Song Lyrics Summarization Inspired by Audio Thumbnailing

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

Abstract

Given the peculiar structure of songs, applying generic text summarization methods to lyrics can lead to the generation of highly redundant and incoherent text. In this paper, we propose to enhance state-of-the-art text summarization approaches with a method inspired by audio thumbnailing. Instead of searching for the thumbnail clues in the audio of the song, we identify equivalent clues in the lyrics. We then show how these summaries that take into account the audio nature of the lyrics outperform the generic methods according to both an automatic evaluation and human judgments.

1 Introduction

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning of a text (Allahyari et al., 2017). Numerous approaches have been developed to address this task and applied widely in various domains including news articles (Cheng and Lapata, 2016), scientific papers (Mei and Zhai, 2008), web content as blogs (Hu et al., 2007), customer reviews (Pecar, 2018) and social media messages (He and Duan, 2018). Just as we may need to summarize a story, we may also need to summarize song lyrics, for instance to produce adequate snippets for a search engine dedicated to an online song collection or for music digital libraries. From a linguistic point of view however, lyrics are a very peculiar genre of document and generic summarization methods may not be appropriate when the input for summarization comes from a specific domain or type of genre as songs are (Nenkova et al., 2011). Compared to news documents, for instance, lyrics have a very different structure. Given the repeating forms, peculiar structure (e.g. the segmentation into verse, chorus, etc.) and other unique characteristics of song lyrics, we need the summarization algorithms to take advantage of these additional elements to more accurately identify relevant information in song lyrics. But just as such characteristics enable the exploration of new approaches, other characteristics make the application of summarization algorithms very challenging, as the presence of repeated lines, the discourse structure that strongly depends on the interrelation of music and words in the melody composition, the heterogeneity of musical genres each featuring peculiar styles and wording (Brackett, 1995), and simply the fact that not all songs tell a story.

In this direction, this paper focuses on the following research questions: *What is the impact of the context in summarizing song lyrics?*. This question is broken down into two sub questions: 1) *How do generic text summarization methods perform over lyrics?* and 2) *Can such peculiar context be leveraged to identify relevant sentences to improve song text summarization?* To answer our research questions, we experiment with generic unsupervised state-of-the-art text summarization methods (i.e. TextRank, and a topic distribution based method) to perform lyrics summarization, and show that adding contextual information helps such models to produce better summaries. Specifically, we enhance text summarization approaches with a method inspired by audio thumbnailing techniques, that leverages the repetitive structure of song texts to improve summaries. We show how summaries that take into account the audio nature of the lyrics outperform the generic methods according to both an automatic evaluation over 50k lyrics, and judgments of 26 human subjects.

In the following, Section 2 reports on related work. Section 3 presents the lyrics summarization task and the proposed methods. Sections 4 and 5 report on the experiments and on the evaluation, respectively. Section 6 concludes the paper.

Proceedings of Recent Advances in Natural Language Processing, pages 328–337,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_038

2 Summarization Methods

This section reports on the related work on both text and audio summarization methods.

2.1 Text Summarization

In the literature, there are two different families of approaches for automatic text summarization: extraction and abstraction (Allahyari et al., 2017). *Extractive summarization methods* identify important elements of the text and generate them verbatim (they depend only on extraction of sentences or words from the original text). In contrast, *abstractive summarization methods* interpret and examine the text to generate a new shorter text that conveys the most critical information from the original text. Even though summaries created by humans are usually not extractive, most of the summarization research has focused on extractive methods. Purely extractive summaries often give better results (Nallapati et al., 2016), due to the fact that latter methods cope with more complex problems such as semantic representation, inference and natural language generation. Existing abstractive summarizers often rely on an extractive pre-processing component to produce the abstract of the text (Berg-Kirkpatrick et al., 2011; Knight and Marcu, 2000). Consequently, in this paper we focus on extractive summarization methods, also given the fact that lyrics *i)* strongly use figurative language which makes abstractive summarization even more challenging; and *ii)* the choice of the words by the composer may also have an importance for capturing the style of the song.

