# Comparing Automated Methods to Detect Explicit Content in Song Lyrics

**Michael Fell, Elena Cabrio, Michele Corazza, Fabien Gandon**
Université Côte d'Azur, CNRS, Inria, I3S, France
{firstname.lastname}@inria.fr

## Abstract

The Parental Advisory Label (PAL) is a warning label that is placed on audio recordings in recognition of profanity or inappropriate references, with the intention of alerting parents of material potentially unsuitable for children. Since 2015, digital providers – such as iTunes, Spotify, Amazon Music and Deezer – also follow PAL guidelines and tag such tracks as "explicit". Nowadays, such labelling is carried out mainly manually on voluntary basis, with the drawbacks of being time consuming and therefore costly, error prone and partly a subjective task. In this paper, we compare automated methods ranging from dictionary-based lookup to state-of-the-art deep neural networks to automatically detect explicit contents in English lyrics. We show that more complex models perform only slightly better on this task, and relying on a qualitative analysis of the data, we discuss the inherent hardness and subjectivity of the task.

## 1 Introduction

All content is not always appropriate for all ages and music is no exception. Content industries have been actively searching for means to help adults determine what is and is not appropriate for children. In USA, in 1985, the Recording Industry Association of America (RIAA) introduced the Parental Advisory label (PAL) in order to alert parents of content unsuitable for children because of profanity or inappropriate references[1]. PAL is "a notice to consumers that recordings identified by this mark may contain strong language or depictions of violence, sex or substance abuse"[2] and that parental discretion is advised. In UK, the British Phonographic Industry (BPI) adds to this list "racist, homophobic, misogynistic or other discriminatory language or behavior; or dangerous or criminal behavior"[3].

In the case of a song, the explicit logo is applied when the lyrics or content of a song matches one of these criteria, raising the problem of detecting and labelling explicit songs in a scalable way.

Within the Natural Language Processing (NLP) community, there have been several efforts to deal with the problem of online abusive language detection, since the computational analysis of language can be used to quickly identify offenses and ease the removal of abusive messages. Several workshops (Park and Fung, 2017; Fišer et al., 2018) and evaluation campaigns (Fersini et al., 2018; Bosco et al., 2018; Wiegand et al., 2018) have been recently organized to discuss existing approaches to abusive language detection, propose shared tasks and foster the development of benchmarks for system evaluation. These have led to the creation of a number of datasets for abusive language detection in different languages, that have been shared within the NLP research community. The SemEval 2019 tasks HatEval (Basile et al., 2019) and OffensEval (Zampieri et al., 2019) have aimed at the multilingual detection of hate speech against women or immigrants and the categorization of hate speech, respectively.

In this direction, and given the similarity with the abusive language detection task, this paper addresses the problem of explicit content detection in song lyrics as a binary classification task: a song can be labelled either as explicit or clean (=not explicit). To this end, we first compare a range of classification methods for the task of explicit lyrics detection, from dictionary lookup to deep neural networks. We then attempt the comparison to the

[1] Parental Advisory https://en.wikipedia.org/wiki/Parental_Advisory

[2] RIAA PAL https://www.riaa.com/resources-learning/pal-standards/

[3] BPI Parent Advisory https://www.bpi.co.uk/media/1047/parental-advisory-guidelines.pdf



*Proceedings of Recent Advances in Natural Language Processing*, pages 338–344,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_039

available related works and shed light on the inherent hardness and subjectivity of the task at hand.

The paper is organized as follows: in Section 2 we survey the state of the art in explicit lyrics detection. In Sections 3 and 4 we introduce the classification methods we apply, and the comparative experimentation. Conclusions end the paper.

*NOTE*: This paper contains examples of language which may be offensive to some readers. They do not represent the views of the authors.

## 2 Related Work

Only a few works on the problem of explicit lyrics detection exist. (Bergelid, 2018) consider a dataset of English lyrics (see Table 1, B18) to which they apply classical machine learning algorithms such as Support Vector Machine (SVM) and Random Forest (RF). As features they extract either (i) tf-idf weighted bag-of-word (BOW) representations of each song text or (ii) represent the lyrics with paragraph vectors (Le and Mikolov, 2014). The explicit labels are obtained from Soundtrack Your Brand [4]. They find the RF with tf-idf BOW to perform best, especially in combination with a random undersampling strategy to the highly imbalanced dataset. They also experiment with adding lyrics metadata to the feature set, such as the artist name, the release year, the music energy level, and the valence/positiveness of a song. This results in marginal improvements for some of their models.

