Nostalgic Sentiment Analysis of YouTube Comments for Chart Hits of the 20th Century

Joe Timoney[1], Adarsh Raj, and Brian Davis

[1] Dept. of Computer Science, Maynooth University, Maynooth, Co. Kildare, Ireland

joseph.timoney@mu.ie, adarsh.raj.2018@mumail.ie

Abstract. Examining the comments associated with YouTube postings of songs from the later decades of the 20th century can be fascinating. Many older people express how nostalgic the music might make them feel for that time in their lives, and how it evokes a desire to be young again. It is interesting to understand whether they reflect a social phenomenon only possible through modern technologies. The aim of this paper is to make an initial investigation. YouTube videos for Number 1 songs from the British charts since the 1960's were identified. Their comments were extracted and labelled as being nostalgic or not. Two Machine learning techniques from the GATE tool were applied to the data for different feature sets to find which technique performed best at classifying nostalgia. The results show that, with cross-validation, the Decision Tree Classifier outperformed the Naïve Bayes. Additionally, it is shown that the feature set has an influence on the accuracy.

Keywords: Sentiment analysis, Machine Learning, YouTube API.

1 Introduction and Outline

According to a Guardian article (Lyne, 2016) we are living in an era of nostalgia. Nostalgia is defined to mean a sentimental longing for a period in the past. In particular, the internet has provided us with a highly accessible portal that we can use revisit the past on a whim. Simon Reynolds says that we are experiencing of a "crisis of overdocumentation," facilitated by "YouTube's ever-proliferating labyrinth of collective recollection" and the ever-growing amount of digital music archives (Harvey, 2011). Rather than trying to recall the past through clouded memories now we can vividly rediscover its cultural artefacts at any time when we chose to do so. YouTube in particular has become an 'accidental repository of billions of videos and, more deliberately, a film and video archive' (Soukup, 2014). YouTube is noted for having surprising pieces of rare ephemera. The fact that YouTube users can post comments on the content they engage with means that it is a participatory medium. Thus, it is a social space that fosters community through the users responding to the comments of others (Thelwall, Sud, & Vis, 2012). Looking at postings of popular music from the previous decades it is hard not to notice from the comments that this music is evoking particularly nostalgic reactions. Examples include 'Very fortunate to have been brought up with music of this caliber. I'm an 80's child and my mum had this on vi-

nyl', 'Man, this brings back the days...Adam Ant was so freaking awesome', and 'my favorite Cher song and yes I remember watching this on the Sonny and Cher Comedy Hour....reminds me of my mother and my childhood'. Anecdotally it can often be seen that people reminisce how this music corresponded to the best time in their lives, that they wished to return to that era, and how they feel that this music was so much better than what is released nowadays.

It was from these observations that the motivation for this work arose. It was very much a preliminary study but the desire was to try to discover more formally how accurate this belief was, that is, and how could the level of nostalgia be measured? Given that YouTube has an API, this means that certain information from the website can be legitimately accessed via a developer account. The goal was to extract sets of comments associated with particular music postings and then investigate how machine learning, as survey in (Medhat, Hassan, & Korashy, 2014), could be applied to determine the proportion of comments that were nostalgic. The added value from this could be it might facilitate further sociological investigations into a modern phenomenon (Lariviere, 2017) (Di Placido, 2016) (writeguy4, 2017) and (Davalos, Merchant, Rose, Lessley, & Teredesa, 2015), and be useful for marketing of products with the help of social media (Gross, 2018).

For convenience, and without compromising the goals, the comments from YouTube postings for number one songs were obtained. The song titles were derived from a UK chart archive that has data from 1952 to the present day. A sample of this data could be classified manually as being nostalgic. This would be used to train a suitable machine learning approach, for various parameterizations, to measure how well it performs at identifying nostalgia. The next section provides more details on the data extraction and labeling procedures. The section after this discusses the machine learning approaches of Naïve Bayes and Decision Trees that were selected for this study. The results of applying these classifiers for various features are given in the subsequent section. The final section derives conclusions and proposes avenues for future work.

