Efficient Lyrics Extraction from the Web

Gijs Geleijnse **Jan Korst**
Philips Research
Prof Holstlaan 4
5656 AA Eindhoven (the Netherlands)
{gijs.geleijnse,jan.korst}@philips.com

Abstract

We present a novel method to extract lyrics from the Web. The aim is to extract a set of multiple versions of the lyrics to a song. Lyrics can be identified within a text by a regular expression. We use a projection of a document to efficiently identify lyrics within the document by mapping it to a regular expression. We describe a method to cluster the multiple versions of the lyrics by filtering out erroneous texts such as lyrics to other songs. For reasons of efficiency, we do this by comparing fingerprints instead of the texts themselves.

Keywords: Lyrics, Web, Google, Regular Expressions.

1. Introduction

People are interested in the lyrics of the songs they listen to. Instead of searching for lyrics on the web ourselves, we are interested whether we can gather lyrics automatically. This can be an attractive feature within an audio device. If the extraction is done fast, the lyrics of a song can be displayed while playing.

EvilLyrics [1] automatically extracts lyrics from a number of large lyrics depot sites. Unlike our method, EvilLyrics makes use of the homogeneous structure of the pages within these websites [1].

In [3] a method is introduced to automatically extract lyrics from the web. This method is the first to extract lyrics from heterogeneous sources. Lyrics are identified within web pages using multiple string alignment. The authors showed that this is an effective method to acquire correct versions, however the method itself has a high time and space complexity. Another drawback is that the method assumes multiple distinct documents containing the lyrics.

Contrary to [3] we are not interested in identifying a *correct* version of the lyrics, but in a number of possible different versions of the lyrics to the song as occurring on the Web. Differences occur due typo's and misheard words, the omission of a repeating chorus, the insertion of line and fragment breaks, etc. In this paper we present an efficient method to extract a number of versions of the lyrics of a song from the Web. Given such a set of versions of the lyrics, a user can select (or construct) a satisfiable version.

2. Method

Our method consists of three phases. In the first phase, we gather documents (potentially containing lyrics) using Google. Then we identify the lyrics from these pages. When we have extracted a set of potential lyrics, in the last phase we filter out irrelevant elements from this set, e.g. lyrics to other songs.

2.1. Collecting documents

We use Google to retrieve a number of pages that are likely to contain the lyrics of the song of interest. To this end we query *allintitle: "song title", "artist name", lyrics.* We extract the URLs of the n best scoring documents, we used $n = 40$. Per web site, we only store the best ranked URL. Moreover, we do not store URLs of documents in other than hypertext format, e.g. with the extensions *.doc* or *.pdf.*

If this approach does not yield n URLs, we send broader queries to Google (*allintitle: "song title", "artist name"* and *"song title", "artist name"*) and extract the resulting URLs.

After gathering the URLs we collect the corresponding documents.

2.2. Extracting lyrics from the documents

After gathering the documents corresponding to the URLs identified, we extract the lyrics from these documents.

We make use of the structural representation of lyrics. Like poetry, lyrics consist of stanzas separated by blank lines. Each stanza consists of one or more lines, where a line is a sequence that ends with an end-of-line marker.

We use these characteristics to identify lyrics in a hypertext. An additional observation used is that the lyrics in general have a uniform lay-out. This implies that within lyrics no html-tags occur other than the end of line tags <br>.

Lyrics within a hypertext can thus be described by a regular expression. If we consider a hypertext document as a string, we can thus extract the first substring that is described by such a regular expression.

For reasons of efficiency however, we first map each substring in the hypertext document separated by an end-of-line

[1] http://www.evillabs.sk/evillyrics/

tag to one of the characters **l**, **b** and **n** according to the following rules.

- b if the substring is empty or the string only consists of white space characters (blank).
- l if the substring does not contain any html-tags and contains between 3 and 80 characters (lyrics line).
- n otherwise (non lyrics).

