# Demo Abstract
# LyricalPalette: A Support System for Writing Lyrics Centred on Sentiment

Jasper Pardijs[1] and Davide Ceolin[2]

[1] Vrije Universiteit Amsterdam
[2] Centrum Wiskunde & Informatica

## 1  Introduction

Creating songs can be a difficult process, which holds for both composing music and writing lyrics (Medd, 2012) (Pattison, 1991). Even though these primarily depend on the inspiration of the writer, the purpose of the songwriter is to share his or her emotions (OED Online, n.d.). This abstract presents a demo system called LyricalPalette that aids songwriters in finding their intended sentiment while writing lyrics. The system provides a text box where the writer can write the song lyric. A sentiment analysis is executed on this text when the user presses the corresponding button. The resulting sentiment score ranges from very positive to very negative. The sentiment analyzer is composed of two classifiers trained on movie reviews. Apart from this overall sentiment, suggestions are presented to the writer. These suggestions are synonyms of the words that have a higher importance in determining the overall emotion of the text. The influence on the overall sentiment of these suggestions is depicted alongside. The writer has the option to implement or delete suggestions via the corresponding buttons. The result is a tool that aims to shaping the right emotional tone of the lyric and provide inspiration. The planned user study involves a qualitative analysis of the system with the aim to find out whether the system is effective in reaching this goal.

## 2  Methods and Resources

LyricalPalette is implemented in Python. It makes indirect use of the NLTK Library to assess the sentiment of the lyric analyzed. Via the API of text-processing, a trained sentiment model is called upon that is trained on movie reviews. This API returns the probability of the analyzed text as being positive, negative and neutral. The highest probability and associated label is considered the result. The API documentation can be found at `http://text-processing.com/docs/sentiment.html`. Apart from this resulting overall sentiment, suggestions for the lyric in progress is computed in the following way. Firstly, the lyric is tokenized and stop words are removed. After this, the resulting list of words is ranked based on the strength of their sentiment polarity. The polarity

can either be positive or negative, but the absolute value is used for comparison. A Python package named TextBlob is used for this, available at `http://textblob.readthedocs.io/en/dev/#`. Following, synonyms of the highest ranked words are extracted with the use of the Python module PyDictionary (`https://pypi.python.org/pypi/PyDictionary/1.3.4`). For each of the first 10 synonyms, the overall sentiment of the lyric is calculated by replacing the original word by this synonym and calculating the sentiment. This effect and the synonym suggestion are assigned to the corresponding buttons, usable by the user if desired.

A screencast demonstrating the system functionalities is available at `https://youtu.be/-9zbrz3zbG8`. A screenshot that shows the performance of the system while writing a lyric can be found in Figure 1.
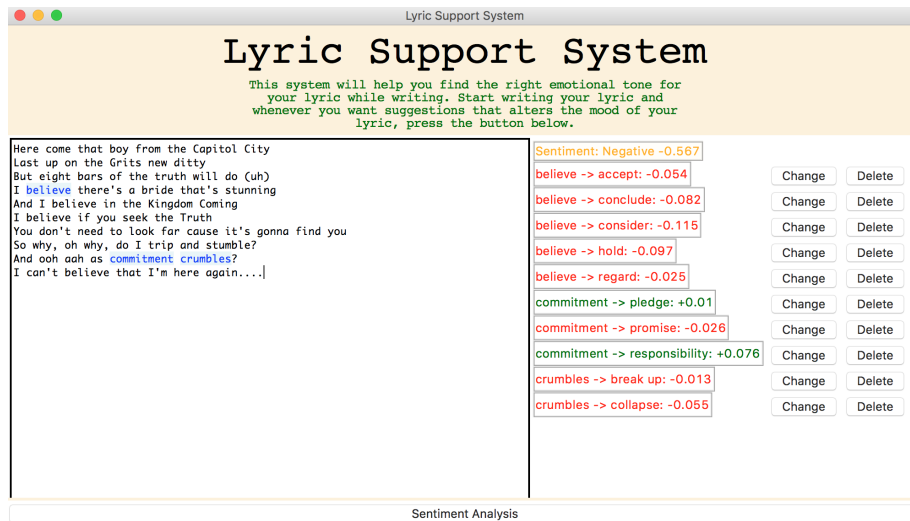
The current source code is available at `https://gist.github.com/JPardijs/31cca8cd5b1458d9dec9cddcd262d88d`.



**Fig. 1.** Screenshot of LyricalPalette.

## References

Medd, J. (2012, May). The rules: Neil hannon on lyric-writing. *The Word*, 46–47. Retrieved from `http://search.proquest.com/docview/1030390091/`

OED Online. (n.d.). *Definition of lyric in english by oxford dictionaries.* Retrieved 2018-02-06, from `https://en.oxforddictionaries.com/definition/lyric`

Pattison, P. (1991). *Songwriting: Essential guide to lyric form and structure: Tools and techniques for writing better lyrics.* Hal Leonard Corporation.