In the following, we focus on *unsupervised* methods for text summarization, the ones targeted in our study (no available gold-standard of human-produced summaries of song texts exists). Most methods have in common the process for summary generation: given a text, the importance of each sentence of that text is determined. Then, the sentences with highest importance are selected to form a summary. The ways different summarizers determine the importance of each sentence may differ: *Statistics-based summarizers* extract indicator features from each sentence, e.g. (Fattah and Ren, 2009) use among others the sentence position and length and named entities as features. *Topic-based summarizers* aim to represent each sentence by its underlying topics. For instance, (Hennig, 2009) apply Probabilistic Latent Semantic Analysis, while Latent Dirichlet Allocation is used in (Arora and Ravindran, 2008) to model each sentence's distribution over latent topics. Another type of summarization methods is *graph-based summarizers*. Three of the most popular graph-based summarizers are TextRank (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004), and (Parveen et al., 2015). These methods work by constructing a graph whose nodes are sentences and whose graph edge weights are sentence similarities. Then, the sentences that are central to the graph are found by computing the PageRank (Page et al., 1999). Contrarily to all previously described methods, systems using *supervised machine learning* form another type of summarizers. For instance, (Fattah, 2014) treats extractive summarization as a binary classification task, where they extract indicator features from sentences of gold summaries and learn to detect the sentences that should be included in a summary.

Context-Specific Summarization. If specific knowledge about the application scenario or the domain of the summarized text is available, generic summarization methods can be adapted to take into account the prior information. In query-based summarization (Otterbacher et al., 2005; Wang et al., 2016), the user's query is taken into account when generating a summary. Summarization of a scientific paper can be improved by considering the citations of it, as in (Delort et al., 2003). However, to the best of our knowledge no summarization methods have been proposed for the domain of song texts. In this paper we present a summarization method that uses prior knowledge about the text it summarizes to help generic summarizers generate better summaries.

Evaluation Criteria and Methods. Summaries should *i)* contain the most important information from input documents, *ii)* not contain redundant information, *iii)* be readable, hence they should be grammatical and coherent (Parveen and Strube, 2015). While a multitude of methods to identify important sentences has been described above, several approaches aim to make summaries less redundant and more coherent. The simplest way to evaluate summaries is to let humans assess the quality, but this is extremely expensive. The factors that humans must consider when giving scores to each candidate summary are grammaticality, non redundancy, integration of most important pieces of information, structure and coherence

(Saggion and Poibeau, 2013). The more common way is to let humans generate possibly multiple summaries for a text and then automatically assess how close a machine-made summary is to the human gold summaries computing ROUGE scores (Lin, 2004), which boils down to measuring n-gram overlaps between gold summaries and automatic summary. More recently there have been attempts to rate summaries automatically without the need for gold summaries (Nenkova et al., 2011). The key idea is that a summary should be similar to the original text in regard to characteristic criteria as the word distribution. (Mackie et al., 2014) find that topic words are a suitable metric to automatically evaluate micro blog summaries.

2.2 Audio Summarization

Lyrics are texts that accompany music. Therefore, it is worthwhile to see if methods in audio summarization can be transferred to lyrics summarization. In audio summarization the goal is to find the most representative parts in a song, in Pop songs those are usually the chorus and the bridge, in instrumental music the main theme. The task of creating short audio summaries is also known as audio thumbnailing (Bartsch and Wakefield, 2005; Chai and Vercoe, 2003; Levy et al., 2006), as the goal is to produce a short representation of the music that fits onto a thumbnail, but still covers the most representative parts of it. In a recent approach of audio thumbnailing (Jiang and Müller, 2015), the authors generate a *Double Thumbnail* from a musical piece by finding the two most representative parts in it. For this, they search for candidate musical segments in an a priori unsegmented song. Candidate musical segments are defined as sequences of music that more or less exactly repeat themselves. The representativeness of each candidate segment to the whole piece is then estimated by their fitness metric. They define the fitness of a segment as a trade-off between how exactly a part is repeated and how much of the whole piece is covered by all repetitions of that segment. Then, the audio segments along with their fitness allow them to create an audio double thumbnail consisting of the two fittest audio segments.

3 Lyrics Summarization

Song texts are arranged in segments and lines. For instance the song text depicted in Figure 1 consists of 8 segments and 38 lines. Given a song text S consisting of n lines of text, $S = (x_1, ..., x_n)$, we define the task of *extractive lyrics summarization* as the task of producing a concise summary sum of the song text, consisting of a subset of the original text lines: $sum(S) \subseteq S$, where usually $|sum(S)| << |S|$. We define the goal of a summary as to preserve key information and the overall meaning of a song text. To address this task, we apply the following methods from the literature: the popular graph-based summarizer TextRank; an adaptation of a topic-based method (TopSum). Moreover, we introduce a method inspired by audio thumbnailing (which we dub Lyrics Thumbnail) which aims at creating a summary from the most representative parts of the original song text. While for TextRank we rely on the off-the-shelf implementation of (Barrios et al., 2016), in the following we describe the other two methods.

