Melody generation based on emotions

Joana Simões
University of Coimbra, Department of Informatics Engineering
joanasimoes@student.dei.uc.pt

Abstract

This project explores the creativity of artificial intelligent systems, aiming to generate melodies based on a desired emotion between joy, sadness, relax and distress. In order to do it, it was used an LSTM because of its prediction characteristics and power of analysing larger patterns. To control the generative model in a way that induces emotions, it was used in a logistic classifier.

The results show that the system has some difficulty in generating songs in certain of the intended emotions, and this fact was confirmed by people who evaluated the system.

1 Introduction

In this work, we propose to find solutions to an AI-aided music generation system based on one's feelings at a specific moment. What fuels us to dive into this project is our desire to discover at which point an AI agent can be creative within the human pattern of what sounds good.

The field of computational creativity has seen significant progress in the last few years. Yet, music generation remains a great challenge when compared with other areas. [7]. Unlike, for example, images, the music changes over time, and the emotions transmitted by the music depend on a lot of characteristics, such as tempo, and it varies from person to person.

In recent years, the use of deep learning algorithms has improved creativity in music. These new techniques used have given people with little music knowledge, the ability to create their own music with the help of AI systems. However, these AI systems still have certain flaws. It's not uncommon for them to fail to create emotional and expressive music, two characteristics that most interfere with the quality of music.

The main goal of the project is to involve people's emotions in this creative process. To that aim, we will explore the influence of emotions on music generation. Besides this, we want to see if all emotions are created equal, or if certain emotions are harder to induce because of their complexity.

To accomplish this, we want to develop an AI system that creates melodies based on a specific emotion. A user enters an emotion and the system creates a unique melody based on that feeling. Finally, the user uses a Likert scale to evaluate the result. The model's accuracy will then be calculated using the evaluations from different user experiments.

This project follows the work of Ferreira et al. in generating music with sentiment [3]. In our project, we intend to build on the work already done but extend the scope to other sentiments, beyond the positive and negative emotions used by Ferreira. To define emotions, this project relies on Russel's work [6], in which he states that emotions can be characterized by their valence and arousal values. The valence is related to the level of pleasantness that, in this case, the music transmits. The arousal is associated with the energy and intensity that the music gives, going from calm to excited music.

The focus of this project are the most basic sentiments, according to Russell's graph, where we have one sentiment per quadrant (see Image 1). Thus, the proposed model extends the induced sentiments to joy, distress, relax, and sad. To create them, an LSTM will be used to generate music and a logistic regression model to control it.

The source code of the developed system can be accessed here.

2 Related Work

Music creation requires aspects both technical and non-technical to produce satisfactory results. The technical aspects, such as chords and arpeggios, are needed to understand how to encode the melodies that will be used to train models. Other non-technical aspects, like expressiveness, are important for evaluating their quality. To add people's emotions to the equation, it's necessary to extract some characteristics of the music and map them into emotions. Russell proposed a model of emotion for this mapping that uses the valence and arousal of the music [6].

Several works have already been made for automatic generating music. There are two groups in this field: symbolic domain generation and audio domain generation. But since this project will be using symbolic domain (through MIDI files), we will only discuss approaches that use it.

Deep learning models have been shown to achieve high-quality scores doing it. Nowadays, most models used to make music are Recurrent Neural Networks (RNNs) or similar models. RNNs are a type of neural network that use sequential information, i.e. previous values of the sequence, to predict future values, and therefore the values depend on each

other. Lately, they have been widely used to predict words, and in text generation. In the ambit of this project, it will be used a variation of RNN, the Long Short-Transform Memory(LSTM) that learns through gradient-descent. They are widely used in problems where it is necessary to remember patterns for a long time, as is the case of text generation and music generation.

In 2016, the magenta team used an RNN to surpass the difficulty in creating long-term structures when generating melodies [8]. A year later, Radford et al. [5] had discovered that certain units of the model being related to positive and negative feelings. Ferreira's model used this finding to train their model to generate positive and negative musics in 2019 [3]. In short, over the years there have been several improvements of the models that are used to produce quality music.

3 Resources

3.1 Dataset

The model will be trained with VGMIDI Dataset [2] [3] of labelled and unlabelled musics in MIDI format.

To train the LSTM to generate music, we will use 869 midi files, 757 files to train and 112 to test the model. Before using this dataset dimension, the model was trained with a smaller one, but the results remained below expectations. Although the dataset has increased, we still know it has a small size for the type of problem it is, however for the computational capacity available, it was not possible to extend the dataset further.

For the classifier, it was used 200 midi files, annotated by 30 people according to Russell's valence-arousal model.

Although the dataset doesn't have the same number of female and male annotators and the same number of persons with the same culture (they are from different countries), meaning that the information may be biased by the people who annotated it, this information will be ignored since it was considered to be out of the scope of the project.