2 Data gathering Procedure

Before gathering the data from the YouTube it was decided to find the songs which reached number one in the UK charts from the years of 1960 to 1970. These songs were scraped from the official chart UK website (Official Charts, 2018). This was relatively easy to do as the URL for the website has a very regular format. For example, for the chart from the 22nd of September 1966 the URL is:

http://www.officialcharts.com/charts/singles-chart/19660922/7501/

while one week later on the 29th of September is:

http://www.officialcharts.com/charts/singles-chart/19660929/7501/

Thus, the second last directory level is a number in the format of *yearmonthday*. Taking into account that some songs remained at the top for more one week, the total number of songs titles obtained was 250. Following this the YouTube API was used to get the comments the comments from these songs which were then saved in a CSV

file. The YouTube API is available in a number of different languages: Java, PHP or Python. Source Code to assist with this is available at the Google developer website. The videos are extracted by providing the `video_id` to the script. There are certain parameters which are mandatory such as the `part`, which when set to be snippet will return a comma separated list of comments. The `maxResult` gives the maximum number of the comments from a video. The largest value this can be set to is one hundred. The `textFormat` for extracting the comment can be set as plain text. A code fragment is given in Fig. 1 to illustrate this.

```
def get_comment_threads(youtube, video_id):
results = youtube.commentThreads().list(

        part="snippet",

        maxResults=100,

        videoId=videoId,

        textFormat="plainText"

      ).execute()
```

Fig. 1. Code illustrating the YouTube API call for data from a video

2.1 Data Sampling

There were 250 song titles gathered and using each of these 100 comments were extracted from YouTube. A population of 25000 comments was obtained. The large number of the comments extracted from the YouTube data meant it would be very difficult to manually analyze all of them. Thus, a sample was selected from the population. To get an appropriate sample size a satisfactory confidence level was required such that this sample would represent the population correctly (Sample Size Calculator, 2012). The sample of the data from the population was obtained by keeping the confidence level at 95% and the margin of error at 4, leading to total of 556 comments. The sample data was then labelled manually using a flag of true or false to show whether a comment is nostalgic or not. Labelling was done based on the presence of nostalgic keywords in the comment such as ‘memories’, ‘reminds’, and ‘remember my childhood’. Approximately 180 comments out of the total of 556 were labelled as nostalgic. This data was then divided into the training and test sets in the proportion of 4:1.

3 The GATE Tool

To carry out the sentimental analysis the NLP tool chosen was the *General Architecture for Text Engineering* (GATE) tool (Cunningham, Maynard, Bontcheva, & Tablan, 2002). This open source package enables the implementation of various language processing tasks such as information extraction and helps the user to create and annotate the corpora and to perform the evaluation. As a framework GATE is highly customizable. There are three resource types in GATE:

- The Language Resource (LR) which represents the various entities such as the corpora, lexicons or the ontologies.
- The Processing Resource (PR) which represents the various entities which are algorithmic such as the generators, parser, or the n-gram modelers.
- The Visualization Resource (VR) which represents the display and editing components.

Textual input is transformed with the GATE software. This creates a GATE document and includes a Language Resource which will contain the input text together with one or more sets of annotations. Annotations are generally updated by algorithms manipulating the document during text analysis.

Various processing resources available in GATE and most important is the tokenizer (Saggion & Funk, 2009), which segments the text of the document in units representing words, punctuation, and other elements. GATE produces a token annotation for each word in the document. Tokens' features computed during this process are their type (word, punctuation, number, space, control character, etc.), their lengths, and their orthographic characteristics (all capitals, all lowercase, capital initial, and so on).

3.1 Using the GATE tool

To carry out the analysis the data is stored as separate training and test sets in a formatted xml file that GATE can read. This is called the corpus. The comments are pre-processed in GATE in order to label the data with the annotation of being nostalgic or non-nostalgic by giving the *instance* as comments, *class* as the rating and the *attributes* as 'True' or 'False'. This will then be used by the machine learning algorithm. An example of the xml is shown in Figure 2

```
<?xml version="1.0" encoding="UTF-8"?>
<dataSet>
<comment nostalgic ="false">He was great why does every-
thing need to be about black and white.  </comment>
```

```
<comment nostalgic ="true">truly you are the wonder of
you, thanks elvis today is your 40th anniversary 2017.
again, thank you for the wonderful memories. </comment>
</dataSet>
```

Fig. 2. Example showing the xml formatting

4 Classification Approaches

Two classification approaches were investigated in this study: Naïve Bayes and Decision Trees (Bishop, 2006). Additional work using Support Vector Machines has been carried out but is not published yet (Raj, 2018).

4.1 Naïve Bayes classifier

Naïve Bayes is a classifier that uses Bayes theorem. Bayes theorem. The implementation of the Naïve Bayes classifier assumes that the data instances are conditionally independent in order to compute the MAP hypothesis. It is a well-known, straightforward algorithm and is not as computationally demanding as other approaches.