3.1 TopSum

We implement a simple topic-based summarization model that aims to construct a summary whose topic distribution is as similar as possible to that of the original text. Following (Kleedorfer et al., 2008), we train a topic model by factorizing a tf-idf-weighted term-document matrix of a song text corpus (see Section 4.2) using non-negative matrix factorization into a term-topic and a topic-document matrix. Given the learnt term-topic matrix, we compute a topic vector t for each new document (song text). In order to treat t as a (pseudo-) probability distribution over latent topics t_i, we normalize t by applying $\lambda t.t/\sum_{t_i \in t} t_i$ to it. Given the distributions over latent topics for each song text, we then incrementally construct a summary by greedily adding one line from the original text at a time (same mechanism as in KLSum algorithm in (Haghighi and Vanderwende, 2009)); that line x^* of the original text that minimizes the distance between the topic distribution t_S of the original text S and the topic distribution of the incremental summary $sum(S)$:

$$x^* = \underset{x \in (S \setminus sum(S))}{\operatorname{argmin}} \{W(t_S,\ t_{sum(S)+x})\}$$

W is the Wasserstein distance (Villani, 2008) and is used to measure the distance between two probability distributions (an alternative to Jensen-Shannon divergence (Louis and Nenkova, 2013)).

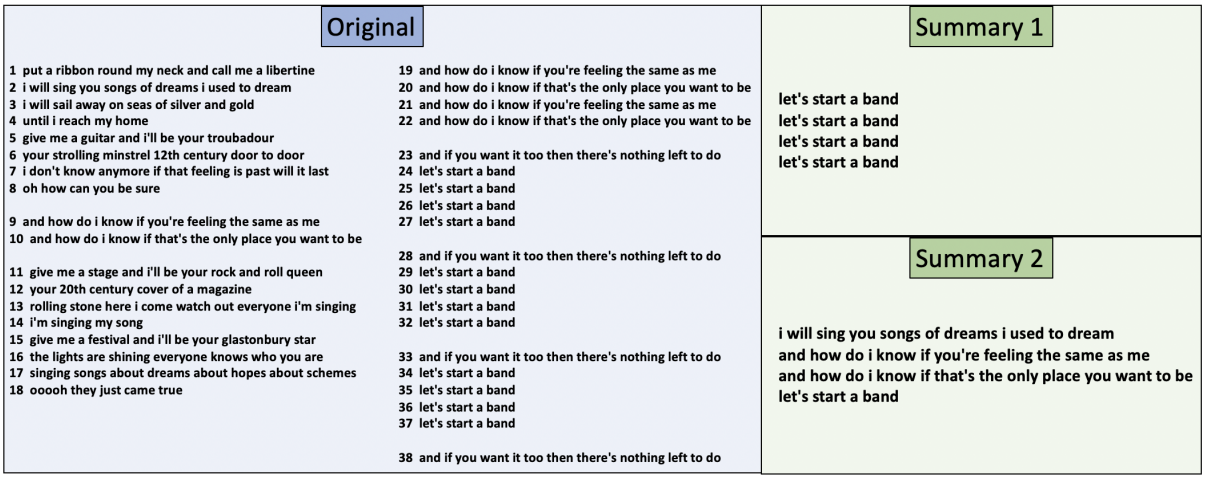

Figure 1: Song text of "Let's start a band" by Amy MacDonald along with two example summaries.

3.2 Lyrics Thumbnail

Inspired by (Jiang and Müller, 2015), we transfer their fitness measure for audio segments to compute the fitness of lyrics segments. Analog to an audio thumbnail, we define a Lyrics Thumbnail as the most representative and repetitive part of the song text. Consequently, it usually consists of (a part of) the chorus. In our corpus the segments are annotated (as double line breaks in the lyrics), so unlike in audio thumbnailing, we do not have to induce segments, but rather measure their fitness. In the following, we describe the fitness measure for lyrics segments and how we use this to produce a summary of the lyrics.