(Chin et al., 2018) apply explicit lyrics detection to Korean song texts. They also use tf-idf weighted BOW as lyrics representation and aggregate multiple decision trees via boosting and bagging to classify the lyrics for explicit content. On their corpus (see Figure 1, C18) they report 78% $F_1$ using the bagging method. Note, that bagging with decision trees is similar to the Random Forest method used by (Bergelid, 2018). Interestingly, they also report a baseline for dictionary lookup, i.e. given a profanity dictionary the song text is classified as explicit if and only if one of its words occurs in the profanity dictionary. With such a baseline they obtain 61% $F_1$.

More recently, (Kim and Mun, 2019) proposed a method to create explicit words dictionaries automatically by weighting a vocabulary according to all words' frequencies in the explicit class vs. the clean class, accordingly. For instance the word "fuck" is typical for explicit lyrics and atypical for clean lyrics. They compare different methods to generate such a lexicon. The achieved performances using solely dictionary lookup range from 49% $F_1$ for a man-made dictionary to 75.6% $F_1$ when using relative class frequencies. Note, that the latter performance is achieved with a dictionary of only 25 words. They work with a corpus of Korean lyrics (see Figure 1, K19). Unlike previous work, they apply a recursive neural network, resulting in 76.6% $F_1$, slightly higher than the simple dictionary lookup. They find performance to increase to 78.1% when combining the vector representation of the RNN with a one-hot vector indicating for each profane word from the dictionary if the lyric contains it. They argue to use the RNN to find such cases where the expliciteness arises from the context and not from a dictionary check. However, no examples of finding this phenomenon are presented.

## 3 Methods for Explicit Lyrics Detection

In this work, we compare a range of classification methods for the task of explicit lyrics detection. Common to all methods is that they classify a full song into one of two mutually exclusive classes - explicit or clean (=not explicit). This means, the decision if a song text is explicit is taken globally. We assess the performance of different classification methods ranging from simple dictionary lookup / lexicon checking to general purpose deep learning language understanding models. We try to identify contextual effects by applying a method that outputs the "importance" for each word (see Section 3.4).

### 3.1 Dictionary-Based Methods

The most straightforward way to implement an automated explicit content detection method, is checking against a dictionary of explicit words. The dictionary can be man-made or automatically created from example explicit and clean lyrics. Then, a classifier uses this dictionary to predict the class of an unseen song text.

#### 3.1.1 Dictionary Creation

It is possible to use handcrafted dictionaries such as Noswearing [5]. Performance using an automatically created lexicon has previously been shown (Kim and Mun, 2019) to improve over the manually created dictionary. We therefore consider only

[4] https://www.soundtrackyourbrand.com

[5] https://www.noswearing.com/

the case of the machine-made dictionary in this work. We generate a dictionary of words that are indicative of explicit lyrics. We define the importance $I$ of a word $w$ for explicit lyrics by the frequency $f(w, ex)$ of $w$ in explicit lyrics compared to its frequency $f(w, cl)$ in clean lyrics:

$$I(w) = f(w, ex)/f(w, cl)$$

We filter out unique and too common words and restrict the number of terms to 1,000 to avoid over-reliance on terms that are very corpus specific. The dictionary $D_n$ of the $n$ words most important for explicit lyrics, is now straightforwardly defined as containing the $n$ words with the highest $I$ score.

#### 3.1.2 Dictionary Lookup

Given a dictionary $D_n$, this method simply checks if a song text $S$ contains any of the explicit terms defined in $D_n$. Then, $S$ is classified as explicit iff it contains at least one explicit term from $D_n$.

#### 3.1.3 Dictionary Regression

This method uses BOW made from $D_n$ as the feature set of a classifier. We used a logistic regression, but RF or SVM have been used alike in (Bergelid, 2018).

