FestivalFinder

Extracting the Lineup from Music-Festival Websites

Lukas Plechinger
Group 8

The Idea

Finding a music festival which fits someone's music taste is hard. Either one scrolls through dozens of lineups on different festival websites or the ”perfect” festival has been found by accident or by getting tailored ads.

However, sometimes there is no awareness of the festival at all, because of poor marketing, wrong target groups in personalized ads etc.

The idea for the project was to write a software which can find the perfect matching festival for a person, based on the musical taste.

The application eventually consists of two components:

- **Collector**
  Automatically extracts the lineup from the Website of the festival.
- **Recommender**
  Gets musical taste by connecting to Spotify, YouTube, etc. and then recommends festivals based on artists a user likes.
  Because of time constraints, just the Collector has been implemented yet.

Extracting Artists

To get a big dataset of festivals and the corresponding lineup, the process of extracting the lineup should work as autonomous as possible. The URL of the webpage where the lineup is listed should be provided and the output should be a set of artists.

The first approach was to experiment with NLP tools (Apache OpenNLP) but the text format (Listing 1) was not ideal for things like entity extraction.

```
Tickets Line Up Lottery News Venue Geländeplan Caravan Camping Komfort Glamping Anreise
FAQ freQteam Green Green Team Green Camping powered by EVN My Account Line up All Days
Thu, 15. Fri, 16. Sat, 17. Daypark Nightpark Swedish House Mafia Friday Macklemore
Saturday Twenty One Pilots Thursday Sunrise Avenue Thursday Dimitri Vegas Like Mike
Saturday Alligatoah Thursday Capital Bra Saturday Prophets Of Rage Friday The Offspring
[…]
```

Listing 1: Raw text (excerpt, stripped HTML tags)[1]

After realizing that using NLP is probably not the right way to extract the artists, a new approach had been found.

A similar algorithm had been used previously to extract tabular data out of HTML which does not contain tables. It uses the following steps:

1. Iterate through the DOM tree. For every tag containing text, search in a database containing artist names for artists with the same name as the text value. Ignore common terms such as weekdays (Monday,..) and country names.
2. Calculate a score for every level in every subtree (of the root), the score is higher, if the number of artists found on the level is higher.
3. Weigh the score: If the text is in a more important tag such as `<h1>`, `<strong>`,…, the score is higher.
4. The subtree and level within the subtree which has the highest score contains the most artists and is probably the one containing all artists.

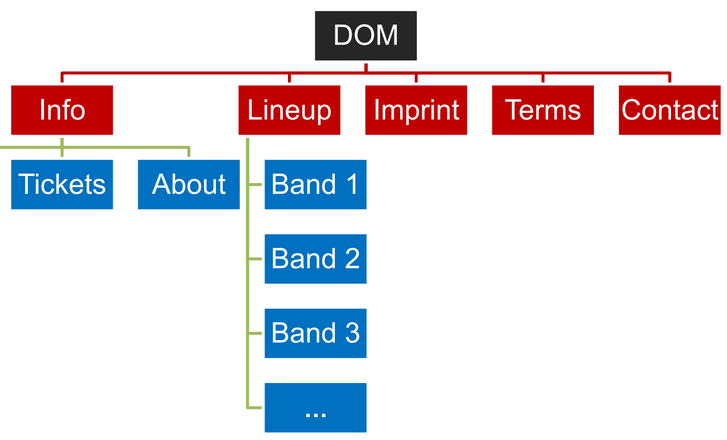

Figure 1: Example of a typical DOM-Tree from a festival website. In this example, the „Lineup“ tree with level 2 is the one with the highest score.

Dataset

As dataset, the open Discogs[2] dataset has been used. It is an open (CC0) and community driven database about music artists, songs, releases, etc.

The data is filled into a Postgres database for persistence and into a Lucene search index for fast searching. An overview of the dataset can be found in table (1).

Dataset outline

Entity	# of entries
Artists	6.291.846
Artist names (name, aliases and „real name“)	15.475.402
Releases (Singles, EPs, Albums,…)	11.231.882
Artists with acc. „Billboard Top 100“ chart position	3.628
Styles (“Dark Metal”, ”Free Jazz”,…)	540
Genres (“Pop”, “Rock”, “Classical”,…)	15

Table 1: Outline of the Discogs dataset[2]

Searching & Evaluation

Because of the nature of the problem, the only reasonable way to verify the data is by manually creating a “gold standard data”. It is created by manually looking at the festival website, searching for the artists on the Discogs website and looking for the correct artist.

By creating this gold standard data, the following problems emerged:

- There are many artists with the same name, e.g. Anna more than 120.
- Sometimes, further googling is necessary to determine the correct artist (country, discography, last album)
- Because Discogs is user-curated, the data quality varies and there are also a lot of duplicates (dump is released monthly and merges records)

As it is extremely common to encounter bands with the same name, the following rules to rank the search results have been made:

- First, search for an artist by name and alias. Matches by the name are weighted higher than matches by the alias (See Table 2 for comparison).
- Festivals tend to invite artists, which released a song/album recently. So artists with newer releases are ranked higher.
- On the other hand, festivals invite popular artists which had a high chart ranking in the past. The chart ranking is a the average ranking of an artist and a number between 0 and 1, 1 means that the artist has been ranked number 1 for all chart songs, 0 that the artist had never been in the charts.

To evaluate the system, a lot some combinations of different search and ranking strategies had been performed and have been compared with the gold standard.

	All		Name Only		Alias-Low		Alias-High		Alias-Middle		Popularity	
Festival	**Full**	**Poss**	**Full**	**Poss**	**Full**	**Poss**	**Full**	**Poss**	**Full**	**Poss**	**Full**	**Poss**
Frequency	0,87	0,94	0,70	0,75	0,73	0,78	0,47	0,51	0,69	0,74	0,70	0,75
Melt	0,82	0,82	0,83	0,83	0,86	0,86	0,62	0,62	0,84	0,84	0,83	0,83
Southside	0,86	0,86	0,78	0,78	0,81	0,81	0,50	0,50	0,79	0,79	0,78	0,78
Sziget	0,77	0,82	0,70	0,74	0,68	0,72	0,44	0,47	0,68	0,72	0,70	0,74

Table 2: Results of the evaluation

The Results in Table 2 show, that the algorithm seems to work very well with success rates >80% (of possible success).

Possible success of 1 would be, that all acts which are in the database were found correctly. The full success could only be 1 if all artists of the festival are also in the Discogs dataset.

- **All**: (Name, Alias, last release year, chart popularity)
- **Alias (low, high, middle):** the weight of the alias in contrast to the name

Future work & Optimization

There are tons of opportunities to extend and optimize the approach.

- Taking genre and styles into account: Festivals usually present artists where those are similar.
- More metrics for popularity: YouTube click count, sold/streamed tracks, use more charts as input, etc. The dataset already contain some helpful info, for example links to YouTube videos
- Allow user input to flag wrong artists
- After the recommender part is implemented, the taste of the users will improve the popularity index

References

1 Frequency Lineup from: www.frequency.at/lineup
2 Discogs dataset: data.discogs.com