Lyrics Fitness Given a segmented song text $S = (S_1, ..., S_m)$ consisting of text segments S_i, where each S_i consists of $|S_i|$ text lines, we cluster the S_i into partitions of similar segments. For instance, the lyrics in Figure 1 consists of 8 segments and 38 lines and the cluster of chorus consists of $\{S_5, S_6, S_7\}$. The fitness Fit of the segment cluster $C \subseteq S$ is defined through the precision pr of the cluster and the coverage co of the cluster. pr describes how similar the segments in C are to each other while co is the relative amount of lyrics lines covered by C:

$$pr(C) = (\sum_{\substack{S_i,S_j \in C \\ i<j}} 1)^{-1} \cdot \sum_{\substack{S_i,S_j \in C \\ i<j}} sim(S_i, S_j)$$

$$co(C) = (\sum_{S_i \in S} |S_i|)^{-1} \cdot \sum_{S_i \in C} |S_i|$$

where sim is a normalized similarity measure between text segments. Fit is the harmonic mean between pr and co. The fitness of a segment S_i is defined as the fitness of the cluster to which S_i belongs:

$$\forall S_i \in C : Fit(S_i) = Fit(C) = 2\frac{pr(C) \cdot co(C)}{pr(C) + co(C)}$$

For lyrics segments without repetition the fitness is defined as zero. Based on the fitness Fit for segments, we define a fitness measure for a text line x. This allows us to compute the fitness of arbitrary summaries (with no or unknown segmentation). If the text line x occurs $f_i(x)$ times in text segment S_i, then its line fitness fit is defined as:

$$fit(x) = (\sum_{S_i \in S} f_i(x))^{-1} \cdot \sum_{S_i \in S} f_i(x) \cdot Fit(S_i)$$

Fitness-Based Summary Analog to (Jiang and Müller, 2015)'s audio thumbnails, we create fitness-based summaries for a song text. A *Lyrics Double Thumbnail* consists of two segments: one from the fittest segment cluster (usually the chorus), and one from the second fittest segment cluster (usually the bridge).[1] If the second fittest cluster has a fitness of 0, we generate a *Lyrics Single Thumbnail* solely from the fittest cluster (usually the chorus). If the thumbnail generated has a length of k lines and we want to produce a summary of $p < k$ lines, we select the p lines in the middle of the thumbnail following (Chai and Vercoe, 2003)'s "Section-transition Strategy" that

[1] We pick the first occurring representative of the segment cluster. Which segment to pick from the cluster is a potential question for future work.

they find to capture the "hook" of the music more likely.[2]

4 Experimental Setting

We now describe the WASABI dataset of song lyrics (Section 4.1), and the tested configurations of the summarization methods (Section 4.2).

4.1 Dataset

From the WASABI corpus (Meseguer-Brocal et al., 2017) we select a subset of 190k unique song texts with available genre information. As the corpus has spurious genres (416 different ones), we focus on the 10 most frequent ones in order to evaluate our methods dependent on the genre. We add 2 additional genres from the underrepresented Rap field (Southern Hip Hop and Gangsta Rap). The dataset contains 95k song lyrics.

To define the length of $sum(S)$ (see Section 3), we rely on (Bartsch and Wakefield, 2005) that recommend to create audio thumbnails of the median length of the chorus on the whole corpus. We therefore estimate the median chorus length on our corpus by computing a Lyrics Single Thumbnail on each text, and we find the median chorus length to be 4 lines. Hence, we decide to generate summaries of such length for all lyrics and all summarization models to exclude the length bias in the methods comparison[3]. As the length of the lyrics thumbnail is lower-bounded by the length of the chorus in the song text, we keep only those lyrics with an estimated chorus length of at least 4. The final corpus of 12 genres consists of 50k lyrics with the following genre distribution: Rock: 8.4k, Country: 8.3k, Alternative Rock: 6.6k, Pop: 6.9k, R&B: 5.2k, Indie Rock: 4.4k, Hip Hop: 4.2k, Hard Rock: 2.4k, Punk Rock: 2k, Folk: 1.7k, Southern Hip Hop: 281, Gangsta Rap: 185.

