse>
    <verse pvers="2">
      <token time="8880" orig-note="C'" tone="4" interval="212">AND </token>
      <token time="9120" orig-note="D'" tone="5" interval="210">I</token>
      <token time="9130" orig-note="D'" tone="5" interval="210">'VE </token>
      <token time="9360" orig-note="C'" tone="4" interval="210">BEEN </token>
      <token time="9600" orig-note="D'" tone="5" interval="210">WOR</token>
      <token time="9840" orig-note="F'" tone="7-" interval="930">KING </token>
      <token time="10800" orig-note="D'" tone="5" interval="210">LI</token>
      <token time="11040" orig-note="C'" tone="4" interval="210">KE </token>
      <token time="11050" orig-note="C'" tone="4" interval="210">A </token>
      <token time="11280" orig-note="D'" tone="5" interval="330">D</token>
      <token time="11640" orig-note="C'" tone="4" interval="90">O</token>
      <token time="11760" orig-note="B" tone="3" interval="330">G</token>
  </verse>
    ...
</song>
```

Figure 1: Two lines of a corpus song: *It's been a hard day-'s night, And I've been wor-king li-ke a d-o-g*

and the duration of the note (`interval`). An example from the corpus, the first two lines from the Beatles' song *A hard day's night*, is shown in Figure 1.

We enriched this annotation by adding new tags (`<bridge>`, `<chorus>`, `<strophe>` and `<other>`) that indicate the various parts of a song, and an attribute (`memorable="true"`) that can be added to any of these parts to signal the "memorable" part of a song (i.e., the part that most people are supposed to quickly recognize). We did this annotation manually for each entry in the corpus, but this step could also be automatized, in case new songs need to be added (Eronen, 2007).

4 Algorithm

The parody generation process is divided into four basic steps: *1)* retrieving the daily news and identifying the most characterizing words of each news piece; *2)* finding new concepts and words evoking the initial text; *3)* generating parodies by replacing words inside the chorus of a song with these concepts, according to musical and linguistic constraints; *4)* producing a final output file for each song, where the words are converted to phonemes and are then aligned with background music from external MIDI files. The files produced by the system are then played with a singing synthesizer, where a virtual voice will actually sing the parody thus created.

***1)* Key concepts from the news** The process starts by downloading the news of the day from important news providers, such as the BBC and the New York Times. Each news article is composed of a headline and a short summary describing its content. Both the headline and the summary are lemmatized and PoS-tagged using the Stanford CoreNLP suite (Manning et al., 2014), which also identifies any named entity present in the text.

The system then discards all the irrelevant tokens and lemmas by removing stop words and keeping only the words that are more characteristic of the specific text, appearing less frequently in a news corpus (Parker et al., 2011). All the named entities are considered relevant, and thus are never removed.

As an example, let us take the headline "Mom protects 2-year-old daughter by biting off dog's ear", where the system will identify the nouns "mom", "dog" and "ear" and the verb "to bite" as characterizing words.

***2)* Search space expansion** To increase the possibilities of finding a match in the third step, the list of key concepts is expanded via WordNet (Fellbaum, 1998), the Oxford Thesaurus (Urdang, 1993) and WikiData (Vrandečić and Krötzsch, 2014).

WordNet is used for finding synonyms and derivationally related forms for lemmas that were found in Step 1. However, words that are too polysemous[5] are not subject to this expansion process, since they might result in unrelated concepts being added to the list. The words thus retrieved are again checked against their probability of being in the news, to discard words that are not specific enough. Similarly, synonyms for each word are obtained through the Oxford thesaurus.

From WikiData the system can extract properties for the named entities found in the article. In particular, it looks for capitals (for countries), countries (for cities or regions) and demonyms (for all the geographical locations), while for people it extracts names, surnames, occupations and fields of work.

Given the previous example, we obtain words such as "mum", "mummy", "mama" (synonyms of "mom"), "hound" (from "dog"), the nouns "chomp" and "bite" and the verbs "to munch" and "to chew" (all from the verb "to bite").

***3)* Assembling the new song** The system then focuses on the most recognizable part of the song. This is usually the chorus (Eronen, 2007), but the XML annotation can indicate otherwise, as stated in Section 3. The goal of this step is replacing words or word sequences, according to various constraints.

Given a word in a song, if the word is at the end of a song line (the last complete word before the `</verse>` tag in the XML file), it will replace it with a related concept only if the concept *i)* rhymes (or is a near-rhyme) with the word; *ii)* it has the same part of speech as the original word; *iii)* they both have the same number of syllables. If the word is in any other position, the rhyme constraint is not enforced. The rhyming information is extracted from the CMU pronunciation dictionary (Rudnicky, 2014).