4.2 Decision Tree Classifier

Decision trees are a widely used algorithm in machine learning since they can be adapted easily to any type of the data. The algorithm is mainly used when there is a need for many hierarchical distinctions. The tree itself is represented as linear structure and can be easily understood. The decision tree consists of a root node, representing the entire data, and decision nodes and leaf nodes which illustrate the classification. The data is passed through the tree to classify the instance. At each of the decision nodes a certain feature from the input is compared with a constant that is recognized during the training phase. The data will pass through all these decision nodes until it reaches a leaf node that represents the particular assigned class

4.3 The Cross-validation technique

A cross validation evaluation is implemented for the sample data. Cross validation is the best way to stretch the validity of the manually annotated data since it enables it to be tested on a large number of the documents. It is very good for checking model effectiveness especially when there is a need to mitigate for overfitting. In *k*-fold cross validation, the data is divided into *k* different subsets. One of the *k* subsets is used as the test data and the remaining data *k*-1 are taken as the training data. The overall error obtained as the average of all the *k* trials and is a measure of the total effectiveness of the model. Every data point gets to be in a validation set exactly once, and gets to be in a training set *k*-1 times. This significantly reduces bias as most of the

data is used for fitting and lowers the variance as most of the data is also being used in validation set. In this work *k* is chosen to be 5.

4.4 Evaluation metric

The Accuracy (%) is the ratio of correctly predicted observations and is formulated using the number of True positives (*TP*), True negatives (*TN*), False Positives (*FP*) and False negatives (*FN*),

$$Accuracy = 100 \times \frac{TP + TN}{TP + TN + FP + FN}$$

5 Results

5.1 Cross validation using Naïve Bayes with different features

Different features from the tokens like string + unigram, root + unigram, root + string + unigram, root + category + unigram, length + unigram, Root + Orth + Unigram, Root + Orth + Category were selected. These features are described by (Saggion & Funk, 2009):

- string: the original, unmodified text of the token.
- unigram: a sequence of tokens is represented as an n-gram with a unigram meaning a sequence of length 1
- Root: the lemmatised, lower-case form of the token (for example, run is the root feature for run, runs, ran, and Running).
- Orth: a code representing the token's combination of upper- and lower-case letters (if it has been classified as a word).
- Category: the part-of-speech (POS) tag, a symbol that represents a grammatical category such as determiner, present-tense verb, past-tense verb, singular noun.

The Naïve Bayes algorithm was applied onto the sampled data. Additionally, the *k*-fold Cross validation technique (*k*=5) was employed. The unigram feature is always assumed. From Table 1, the first result obtained is for the feature of string, indicating that we are using all the token string in the test comments. It gives a good result with an Accuracy value of 0.79. This means that the 79% of the comments are correctly predicted as being either nostalgic or non-nostalgic, the remaining 21% being classified incorrectly. When the feature is selected as the length of the token, the Accuracy obtained is only 0.64 which means that the feature length of the token is not as good a predictor of whether the comment is nostalgic or not. When the feature is selected as the root of the token, the Accuracy obtained is 78.15%. This is not as good as the string feature. Next the feature of string and the root of the tokens are selected. The Accuracy obtained is 77.8%, that is the percentage of the comments that are correctly classified, with the rest of the comments being misclassified. If the feature category and the root of the tokens are selected as the features the Accuracy obtained is 77.6%. When the selected features are the Orth and the root of the Token the accuracy ob-

tained is 78.51%. Lastly, if the features selected are the Orth, root and the category of the Tokens the accuracy obtained is 77.01%.

Table 1. Naïve Bayes Cross-Validation Result with different Token Attributes

Token Attribute	Accuracy (%)
String	79
Length	64
Root	78.15
Root and String	77.8
Root and Category	77.61
Root and Orth	78.51
Root, Orth and Category	77.01

Overall, using cross-validation with the Naïve Bayes the results are in the range of 75-80% accuracy. Only the feature of length gave a significantly poorer 64% accuracy.