4.2 Models and Configurations

We create summaries using the three summarization methods described in Section 3, i.e. a graph-based (TextRank), a topic-based (TopSum), and fitness-based (Lyrics Thumbnail) method, plus two additional combined models (described below). While the Lyrics Thumbnail is generated from the full segment structure of the lyrics including its duplicate lines, all other models are fed with unique text lines as input (i.e. rendundant lines are deleted). This is done to produce less redundant summaries, given that for instance, TextRank scores each duplicate line the same, hence it may create summaries with all identical lines. TopSum can suffer from a similar shortcoming: if there is a duplicate line close to the ideal topic distribution, adding that line again will let the incremental summary under construction stay close to the ideal topic distribution. All models were instructed to produce summaries of 4 lines, as this is the estimated median chorus length in our corpus (see Section 4.1). The summary lines were arranged in the same order they appear in the original text.[4] We use the TextRank implementation[5] of (Barrios et al., 2016) without removing stop words (lyrics lines in input can be quite short, therefore we avoid losing all content of the line if removing stop words). The topic model for TopSum is built using non-negative matrix factorization with scikit-learn[6] (Pedregosa et al., 2011) for 30 topics on the full corpus of 190k lyrics.[7] For the topical distance, we only consider the distance between the 3 most relevant topics in the original text, following the intuition that one song text usually covers only a small amount of topics. The Lyrics Thumbnail is computed using String-based distance between text segments to facilitate clustering. This similarity has been shown in (Watanabe et al., 2016) to indicate segment borders successfully. In our implementation, segments are clustered using the DBSCAN (Ester et al., 1996) algorithm.[8] We also produce two summaries by combining TextRank + TopSum and TextRank + TopSum + Lyrics Thumbnail, to test if summaries can benefit from the complementary perspectives the three different summarization methods take.

Model Combination For any lyrics line, we can obtain a score from each of the applied methods. TextRank provides a score for each line, TopSum provides a distance between the topic distributions of an incremental summary and the original text, and fit provides the fitness of each line. We treat our summarization methods as blackboxes and use a simple method to combine the scores the different methods provide for each line. Given the

[2] They also experiment with other methods to create a thumbnail, such as section initial or section ending.

[3] We leave the study of other measures to estimate the summary length to future work.

[4] In case of repeated parts, the first position of each line was used as original position.

[5] https://github.com/summanlp/textrank

[6] https://scikit-learn.org

[7] loss='kullback-leibler'

[8] eps=0.3, min_samples=2

original text separated into lines $S = (x_1, ..., x_n)$, a summary is constructed by greedily adding one line x^* at a time to the incremental summary $sum(S) \subseteq S$ such that the sum of normalized ranks of all scores is minimal:

$$x^* = \text{argmin} \bigcup_x \{\sum_A \text{R}_\text{A}(x)\}$$

Here $x \in (S \setminus sum(S))$ and $A \in \{\text{TextRank}, \text{TopSum}, \text{fit}\}$. The normalized rank $\text{R}_\text{A}(x)$ of the score that method A assigns to line x is computed as follows: first, the highest scores[9] are assigned rank 0, the second highest scores get rank 1, and so forth. Then the ranks are linearly scaled to the [0,1] interval, so each sum of ranks $\sum_A \text{R}_\text{A}(x)$ is in [0,3].

Model Nomenclature For abbreviation, we call the TextRank model henceforth M_r, the TopSum model M_s, the fitness-based summarizer M_f, model combinations M_{rs} and M_{rsf}, respectively.

5 Evaluation

We evaluate the quality of the produced lyrics summary both soliciting human judgments on the goodness and utility of a given summary (Section 5.1), and through an automatic evaluation of the summarization methods (Section 5.2) to provide a comprehensive evaluation.

5.1 Human Evaluation

We performed human evaluation of the different summarization methods introduced before by asking participants to rate the different summaries presented to them by specifying their agreement / disagreement according to the following standard criteria (Parveen and Strube, 2015):

Informativeness: The summary contains the main points of the original song text.

Non-redundancy: The summary does not contain duplicate or redundant information.

Coherence: The summary is fluent to read and grammatically correct.

Plus one additional criterion coming from our definition of the lyrics summarization task:

Meaning: The summary preserves the meaning of the original song text.