These constraints are enforced to ensure that the rhythmic properties of the lyrics keep unchanged. In particular, keeping the count of syllables constant means that the synthesizer should be able to sing the word at the same pace of the original, while the rhyme at the end of a song line is maintained to avoid disrupting rhyming with other line endings.

Non-content words are not modified and, when multiple substitutions are possible, the system chooses the one that better fits the context, according to a language model (Brants and Franz, 2006).

For the song in Figure 1 the system would swap "day" with "ear", since they have the same part of speech and the same number of syllables. The word "night" at the end of the first song line would be replaced with "bite", since in this position there is also the rhyming constraint.

***4)* Final output** Finally, once the substitution step is completed, the system needs to output a file that can be opened in Vocaloid (Kenmochi and Ohshita, 2007), a commercial singing synthesizer. To do so, it has to consider, for each word, whether it is all sung on the same note (e.g. "been" or "hard" in Figure 1) or if instead it is split across multiple notes (e.g. "working", which is split across two `<token>` tags, or "dog", which is sung as "d-o-g").

In the first case, nothing has to be done, since Vocaloid will automatically derive the correct pronunciation for the word from its spelling.

For the other case, however, not only is a grapheme-to-phoneme conversion (Black et al., 1998) needed to get the pronunciation of the word, but the system also needs to correctly split the phonemes so they match how graphemes are divided across notes.

Continuing with our example, the word "munching" (that replaces "working") will be converted to "m V n tS I N", i.e. its phonetic representation in the X-SAMPA phonetic alphabet that Vocaloid uses. Then, since "working" was split as "wor" and "king", the system has to divide the pronunciation, so on the first note the synthesizer will sing "m V n", while on the second note it will sing "tS I N".

For every word it also considers the musical features given from the corpus (e.g. pitch and duration), and uses all these to produce an XML output file that can be read in the Vocaloid singing synthesizer. A MIDI track is also added to provide the background instruments.

Once this file is opened in Vocaloid, the parody created by the system can be sung directly or exported to a WAV file.

[5] We defined, empirically, a threshold of 6 senses.

The resulting song can be listened to at `http://youtu.be/jjv0TNFgkoo`.

5 Discussion

Combining language and music is a natural and very popular form of expression. Music fragments tend to be easily recognizable and often it gives pleasure to reproduce them, even reinforcing their memorability. The rhythm and musical constraints associated to the text, make the text itself easy to remember. Popular songs in particular are an excellent candidate for optimal innovation, i.e. changing some minimal elements in the text of the songs so to obtain an evocative effect on some other novel concept, while preserving the pleasure of the recognition and appreciation of the well acquainted song. In fact, this technique is often used for mocking purposes and other entertainment settings, but also in advertisements and other scenarios oriented toward attention grabbing and influencing the attitude of people.

In this paper we have presented a system that applies well-established NLP techniques and rhythm adaptation strategies to the domain of songs, with the aim of minimally changing lyrics to introduce or suggest a new concept, while keeping all the metrical and musical aspects that guarantee that the outcome is still similar to the original song. Minimal changes tend to emphasize the difference and evoke the new concept brought into the song.

An initial evaluation of the system is showing promising results. We asked 3 CrowdFlower annotators to compare 10 parodies with the unmodified songs, both "performed" by Vocaloid, and decide which ones are more grammatical (if any), and whether the parody is more related to the headline from which the key concepts are derived. Finally, we also asked whether the parody was fun. Each song was annotated 3 times, and the ratings were aggregated using majority voting.

It is very interesting to note that the force of music is so strong that small variations that have very good properties of rhyming and rhythm coherence with the original song are often acceptable, even if they do not obey grammatical or semantic constraints. Considering the song we have used throughout Section 4, for example, we have a grammatically correct but semantically invalid replacement when "a hard day's night" becomes "a hard ear's bite", but the evaluation shows that even grammatically incorrect lyrics can be rated as acceptable. More in general, 7 out of 10 modified songs were rated as being as grammatical as the originals. A more complete evaluation could provide insights for determining when to relax correctness in favor of the evocative power of words.

The relatedness ratings confirm the effectiveness of the method for identifying key concepts and expanding them: 9 out of 10 parodies are rated as being more related than the original song, with the remaining one being as related as the original (due to the particular wording of the latter).

Finally, 6 out of 10 parodies were considered fun. While this is still the majority of the parodies, we would like to determine if this percentage can increase when users are only shown parodies of songs that they already know, a condition that we did not test for. A more thorough evaluation, that takes into account this and other problems, is currently in progress. Once completed, we hope to determine whether song parodies can positively influence the recall of news at a later time.

Further enhancements to the system could be developed. For example, in the current version, Vocaloid is used for synthesizing the song with the modified lyrics. However, the "singing" technology is in continuous and fast evolution, and the modularity of the system allows for an easy accommodation of any new synthesizer. For instance, it could be integrated with the state of the art in synthesizers (Bonada et al., 2016b; Bonada et al., 2016a), where the quality of the generated voice is already much higher than the one of Vocaloid. Other developments will include a selection mechanism that, for each news article, selects the best "disruptive" parody.