5.2 Cross validation using Decision Tree with different features

For all the Decision Tree evaluations the default values of two parameters available in GATE were used with *maxDepth*(*m*)=5 and *minInfoGainSplit*(*i*)=0.005. Again, the unigram feature is always assumed. From Table 2 the first result is obtained when the feature is the token string from all the test comments. It gives a good result with an Accuracy of 82.49%. A lower value for Accuracy of 70% is obtained by changing the feature to be the length of the token. When the feature is selected as the root of the token, the Accuracy obtained is 86%. Following this, when the features of string and the root of the tokens are selected as the features and the accuracy obtained is 84.11%. If the feature selected are the Orth and the root of the tokens the Accuracy obtained 86.09%. Lastly, selecting the features of Category, Root and the Orth of the Tokens leads to an Accuracy of 84.83%.

From Table 2 it can be observed that that from all the results for cross-validation using a Decision Tree every value is above 80% except for the feature of the length of the token.

Table 2. Decision Tree Cross-Validation Result with different Token Attributes

Token Attribute	Accuracy (%)
String	82.49
Length	70
Root	86
Root and String	84.11
Root and Category	84.83
Root and Orth	86.09
Root, Orth and Category	84.83

6 Conclusions and Future Work

This intention here was to make a preliminary examination of the application of Machine learning techniques to comments gathered from music videos on YouTube to determine how well they could identify nostalgic comments. Two machine learning algorithms were tested: Naïve Bayes and Decision Trees. Additionally, 5-fold cross-validation was used. The analysis was implemented using the GATE tool. The results showed that in the case of Naïve Bayes the best feature was String and the highest accuracy obtained was 79%. In contrast, the best features for the Decision Tree approach was Root and Orthography, which gave a higher accuracy of 86.09%.

Future work will extend this to try more classifiers. In particular, the results for the Support Vector Machine (SVM) for various parameter values will be examined (Raj, 2018). Other techniques such as Deep learning could be investigated. More data should also be gathered from more recent decades to discover how long the nostalgic behavior lasts for, that is, does it disappear at the turn of the century or is it still reflected in the opinions of music fans for the music of the 'noughties'.

7 References

Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Ney York: Spring-Verlag.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: an Architecture for Development of Robust HLT Applications. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.* Philadelphia, Pennsylvania.

Davalos, S., Merchant, A., Rose, G., Lessley, B., & Teredesa, A. (2015, Nov.). 'The good old days': An examination of nostalgia in Facebook posts. *International Journal of Human-Computer Studies, 83*, 83-93.

Di Placido, D. (2016, December). *2016: The Year We Hit Peak Nostalgia*. Retrieved from forbes.com: https://www.forbes.com/sites/danidiplacido/2016/12/30/2016-the-year-of-nostalgia/#4ec8a72f7aec

Gross, J. (2018, May). *Use Nostalgia To Improve Your Marketing Results*. Retrieved from forbes.com: https://www.forbes.com/sites/forbesagencycouncil/2018/05/24/use-nostalgia-to-improve-your-marketing-results/#5d8dc85862b9

Harvey, E. (2011). 'Retromania': Why Is Pop Culture Addicted to Its Own Past? *The Atlantic*.

Lariviere, C. (2017, March). *Nostalgia in the Age of Social Media: Identity, Meaning & Connection*. Retrieved from medium.com: https://medium.com/@christinelariviere/nostalgia-in-the-age-of-social-media-identity-meaning-connection-da9f31d2c413

Lyne, C. (2016, July 9). How nostalgia took over the world (and why that's no bad thing). *The Guardian*. The Guardian.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal, 5*(4), 1093-1113.

Official Charts. (2018). (The Official UK Charts Company) Retrieved from http://www.officialcharts.com/

Raj, A. (2018). *Sentiment Analysis of the Nostalgic Comments on the songs of 20th Century from YouTube.* Maynooth University, Computer Science. Maynooth: Unpublished MSc thesis.

Saggion, H., & Funk, A. (2009). Extracting Opinions and Facts for Business Intelligence. *RNTI: Revista Negócios e Tecnologia da Informaçao, 17*, 119-146.

Sample Size Calculator. (2012). (Creative Research Systems) Retrieved from https://www.surveysystem.com/sscalc.htm

Soukup, P. (2014). Looking at, with, and through YouTube. *Communication Research Trends, 33*(3), 3-36.

Thelwall, M., Sud, P., & Vis, F. (2012). Commenting on YouTube videos: From guatemalan rock to El Big Bang. *Journal of the American Society for Information Science and Technology, 63*(3), 616-629.

writeguy4. (2017, July). *The Future of Nostalgia: Social Media's Impact on "The Good Old Days"*. Retrieved from wordpress: https://writeguyink.wordpress.com/2017/07/12/the-future-of-nostalgia-social-medias-impact-on-the-good-old-days/