### 3.2 Tf-idf BOW Regression

Similar to the Dictionary Regression, but the BOW contains the whole vocabulary of a training sample instead of only the explicit terms. The word features are weighted with the well-known tf-idf weighting scheme.

### 3.3 Transformer Language Model

Recently, approaches based on self-attention (Vaswani et al., 2017) have been proposed and have proven effective for natural language understanding tasks. These models are structured as an encoder-decoder, and they are trained on unsupervised tasks (such as masked language modelling) in order to learn dense representations of sentences or documents. These models differ from more traditional recurrent neural networks in different aspects. In particular, while recurrent models can process sequences (in NLP, typically word embeddings) in order, transformers use a joint model of the right and left context of each word in order to encode an entire sequence or document. Additionally, transformers are typically less computationally expensive than recurrent models, especially when trained on a GPU accelerator.

One of the most successful transformer-based models proposed in the last few years is BERT (Devlin et al., 2018). This model is composed of multiple transformers connected by residual connections. Pre-trained models are provided by the authors, and they are used in our work to perform explicit language detection in lyrics, without retraining the full model.

### 3.4 Textual Deconvolution Saliency

We use the Textual Deconvolution Saliency (TDS) model of (Vanni et al., 2018), which is a Convolutional Neural Network (CNN) for text classification. It is a simple model containing an embedding layer for word representations, a convolutional layer with max pooling and two fully connected layers. The interesting part about this model is that they manage to reverse the convolution. Given the learned feature map (the output of the convolution before max pooling) of the CNN, they upsample it to obtain a 3-dimensional sample with dimensions (#words, embedding size, #filters). The TDS for each word is now defined as the sum along the embedding axes of the output of the deconvolution. The TDS represents the importance of each word of the input with respect to the learned feature maps. We use this model with the goal to find local explanations for the global decision of the classification as explicit or clean. Such explanations can arise from contexts or phrases that the model assigns a high importance.

## 4 Experimental Setting and Evaluation

We compare the different methods as introduced in the previous section to the task of explicit lyrics detection. We attempt a comparison to the related work as well, although due to different datasets comparing the reported scores directly is problematic. We finally analyze the classification qualitatively with examples, and demonstrate the intrinsic hardness and subjectivity of the explicit lyrics detection task.

**Abbreviations used**: to refer to related works in Table 1 and 3, we use the following abbreviations. B18 stands for (Bergelid, 2018), C18 is (Chin et al., 2018), K19 means (Kim and Mun, 2019), while Ours is this work.

### 4.1 Dataset

The WASABI database (Meseguer-Brocal et al., 2017) contains song-wise labels for explicit lyrics,

| *Work* | *total* | *explicit* | *ratio* | *language* |
|---|---|---|---|---|
| B18 | 25,441 | 3,310 | 13.0% | English |
| C18 | 27,695 | 1,024 | 3.7% | Korean |
| K19 | 70,077 | 7,468 | 10.7% | Korean |
| WAS | 179,391 | 17,808 | 9.9% | English |

Table 1: Overview of our dataset WAS (# songs) and comparison to the related works.

such as explicit, unknown, no advice available, or clean (=not explicit). These labels are provided by the music streaming service Deezer [6]. We selected a subset of English song texts from the corpus which are tagged as either explicit or clean. We filtered out duplicate lyrics and such that contain less than 10 tokens. Finally, our dataset (WAS) comprises of 179k lyrics, with a ratio of explicit lyrics of 9.9%. The details and comparison with related works datasets are depicted in Table 1.

For training any of the models described in the previous section, we once randomly split the data into training-development-test sets with the common 60%-20%-20% ratio. We tuned the hyperparameters of the different classification algorithms on the development set to then test with the best performing parameters on the test set. As evaluation metrics we use precision (*P*), recall (*R*), and f-score ($F_1$). Unless stated otherwise, the scores are macro-averaged over the two possible classes.