An experimental psychologist expert in Human Computer Interaction advised us in defining the questionnaire and setting up the experiment. 26 participants - 12 nationalities, 18 men, 8 women, aged from 21 to 59 - were taking a questionnaire (Google Forms), consisting of rating 30 items with respect to the criteria defined before on a Likert scale from 1 (low) to 5 (high). Each participant was presented with 5 different summaries - each produced by one of the previously described summarization models - for 6 different song texts. Participants were given example ratings for the different criteria in order to familiarize them with the procedure. Then, for each song text, the original song text along with its 5 summaries were presented in random order and had to be rated according to the above criteria. For the criterion of Meaning, we asked participants to give a short explanation in free text for their score. The selected 6 song texts[10] have a minimum and a median chorus length of 4 lines and are from different genres, i.e. Pop/Rock (4), Folk (1) and Rap (1), similar to our corpus genre distribution. Song texts were selected from different lengths (18-63 lines), genders of singer (3 male, 3 female), topics (family, life, drugs, relationship, depression), and mood (depressive, angry, hopeful, optimistic, energetic). The artist name and song title were not shown to the participants.

Results Figure 2 shows the ratings obtained for each criterion. We examine the significant differences between the models performances by performing a paired two-tailed t-test. The significance levels are: $0.05^*, 0.01^{**}, 0.001^{***}$, and $n.s.$ First, Informativeness and Meaning are rated higher** for the combined model M_{rs} compared to the single models M_r and M_s. Combining all three models improves the summaries further: both for Informativeness and Meaning the model M_{rsf} is rated higher*** than M_{rs}. Further, summaries created by M_{rsf} are rated higher*** in Coherence than summaries from any other model - except from M_f ($n.s.$ difference). Summaries are rated on the same level ($n.s.$ differences) for Non-redundancy in all but the M_r and M_f summaries, which are perceived as lower*** in Non-redundancy than all others. Note, how the model M_{rsf} is more stable than all others by exhibiting lower standard deviations in all criteria except

[9] In the case of topical distance, a "higher score" means a lower value.

[10] "Pills N Potions" by Nicki Minaj, "Hurt" by Nine Inch Nails, "Real to me" by Brian McFadden, "Somebody That I Used To Know" by Gotye, "Receive" by Alanis Morissette, "Let's Start A Band" by Amy MacDonald

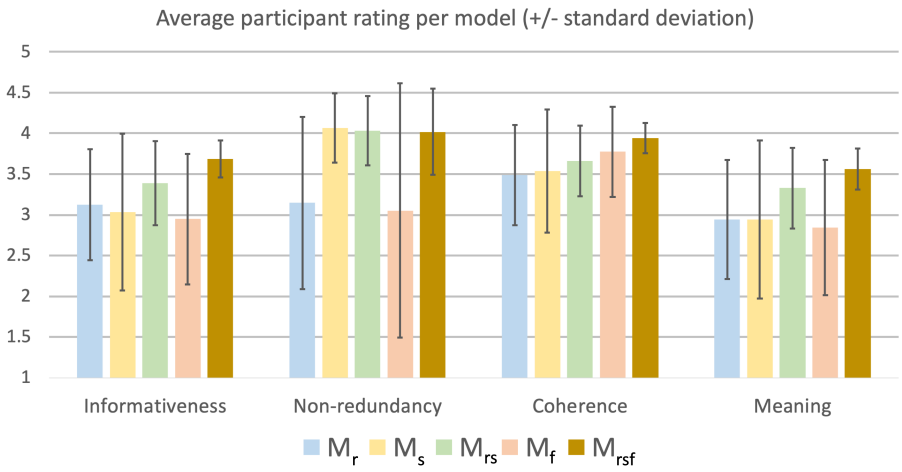

Figure 2: Human ratings per summarization model in terms of average and standard deviation.

Non-redundancy. The criteria Informativeness and Meaning are highly correlated (Pearson correlation coefficient 0.84). Correlations between other criteria range between 0.29 and 0.51.

Overall, leveraging the Lyrics Fitness in a song text summary improves summary quality. Especially with respect to the criteria that, we believe, indicate the summary quality the most - Informativeness and Meaning - the M_{rsf} method is significantly better performing and more consistent.

Figure 1 shows an example song text and example summaries from the experiment. Summary 1 is generated by M_f and consists of the chorus. Summary 2 is made by the method M_{rsf} and has relevant parts of the verses and the chorus, and was rated much higher in Informativeness and Meaning. We analyzed the free text written by the participants to comment on the Meaning criterion, but no relevant additional information was provided (the participants mainly summarized their ratings).

5.2 Automatic Evaluation

We computed four different indicators of summary quality on the dataset of 50k songs described in Section 4.1. Three of the criteria use the similarity between probability distributions P, Q, which means we compute the Wasserstein distance between P and Q (cf. Section 3.1) and apply $\lambda x.\ x^{-1}$ to it.[11] The criteria are:

Distributional Semantics: similarity between the word distributions of original and summary, cf. (Louis and Nenkova, 2013). We give results relative to the similarity of the best performing model (=100%).

