Emotion Patterns in Music Playlists

Sara Giammusso[1][2] Mario Guerriero [1][2]

[1]MSc student in Data Science Department, EURECOM, Télécom ParisTech, France
[2]MSc student in Department of Control and Computer Engineering, Politecnico di Torino, Italy

Third Project meeting

Table of Contents

1. Introduction
2. LyricsAverageNN: Model Description
3. LyricsAverageNN: Model Evaluation
4. LyricsNN: Model Description
5. LyricsNN: Model Evaluation
6. Conclusion
7. References

Table of Contents

1. Introduction
2. LyricsAverageNN: Model Description
3. LyricsAverageNN: Model Evaluation
4. LyricsNN: Model Description
5. LyricsNN: Model Evaluation
6. Conclusion
7. References

Introduction

Previously On Sara&Mario Project...

We analyzed already existent systems and we came up with some decisions about what to do.

Next steps:

- Start building some simple classifiers
- Evaluate those simple classifiers
- See how we could improve it

Table of Contents

1 Introduction

2 LyricsAverageNN: Model Description

3 LyricsAverageNN: Model Evaluation

4 LyricsNN: Model Description

5 LyricsNN: Model Evaluation

6 Conclusion

7 References

The Idea

- Based on the same idea of the Nearest Neighbour
- Find the "closest" song and get its label
- We are not actually comparing lyrics to each other
- We compare lyrics we want to classify to 4 points
    - Each point is supposed to be representative of an emotion

Why 4 points?

- Simplicity
- Speed
- We just wanted to get some quick insights

How it works

- Evaluate the word vector norm for each lyrics
- Group (compute average) lyrics by emotion
- Given a lyric find the closest representative point

Table of Contents

1. Introduction
2. LyricsAverageNN: Model Description
3. LyricsAverageNN: Model Evaluation
4. LyricsNN: Model Description
5. LyricsNN: Model Evaluation
6. Conclusion
7. References

1-fold validation

We used the simplest validation technique: 1-fold validation

- 90% for training
- 10% for validation

Accuracy

- Not very impressive :-(
- Around 28% accuracy

Table of Contents

1 Introduction

2 LyricsAverageNN: Model Description

3 LyricsAverageNN: Model Evaluation

4 LyricsNN: Model Description

5 LyricsNN: Model Evaluation

6 Conclusion

7 References

The Idea

DISCLAIMER

This model is slower then the previous one

- A more general version of the LyricsAverageNN classifier
- Based on the same idea of the k-Nearest Neighbour (again)
- Find the "closest" song and get its label
- Now we are comparing lyrics to each other

How it works

- Evaluate the word vector norm for each lyrics
- Do NOT group anything
- Given a lyric find the closest lyrics and get their majority label

Table of Contents

1 Introduction

2 LyricsAverageNN: Model Description

3 LyricsAverageNN: Model Evaluation

4 LyricsNN: Model Description

5 LyricsNN: Model Evaluation

6 Conclusion

7 References

1-fold validation

We used the same validation technique we used for the previous model

- 90% for training
- 10% for validation

Accuracy

- Not very impressive :-(
- 33% accuracy (k=1)
- 30% accuracy (k=3)
- 29% accuracy (k=5)

Table of Contents

1. Introduction
2. LyricsAverageNN: Model Description
3. LyricsAverageNN: Model Evaluation
4. LyricsNN: Model Description
5. LyricsNN: Model Evaluation
6. Conclusion
7. References

What's next?

- We didn't use Fastext word vectors for the sake of time...
- We need better feature engineering
- Move to better classifiers (it didn't make sense to use them for now)
- Find out some better pre-processing techniques
- Implement better validation technique (cross-validation!!!)

Table of Contents

1 Introduction

2 LyricsAverageNN: Model Description

3 LyricsAverageNN: Model Evaluation

4 LyricsNN: Model Description

5 LyricsNN: Model Evaluation

6 Conclusion

7 References

References I

[1] spaCy: Industrial-Strength Natural Language Processing
`https://spacy.io/`

[2] Intro to NLP with SpaCy
`https://nicschrading.com/project/Intro-to-NLP-with-spaCy/`