### 4.2 Hyperparameters

For the dictionary-based methods, we found the ideal dictionary size to be 32 words for the lookup and 128 words for the regression. The Tf-idf BOW regression performed best when the full vocabulary of unigrams and bigrams was used. We used the sklearn implementation of logistic regression with the class weighting scheme 'balanced' to account for the class imbalance in the dataset. We used TDS with max sequence length 512 and dropout probability 50%. As is the default with TDS, corpus-specific word vectors were trained using Word2Vec (Mikolov et al., 2013) with dimensionality 128. The BERT model comes pretrained and no further pre-training was performed. We used the smaller of the two published models. BERT then was finetuned to our task using max sequence length 256 and batch size 16, otherwise default parameters for text classification task learning.

[6]https://www.deezer.com

### 4.3 Results

Overall, the results of the different classification methods we tried are all close to each other. The simple dictionary lookup with 32 words performs comparably to the deep neural network with 110M parameters (BERT base model). As baseline, we include the majority class classifier that always predicts the clean class. Furthermore, all related works show similar tendencies of performance on their respective datasets. The results of all the different methods we applied are depicted in Table 2 and described in the following.

The majority class classifier delivers a performance of 47.4% $F_1$, which is the only outlier in the sense that this is far below any other model. The dictionary lookup with a vocabulary of the 32 most indicative explicit words obtains a balanced performance as precision and recall are close to each other, the overall performance is 77.3% $F_1$. The dictionary regression performs somewhat better in terms of f-score (78.5% $F_1$), achieving this with the highest overall recall of 81.5%, but it has lower precision. The tf-idf BOW regression performs very similarly to the dictionary regression. This proves that a limited number of words influences the overall performance of the models, and that they do not need to consider the whole vocabulary, just the most offensive words. The increased vocabulary of 929k unigrams and bigrams is gigantic compared to the explicit words dictionary (32 words). As most of these n-grams may be noise to the classifier, this could explain the slight decrease in performance over the dictionary regression. Finally, the neural-network-based methods behave a bit differently: the BERT language model is clearly better in precision (84.4%) over all other models - the second best is TDS with 81.2%. However, BERT performs the worst in recall with only 73.7%. The overall performance of BERT is average with 77.7% $F_1$. Finally, TDS performs best in terms of 79.6% $F_1$. We tested if TDS outperforming BERT was due to TDS using domain-specific word vectors trained on our corpus (BERT is trained on books and Wikipedia). This was not the case as TDS performed almost identically, when using generic word vectors (GloVe, 200d): 80.4% *P*, 78.7% *R*, 79.5% $F_1$.

A closer look at the classification performance shows that the $F_1$ scores for the minority class (explicit lyrics) is highest with TDS (63%) and lowest with the dictionary lookup (58.9%). The majority

| *Model* | *P* | *R* | $F_1$ |
|---|---|---|---|
| Majority Class | 45.0 | 50.0 | 47.4 |
| Dictionary Lookup | 78.3 | 76.4 | 77.3 |
| Dictionary Regression | 76.2 | 81.5 | 78.5 |
| Tf-idf BOW Regression | 75.6 | 81.2 | 78.0 |
| TDS Deconvolution | 81.2 | 78.2 | 79.6 |
| BERT Language Model | 84.4 | 73.7 | 77.7 |

Table 2: Performances of our different models on the WAS dataset. Values in percent.

| *Work* | *Model* | $F_1$ |
|---|---|---|
| **Ours** | Dictionary Lookup | 77.3 |
| **Ours** | Dictionary Regression | 78.5 |
| C18 | Man-made Dictionary | 61.0 |
| K19 | Man-made Dictionary | 49.0 |
| K19 | Dictionary Lookup | 75.6 |
| **Ours** | Tf-idf BOW Regression | 78.0 |
| C18 | Tf-idf BOW | 78.0 |
| C18 | Tf-idf BOW+ | 80.0 |
| B18 | Tf-idf BOW | 67.5 |
| B18 | Tf-idf BOW+ | 82.6 |
| **Ours** | TDS Deconvolution | 79.6 |
| **Ours** | BERT Language Model | 77.7 |
| K19 | HAN | 76.7 |
| K19 | HAN + Dictionary | 78.1 |

Table 3: Performances of dictionary-based methods (top), tf-idf BOW models (middle) and deep models (below). Note that different works use different datasets. Ours always uses the WAS dataset. Values in percent.

class (clean lyrics) on the other hand is best detected by BERT (96.3% $F_1$) and worst with the tf-idf BOW (95.1% $F_1$).