[11]This works as we always deal with distances > 0.

Topical: similarity between the topic distributions of original and summary. Restricted to the 3 most relevant topics of the original song text. We give results relative to the similarity of the best performing model (=100%).

Coherence: average similarity between word distributions in consecutive sentences of the summary, cf. (ShafieiBavani et al., 2018). We give results relative to the coherence of the original song text (=100%).

Lyrics fitness: average line-based fitness fit (cf. Section 3) of the lines in the summary. We give results relative to the Lyrics fitness of the original song text (=100%).

Results When evaluating each of the 12 genres, we found two clusters of genres to behave very similarly. Therefore, we report the results for these two groups: the *Rap* genre cluster contains Hip Hop, Southern Hip Hop, and Gangsta Rap. The *Rock / Pop* cluster contains the 9 other genres. Results of the different automatic evaluation metrics are shown in Table 1. Distributional Semantics metrics have previously been shown (Louis and Nenkova, 2013; ShafieiBavani et al., 2018) to highly correlate with user responsiveness judgments. We would expect correlations of this metric with Informativeness or Meaning criteria therefore, as those criteria are closest to responsiveness, but we have found no large differences between the different models for this criterion. The summaries of the M_s model have the highest similarity to the original text and the M_f have the lowest similarity of 90%. The difference between the highest and lowest values are low.

For the Topical similarity, the results are mostly

Evaluation criterion	Genre	M_r	M_s	M_{rs}	M_f	M_{rsf}	original text
Distributional Semantics [%]	Rock / Pop	92	100	97	90	93	n/a
	Rap	94	100	99	86	92	
	$\sum$	92	100	98	90	93	
Topical [%]	Rock / Pop	44	100	76	41	64	n/a
	Rap	58	100	80	48	66	
	$\sum$	46	**100**	77	<u>42</u>	**64**	
Coherence [%]	Rock / Pop	110	95	99	99	100	100
	Rap	112	115	112	107	107	
	$\sum$	110	97	101	100	101	
Lyrics fitness [%]	Rock / Pop	71	53	63	201	183	100
	Rap	0	0	0	309	249	
	$\sum$	62	<u>47</u>	55	**214**	**191**	

Table 1: Automatic evaluation results for the 5 summarization models and 2 genre clusters. Distributional Semantics and Topical are relative to the best model (=100%), Coherence and Fitness to the original text (=100%).

in the same order as the Distributional Semantics ones, but with much larger differences. While the M_s model reaches the highest similarity, this is a self-fulfilling prophecy, as summaries of M_s were generated with the objective of maximizing topical similarity. The other two models that incorporate M_s (M_{rs} and M_{rsf}), show a much higher topical similarity to the original text than M_r and M_f.

Coherence is rated best in M_r with 110%. All other models show a coherence close to that of the original text - between 97% and 101%. We believe that the increased coherence of M_r is not linguistically founded, but merely algorithmic. M_r produces summaries of the most central sentences in a text. The centrality is using the concept of sentence similarity. Therefore, M_r implicitly optimizes for the automatic evaluation metric of coherence, based on similar consecutive sentences. Sentence similarity seems to be insufficient to predict human judgments of coherence in this case.

As might be expected, methods explicitly incorporating the Lyrics fitness produce summaries with a fitness much higher than the original text - 214% for the M_f and 191% for the M_{rsf} model. The methods not incorporating fitness produce summaries with much lower fitness than the original - M_r 62%, M_s 47%, and M_{rs} 55%. In the Rap genre this fitness is even zero, i.e. summaries (in median) contain no part of the chorus.

Overall, no single automatic evaluation criterion was able to explain the judgments of our human participants. However, considering Topical similarity and fitness together gives us a hint. The model M_f has high fitness (214%), but low Topical similarity (42%). The M_s model has the highest Topical similarity (100%), but low fitness (47%). M_{rsf} might be preferred by humans as it strikes a balance between Topical similarity (64%) and fitness (191%). Hence, M_{rsf} succeeds in capturing lines from the most relevant parts of the lyrics, such as the chorus, while jointly representing the important topics of the song text.

