Lyric Sentiment Analysis

Overview:

Sentiment Analysis in general is an automated process of understanding an opinion about a given subject from written or spoken language. It can be used to identify polarity i.e. positive or negative opinion about something or fine-classification like detecting emotion or genres of music.

Algorithms could be:

1. Rule-based, which perform sentiment analysis based on manually crafted rules (e.g.: Give predefined lists of positive and negative words. Count number of positive and negative words in the text and return overall polarity depending on which ones appear more). This approach is pretty much outdated.
2. Automatic, which use machine learning classifiers (positive, negative, neutral etc.). Feature extractor transfers text input into a feature vector. Pairs of feature vectors and tags (e.g. positive, negative, neutral) are fed to the classifier which makes a decision.
3. A combination of the above two.

Good to know:

1. Word embeddings – NLP modelling technique which, crudely speaking, turns text into numbers/vectors which are then passed to a deep learning algorithm. It is a way of building a low dimensional vector representation from text, which preserves contextual similarity of words. Word embeddings are a type of representation that allows words with similar meaning to have a similar representation.
2. N-gram – a contiguous sequence of *n* items from a given sample of text/speech. They could be phonemes/syllables/letters/words etc. N-gram models are used to predict the next item in a sequence in the form of a (n - 1) order Markov model.
3. Bag-of-words model – a way of representing text data when it needs to be sent to ML algorithms (vectors of numbers) i.e. feature extraction. Uses sparse vectors (one-hot vector with 1 where the word occurs and 0 everywhere else). Pretty much outdated.
4. Contiguous BOW – is the same as BOW, but instead of using sparse vectors to represent words, it uses dense vectors (word embeddings).

Current state-of-the-art in sentiment analysis uses RNNs, specifically Long Short-Term Memory (LSTM) models and bi-directional LSTM models. When it comes to understanding the context of a sentence, it becomes crucial to know the whole sentence and not just a few sentiment keywords (e.g. “Teddy Roosevelt was a great president” and “Teddy bears are on sale” have the same word “Teddy” which mean very different things. In view of the same, bi-directional LSTMs use information from both the past and the future to decipher this).

The following paper summarises that LSTMs and bi-directional LSTMs are particularly good at fine-grained sentiment tasks (i.e. classifying into multiple classes like positive, negative, neutral etc.) when compared to CNNs. However, RNNs can look back only a few steps and are thus not able to represent long text sequences well. Also, RNNs are slower to train when compared to CNNs. It also evaluates a few other algorithms on state-of-the-art datasets:
http://www.aclweb.org/anthology/W17-5202

The following is a blog detailing state-of-the-art in text-based sentiment analysis which talks about RNNs, Gated Recurrent Units (GRU), CNNs and also a few non-neural network based methods. It includes links to several state-of-the-art research papers as well: https://blog.paralleldots.com/data-science/breakthrough-research-papers-and-models-for-sentiment-analysis/

Also, the skip-gram model and continuous bag of words model developed in Google based on Recurrent neural network language model made a breakthrough in NLP and seems to be the state-of-the-art. Deep RNNs (as specified in the Coursera deep learning course by Andrew Ng) could also be used to perform sentiment analysis, although they can become computationally very expensive.

When it comes to lyric sentiment analysis in particular, there arise a few problems when compared to general text sentiment analysis, such as:

1. Many words within songs do not contribute a lot to sentiment
2. Not enough information to work with since lyrics can be sparse
3. Negations and modifiers that lie before or after a particular sentiment keyword may change the context
4. Nouns and verbs used to express sentiment are ambiguous
5. N-grams, which are popular for general text analysis, do not usually capture context or poetic expression. (Although N-grams are now being replaced by neural language models)
6. Deep learning is data hungry and there isn't a lot of labelled data for lyric tags (tags from last.fm seem to be used the most in all ongoing research, and lyrics are taken from the million song and musiXmatch datasets).

Also, web scraping, cleaning lyrics and removing annotations like '[intro/verse/chorus], [chorus x2]' etc. need to be performed before working with common web-based lyric hosting websites.