We attempt a comparison of the different approaches used in the different related works as well as ours. While the scores achieved (see Table 3) are not strictly comparable, we can see clear tendencies. According to K19, a man-made dictionary is inferior to an automatically generated one. This is supported by the man-made lexicon in C18 performing subpar to their tf-idf BOW. An appropriate lexicon of explicit terms, on the other hand, can compete with a tf-idf BOW model, as we showed with both the dictionary lookup and the regression performance. This is further supported by the generated dictionary of K19 which competes with the deep HAN model. Optimizations to the standard tf-idf BOW models are marked with the + sign. Restricting the POS tags to more likely ones found in explicit terms (C18) improves performance slighly. Using random undersampling to fight the imbalanced class problem (B18) increases performance drastically, however makes the problem somewhat different from the imbalanced problem. The final takeaway is that deep models do not necessarily outperform shallow models. Neither HAN, TDS, nor BERT deliver much higher scores than the dictionary-based or the BOW method.

### 4.4 Qualitative Analysis

In this section we analyze examples of explicit content lyrics and point to the inherent hardness and subjectivity in classifying and even labelling such data.

#### 4.4.1 Explicitness in Context?

The highest difference in model performance we measured between the deep TDS model (79.6% $F_1$) and the dictionary lookup (77.3% $F_1$). We analyzed why the TDS method performed better than the dictionary lookup by inspecting those examples that (i) were explicit, (ii) were tagged as clean by the dictionary lookup, and (iii) were detected as explicit by TDS with high confidence. [7]

From the 13 examples analyzed, we found three main phenomena: (1) Four texts contained explicit terms that were not contained in the dictionary of explicit terms. Words such as *f**kin', motherf**kers* were too rare to be included in the generated lexicon and other words like *fucking, cunt, cum, shit* were not uniquely contained in explicit lyrics. The reason why this is the case can be traced back to problems in the annotations or the fact that these words are relatively frequently used in lyrics. (2) Five texts whose explicitness arises in context rather than on a word level. Examples with violent context found were "organization with horns of satan performs the ancient rituals" or "bombin on mc's, crushin crews with ease". There were also instances of sexual content such as "give it to him down in the parking lot in the backseat, in the backseat of the car". Note that the words {give, it, to, him} in isolation do not belong to an explicit terms list and the sexuality arises from the context. Similarly in "(turn the lights on) so i can see that ass work". Also here, putting "ass" in an explicit terms dictionary is tempting but may not be ideal,

[7] The last layer of TDS outputs probabilities for the input text being explicit or clean. We looked at examples where the explicit class was predicted with at least 80% probability.

as its meaning is not necessarily explicit. (3) Four texts appeared to have been mislabelled since no explicitness could be found. We found for three of them that the album the song is contained in is tagged as explicit. In cases as these, inheriting the label from the album is wrong, but it seems this is exactly what had happened here. In one Raggae lyric, in particular, we found no explicit content, so we suspect the song was mislabelled.

Since we found some annotation to be problematic, we will discuss difficulties that arise from annotating explicitness in lyrics.

#### 4.4.2 How Hard is this Task?

As stated in the introduction, the explicit label is voluntary and we will argue that it is also somewhat subjective in its nature. There are lyrics which are not tagged as explicit although they have profanity in them. Consider for example the song *Bitch* by Meredith Brooks. While it already contains profanity in the title, it does not carry the explicit label and one can argue that in the context of the song, the term "bitch" is used as a contrastive term and to raise attention to the struggle the songwriter sees in her life, torn between potentially conflicting expectations of society ("I'm a little bit of everything - All rolled into one - I'm a bitch, I'm a lover - I'm a child, I'm a mother - I'm a sinner, I'm a saint - I do not feel ashamed").

Another example is *Check Your Head* by Buckcherry where it says "Ooh and you still bitch about your payments" where "bitch" is used as a verb and one can argue that the acceptance in this verb form is higher than in the noun form. A similar case where the part of speech influences the perceived level of profanity is *Hail Hail Rock 'n' Roll* by Discipline. It contains the line "the band starts to play loud as fuck".