6 Conclusion

In this paper we have defined and addressed the task of lyrics summarization. We have applied both generic unsupervised text summarization methods (TextRank and a topic-based method we called TopSum), and a method inspired by audio thumbnailing on 50k lyrics from the WASABI corpus. We have carried out an automatic evaluation on the produced summaries computing standard metrics in text summarization, and a human evaluation with 26 participants, showing that using a fitness measure transferred from the musicology literature, we can amend generic text summarization algorithms and produce better summaries.

In future work, we will model the importance of a line given the segment to avoid cutting off important parts of the chorus, as we sometimes observed. Moreover, we plan to address the challenging task of abstractive summarization over song lyrics, with the goal of creating a summary of song texts in prose-style - more similar to what humans would do, using their own words.

Acknowledgement

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

References

Mehdi Allahyari, Seyed Amin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. 2017. Text summarization techniques: A brief survey. *CoRR*, abs/1707.02268.

Rachit Arora and Balaraman Ravindran. 2008. Latent dirichlet allocation based multi-document summarization. In *Proceedings of the second workshop on Analytics for noisy unstructured text data*, pages 91–97. ACM.

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.

Mark A. Bartsch and Gregory H. Wakefield. 2005. Audio thumbnailing of popular music using chroma-based representations. *Trans. Multi.*, 7(1):96–104.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 481–490, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Brackett. 1995. *Interpreting Popular Music*. Cambridge University Press.

Wei Chai and Barry Vercoe. 2003. Music thumbnailing via structural analysis. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 223–226.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494. Association for Computational Linguistics.

Jean Yves Delort, Bernadette Bouchon-Meunier, and Maria Rifqi. 2003. Enhanced web document summarization using hyperlinks. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, HYPERTEXT '03, pages 208–215, New York, NY, USA. ACM.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Mohamed Abdel Fattah. 2014. A hybrid machine learning model for multi-document summarization. *Applied intelligence*, 40(4):592–600.

Mohamed Abdel Fattah and Fuji Ren. 2009. Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. *Comput. Speech Lang.*, 23(1):126–144.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics.

Ruifang He and Xingyi Duan. 2018. Twitter summarization based on social network and sparse reconstruction. In *AAAI*.

Leonhard Hennig. 2009. Topic-based multi-document summarization with probabilistic latent semantic analysis. In *Proceedings of the International Conference RANLP-2009*, pages 144–149.

Meishan Hu, Aixin Sun, Ee-Peng Lim, and Ee-Peng Lim. 2007. Comments-oriented blog summarization by sentence extraction. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 901–904, New York, NY, USA. ACM.

Nanzhu Jiang and Meinard Müller. 2015. Estimating double thumbnails for music recordings. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 146–150.

Florian Kleedorfer, Peter Knees, and Tim Pohle. 2008. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Ismir*, pages 287–292.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 703–710. AAAI Press.

Mark Levy, Mark Sandler, and Michael Casey. 2006. Extraction of high-level musical structure from audio data and its application to thumbnail generation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V. IEEE.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2).

Stuart Mackie, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014. On choosing an effective automatic evaluation metric for microblog summarisation. In *Proceedings of the 5th Information Interaction in Context Symposium*, pages 115–124. ACM.

Qiaozhu Mei and ChengXiang Zhai. 2008. Generating impact-based summaries for scientific literature. In *ACL*.

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.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

Jahna Otterbacher, Güneş Erkan, and Dragomir R Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 915–922. Association for Computational Linguistics.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. 2015. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1954.

Daraksha Parveen and Michael Strube. 2015. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1298–1304. AAAI Press.

Samuel Pecar. 2018. Towards opinion summarization of customer reviews. In *Proceedings of ACL 2018, Student Research Workshop*, pages 1–8. Association for Computational Linguistics.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Mathieu Brucher, Mathieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Horacio Saggion and Thierry Poibeau. 2013. *Automatic Text Summarization: Past, Present and Future*, pages 3–21. Springer, Berlin, Heidelberg.

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. 2018. Summarization evaluation in the absence of human model summaries using the compositionality of word embeddings. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 905–914. Association for Computational Linguistics.

Cédric Villani. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2016. A sentence compression based framework to query-focused multi-document summarization. *arXiv preprint arXiv:1606.07548*.

Kento Watanabe, Yuichiroh Matsubayashi, Naho Orita, Naoaki Okazaki, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, Jordan Smith, and Masataka Goto. 2016. Modeling discourse segments in lyrics using repeated patterns. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1959–1969.