Research in the field:

The following PhD. Thesis by Erion Cano (2018) explores text-based sentiment analysis and music emotion recognition using deep learning techniques. The author presented a couple of neural network architectures for sentiment analysis that are based on CNNs applied on top of pretrained word embedding representations of texts. They follow the same fundamental design of the top performing architectures on image recognition tasks. Based on obtained results, this design appears successful in text analysis as well.
https://arxiv.org/pdf/1810.03031.pdf

The following paper uses the million song dataset to evaluate the performance of a system that only uses lyrics for music mood classification. Songs with negative valence and positive arousal were classified with the highest accuracy (at around 70%).
https://www.insight-centre.org/sites/default/files/publications/15.092_an_exploration_of_mood_classification_in_the_million_songs_dataset.pdf

Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling:
https://www.aclweb.org/anthology/C/C16/C16-1329.pdf

Paper describing the use of Gated Neural Networks for targeted sentiment analysis:
https://pdfs.semanticscholar.org/688d/8718e662d931d8c0ab1bfd03314c2ba711af.pdf?_ga=2.160728829.1346256656.1547054160-1023861626.1541696003

Paper describing the use of recurrent convolutional neural networks for text classification:
https://webcache.googleusercontent.com/search?q=cache:IsFNq0llGBIJ:https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/download/9745/9552+&cd=1&hl=en&ct=clnk&gl=uk

This PhD. Thesis by Xiau Hu (2010) (no deep learning) uses SVMs to show that combining lyric analysis, audio features and social tags improved music mood classification tasks. Complementing audio with lyrics could reduce the number of training examples required to achieve the same performance level as single-source systems. It could also reduce the length of audio required to train the model.
https://core.ac.uk/download/pdf/4826870.pdf

This paper compares the performance of lyric features against audio features to classify music mood. 18 mood categories were tested and lyric features outperformed audio features in 7 of them. Analysing the lyric feature types which outperformed audio features indicated a strong and obvious semantic connections to the categories. No such semantic connection was observed in the case where audio features outperformed lyrics. Lyric-based features performed badly in the negative valence, negative arousal quadrant while audio features performed better here.
http://ismir2010.ismir.net/proceedings/ismir2010-106.pdf

This paper explores classifying songs looking at a combination of lyrics (inferring the mood through valence/arousal model) and basic audio features (no deep learning):
https://pdfs.semanticscholar.org/0def/091627fade56002fa3438fde488e6e6255ea.pdf

Paper describing the use of tf*idf (term frequency-inverse document frequency) based on lyrics to classify music mood:
https://ilk.uvt.nl/downloads/pub/papers/ismir.pdf

Paper (old) describing Sentiment Vector Space Model for lyric-based sentiment classification:
https://aclanthology.info/pdf/P/P08/P08-2034.pdf

Resources:

Deep learning for NLP. Goes into the theory behind CNNs, RNNs, GRUs used for NLP:
http://www.lix.polytechnique.fr/Labo/Antoine.Tixier/dl_nlp_notes.pdf

C code for Contiguous BOW and skip-gram architectures for computing vector representations of words: https://code.google.com/archive/p/word2vec

Python library for unsupervised semantic modelling from plain text:
https://radimrehurek.com/gensim/models/word2vec.html
An interesting blog on lyric sentiment analysis for modern music. Talks about how song writing changed over the years and uses multiple stats and models to compare different artists over time.
https://michaeljohns.github.io/lyrics-lab/

A good summary of sentiment analysis with details about deep learning techniques used to perform state-of-the-art text classification:
https://monkeylearn.com/sentiment-analysis/

Blog explaining sentiment analysis on song lyrics (from web scraping/data collection till sentiment analysis):
https://old.opendatascience.com/blog/sentiment-analysis-on-lyrics/

APIs:

Python APIs (Open source):

1. NLTK and SpaCy (NLP libraries) (https://www.nltk.org/) (https://spacy.io/)
2. Scikit-learn, Keras, PyTorch, TensorFlow (ML libraries)

SaaS APIs:

1. MonkeyLearn (https://monkeylearn.com/)
2. Google Cloud NLP (https://cloud.google.com/natural-language/)
3. Amazon Comprehend (https://aws.amazon.com/comprehend/)
4. Lexalytics (https://www.lexalytics.com/)
5. Aylien (https://aylien.com/)
6. MeaningCloud (https://www.meaningcloud.com/)
7. Rosette (https://www.rosette.com/capability/sentiment-analysis/)