We encounter a different kind of problem when dealing with substance abuse or other drug-related content. It is evident that the legal status of the substances mentioned plays a major role in how such content is labelled. This is further complicated by the fact that legislation about substances can vary wildly between different countries. The labels applied to this content are not culture-invariant, and furthermore changes in the societal view can lead to labels that are not relevant anymore. This, like other examples, shows why the labels applied to lyrics are subject to change in different cultures and time periods.

Another aspect that is very sensitive to time periods and cultures comes from words themselves: an inoffensive word can become offensive in slang or common language. One such example can be found in Johnny Cash's *The Christmas Guest*: "When the cock was crowing the night away - The Lord appeared in a dream to me". Here, cock means male chicken, as opposed to the offensive meaning that is now arguably more common.

We finally want to raise attention to the problem of genre confounding. We found that the genre *Hip Hop* contributed by far the most to all explicit lyrics - 33% of all *Hip Hop* lyrics. Since only about 5% of the whole corpus are tagged as *Hip Hop*, this genre is highly overrepresented. This raises the question in how far our task is confounded with genre classification. When inspecting the explicit terms dictionaries we have created, we clearly see that genre bias reflected. The dictionary of 32 terms that we used for the dictionary lookup method consists approximately half of terms that are quite specific to the Rap genre, such as glock, gat, clip (gun-related), thug, beef, gangsta, pimp, blunt (crime and drugs). Finally, the terms holla, homie, and rapper are arguably no causes for explicit lyrics, but highly correlated with explicit content lyrics. Biasing an explicit lyrics detection model away from genres is an interesting future direction of work.

## 5 Conclusion

Classifying song lyrics as explicit or clean is an inherently hard task to accomplish since what is considered offensive strongly depends on cultural aspects that can change over time. We showed that shallow models solely based on a dictionary of profane words achieve a performance comparable to deep neural networks. We argued that even the hand-labelling is highly subjective, making it problematic to automatically detect if a song text should be tagged as explicit or clean.

We propose as a possible simplification and objectification to study the local detection of explicit content. If we present an authority a report on found trigger words, found contextual sexual content, and alike, they can come to their own subjective conclusion about the final label of the text.

## Acknowledgement

This work is partly funded by the French Research National Agency (ANR) under the WASABI project (contract ANR-16-CE23-0017-01).

## References


Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

Linn Bergelid. 2018. Classification of explicit music content using lyrics and music metadata.

Cristina Bosco, Felice Dell'Orletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. 2018. Overview of the EVALITA 2018 hate speech detection task. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy.*

Hyojin Chin, Jayong Kim, Yoonjong Kim, Jinseop Shin, and Mun Y Yi. 2018. Explicit content detection in music lyrics using machine learning. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 517–521. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. In *IberEval@SEPLN*, volume 2150 of *CEUR Workshop Proceedings*, pages 214–228. CEUR-WS.org.

Darja Fišer, Ruihong Huang, Vinodkumar Prabhakaran, Rob Voigt, Zeerak Waseem, and Jacqueline Wernimont. 2018. Proceedings of the 2nd workshop on abusive language online (alw2). In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Association for Computational Linguistics.

Jayong Kim and Y Yi Mun. 2019. A hybrid modeling approach for an automated lyrics-rating system for adolescents. In *European Conference on Information Retrieval*, pages 779–786. Springer.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Gabriel Meseguer-Brocal, Geoffroy Peeters, Guillaume Pellerin, Michel Buffa, Elena Cabrio, Catherine Faron Zucker, Alain Giboin, Isabelle Mirbel, Romain Hennequin, Manuel Moussallam, Francesco Piccoli, and Thomas Fillon. 2017. WASABI: a Two Million Song Database Project with Audio and Cultural Metadata plus WebAudio enhanced Client Applications. In *Web Audio Conference 2017 – Collaborative Audio #WAC2017*, London, United Kingdom. Queen Mary University of London.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45. Association for Computational Linguistics.

Laurent Vanni, Mélanie Ducoffe, Carlos Aguilar, Frederic Precioso, and Damon Mayaffre. 2018. Textual deconvolution saliency (tds): a deep tool box for linguistic analysis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 548–557.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *CoRR*, abs/1903.08983.