The results of this work suggest that our system could be used for help in the production of convincing musical parodies. As far as possible applications are concerned, we shall study the adaptation of the system to the advertising domain, where these parodies are commonly used. In this case, we plan to extract properties of the advertised product and use those as concept words for the modification step.

Acknowledgments

This work was partially supported by a Google Digital News Initiative (DNI) grant.

References

Alan W. Black, Kevin Lenzo, and Vincent Pagel. 1998. Issues in building general letter to sound rules. In *Third ESCA/COCOSDA Workshop on Speech Synthesis*, pages 77–80, Blue Mountains, Australia.

Jordi Bonada, Martí Umbert, and Merlijn Blaauw. 2016a. Audio examples for the singing synthesis challenge 2016. Retrieved October 11, 2016 from `http://www.dtic.upf.edu/~jbonada/BonSSChallenge2016.rar`.

Jordi Bonada, Martí Umbert, and Merlijn Blaauw. 2016b. Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016. In *Proceedings of INTERSPEECH 2016: Special Session*, pages 1230–1234, San Francisco, USA.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. Linguistic Data Consortium.

Antti Eronen. 2007. Chorus detection with combined use of mfcc and chroma features and image processing filters. In *Proceedings of the 10th International Conference on Digital Audio Effects*, pages 229–236, Bordeaux, France.

Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC'06*, pages 417–422.

Jason Fell. 2013. Goldieblox video about girls becoming engineers goes viral, 11. Retrieved October 11, 2016 from `https://www.entrepreneur.com/article/230055`.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library, New York, USA.

Lorenzo Gatti, Marco Guerini, Oliviero Stock, and Carlo Strapparava. 2014. Subvertiser: mocking ads through mobile phones. In *Proceedings of IUI'14*, pages 41–44.

Lorenzo Gatti, Gözde Özbal, Marco Guerini, Oliviero Stock, and Carlo Strapparava. 2015. Slogans are not forever: Adapting linguistic expressions to the news. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI-2015)*, Buenos Aires, Argentina, July.

Rachel Giora, Ofer Fein, Ann Kronrod, Idit Elnatan, Noa Shuval, and Adi Zur. 2004. Weapons of mass distraction: Optimal innovation and pleasure ratings. *Metaphor and Symbol*, 19(2):115–141.

Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of EMNLP'10*, pages 524–533.

Marco Guerini, Carlo Strapparava, and Oliviero Stock. 2011. Slanting existing text with Valentino. In *Proceedings of IUI'11*, pages 439–440.

Michelle Heaton and Kelly Paris. 2006. The effects of music congruency and lyrics on advertisement recall. *Journal of Undergraduate Research IX*.

Hideki Kenmochi and Hayato Ohshita. 2007. Vocaloid - commercial singing synthesizer based on sample concatenation. In *Proceedings of INTERSPEECH 2007*, pages 4009–4010, Antwerp, Belgium.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, USA.

Rada Mihalcea and Carlo Strapparava. 2012. Lyrics, music, and emotions. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 590–599, Jeju, Korea, July.

Adrian C. North, Liam C. Mackenzie, Ruth M. Law, and David J. Hargreaves. 2004. The effects of musical and voice "fit on responses to advertisements1. *Journal of Applied Social Psychology*, 34(8):1675–1708.

Gözde Özbal, Daniele Pighin, and Carlo Strapparava. 2013. BRAINSUP: Brainstorming Support for Creative Sentence Generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1446–1455, Sofia, Bulgaria, August. Association for Computational Linguistics.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition. DVD.

Alex Rudnicky. 2014. The cmu pronouncing dictionary, release 0.7b. Retrieved October 11, 2016 from `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

Oliviero Stock and Carlo Strapparava. 2006. Laughing with HAHAcronym, a computational humor system. In *Proceedings of AAAI'06*, pages 1675–1678.

J. M. Toivanen, H. Toivonen, A. Valitutti, and O. Gross. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of ICCC'12*, pages 175–179.

Laurence Urdang. 1993. *The Oxford thesaurus: an AZ dictionary of synonyms*. Clarendon Press, Oxford, UK.

A. Valitutti, C. Strapparava, and O. Stock. 2009. Graphlaugh: a tool for the interactive generation of humorous puns. In *Proceedings of ACII'09 Demo track*, pages 634–636.

Alessandro Valitutti, Hannu Toivonen, Antoine Doucet, and M. Jukka Toivanen. 2013. "Let everything turn well in your wife": Generation of adult humor using lexical constraints. In *Proceedings of ACL'13*, pages 243–248.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.