We now have a mapping of the document to a sequence S of the form $(\mathbf{l}|\mathbf{b}|\mathbf{n})^+$. We match this sequence to the regular expression describing lyrics given by

$$R = (\mathbf{l}^{1-20} \cdot \mathbf{b})^{1-12} \cdot \mathbf{l}^{1-20} \quad (1)$$

where $\mathbf{a}^{i-j}$ denotes "a sequence of **a**s of a length between i and j".

Now we match the sequence S to R and extract the first substring that matches. The small size of the sequence S assures an efficient mapping with a time complexity upper-bound of $O(|S|)$ [2].

We use the corresponding substring in the document to obtain the lyrics within it. This leads to a set L of extracted data with a maximal size of n.

2.3. Lyrics identification by clustering

We observed that we indeed obtain a number of versions of the lyrics of some song a. However, not all extracted texts are lyrics of a. The main problem is that we extract the wrong lyrics from a page containing multiple lyrics. Also lists such as the tracks of an album that match the regular expression are extracted.

We could use multiple string alignment on the strings in L to compute some 'correct' version of the lyrics of a. However we are interested in efficiently obtaining a set of versions of the lyrics as they occur on the web. We therefore use a fingerprinting technique to filter out erroneously extracted data.

Fingerprints of the lyrics of song a should on the one hand be discriminative from lyrics of other songs or other texts. On the other hand, we would like multiple versions of lyrics to have the same fingerprints.

We therefore choose to select the m longest words in the lyrics as the fingerprints. In general, the longest words are the most discriminative in a text. If the longest words in two texts have no overlap, we can assume that the texts handle a different topic.

Per string in L we identify the fingerprint, we used $m = 5$. We use the assumption that two strings are both versions of the same lyrics, if they share at least k fingerprints. In our experiments we used $k = 3$ as a threshold for two texts being both versions of the same lyrics.

Having a list L of potential lyrics each with m fingerprints, we sort this list based on the sum of length of the fingerprints.

The first elements in the L have the longest fingerprint words and thus we expect the complete versions of the lyrics is the first part of the list. The top element in the list is selected as the reference element. We compare each of the fingerprints in L to this reference element. If a text shares at least k fingerprints it is added to the cluster with the reference element. Else, a new set is created and this text is selected as reference element for this set.

3. Experiments

We tested our method on the set of song titles used in [3]. For 239 of the 258 songs, the selected cluster relates to the lyrics of the intended song (93%). For 13 songs no lyrics were found because the different versions that were found were too diverse. One of these 13 songs consisted out of one fragment with too many lines (*Absolute Beginner* by *Das Boot*). For the other 12 songs, 10 songs consist out of a single fragment. Since our regular expression only accepts lyrics with at least two fragments, these 10 lyrics were not recognized.

In addition, for 6 songs a final clustering was found that turns out not to contain the lyrics of the song. For 4 of these 6 songs, the song title of the intended song appears in the lyrics of another song by the same artist. The intended song consists of one fragment while the found song consists of multiple fragments.

4. Conclusions and future work

We have developed an method to extract lyrics from heterogeneous sources from the Web using regular expressions. After having extracted a set of texts, we generate a cluster of (possibly multiple) versions of the lyrics we are interested in. The methods presented here are efficient, considering the possibility to use them in real-time applications such as MP3-players.

In future work, we can improve our results by filtering out annotations from extracted lyrics. If names of artists, composers and websites are removed, the fingerprinting phase can become more effective. We can construct or search for a 'correct' version of the lyrics among the set of versions using multiple string alignment methods.

References

[1] V. Crescenzi and G. Mecca. Automatic information extraction from large websites. *Journal of the ACM*, 51(5):731–779, 2004.

[2] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, Cambridge, UK, 1997.

[3] P. Knees, M. Schedl, and G. Widmer. Multiple Lyrics Alignment: Automatic Retrieval of Song Lyrics. In *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 564–569, London, UK, September 2005.