3.2 Frameworks

The project will be implemented in Python with the help of other libraries. To handle MIDI files, we will use the library Music21, which allows converting MIDI files to encoded notation, and vise-versa. The LSTM and the classifier were created, respectively, with TensorFlow and Scikit-learn.

4 Approach

4.1 Process

The first step was mapping basic emotions (joy, sadness, distress and relax) into their numerical values of valence and arousal, based on Russell's' circumplex model [6], see Table 1. Then, the initial dataset was modified to include one of these categories for each music, based on their values of valence and arousal.

The project then followed a similar approach to the one used by Ferreira et al. [3]. First, it was created a LSTM that generates melodies based on a previous set of melodies given to it. Next, the classifier was created so that it identifies the transmitted emotion of a certain melody using the labelled dataset. Then, with the two models trained, they were modified, so that the system can generate new melodies based on a given emotion. Finally, the generated music was evaluated with the classifier model previously created.

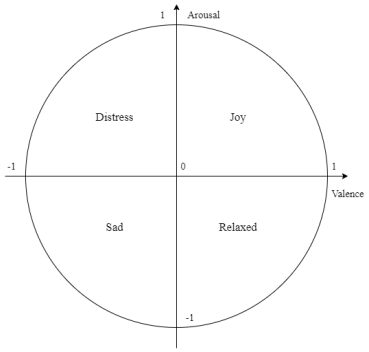

Figure 1: Circumplex model of Russel

The final results were evaluated by the users as described in the evaluation section. This human evaluation was taken into account to measure the accuracy of our model.

Data annotation

The VGMIDI labelled dataset [3] contains musics and a JSON file with the pieces (musics) annotated. This last file contains, among other parameters not used in the scope of this project, the valence and arousal that each of the annotators considered for each interval in the music.

To extract the content of the file, the original code [3] was modified so that it returns a JSON file where each item contains all the information of each music. After that, the resulted data in a JSON file was converted to a CSV format in order to be easier to handle and analyse. The arousal and valence parameters, previously a list of lists with the values of each user, were reduced to just the average of all the values. A new column has also been added to the CSV dataset, this column includes the mapping of the valence and arousal values (in a range of -1 to 1) into the emotion according to the Russel's circumplex [6] and illustrated in figure 1.

After preprocessing the dataset to include the emotions, it was noticed that the dataset was imbalanced (see graph 2). The negative classes, in terms of valence, had fewer samples than the positive ones. This imbalance could explain the results achieved by Ferreira et al. [3] where the negative musics had worst results than the ones achieved for the positive melodies generated.

The results in graphic 2 show the imbalance present in the dataset. As shown in the table, the dataset has a small percentage of samples of the classes "Sad" and "Distress" when compared to the other classes. This imbalance will have impact on the classifier results, since it will have difficulty to classify melodies that induce negative emotions: Distress and Sadness.

To understand the imbalance within the dataset, a distribution analysis was carried out according to their valence and energy values of the songs in the dataset.

The analysis showed (see graph 3) that, despite being in

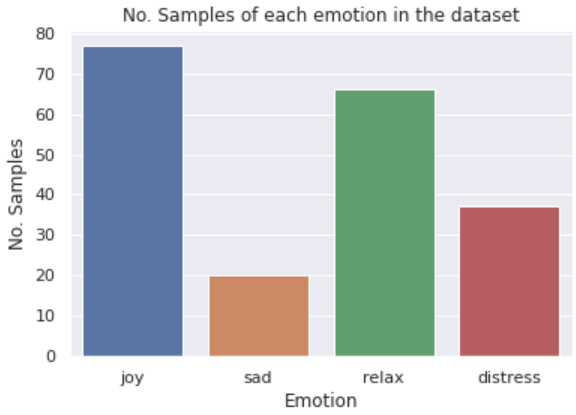

Figure 2: Samples of each class of emotions in the dataset

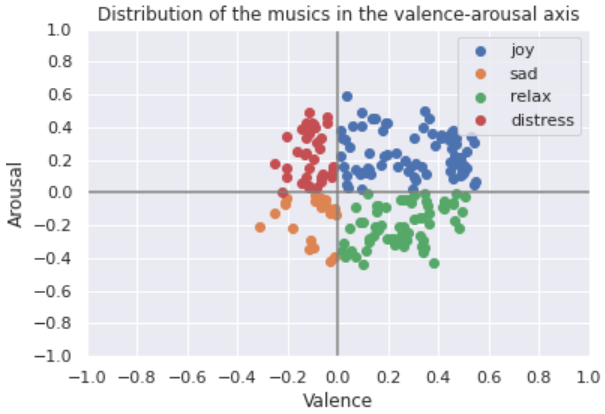

Figure 3: Distribution of songs according to valence and arousal, proposed by Russel

different quadrants, the songs have values in a range below the interval [-1, 1], in many cases very close to the border axes between quadrants. It is also possible to note that the songs with negative valence (distress and sad) present a range even lower on the valence axis, being distributed very close to the valence = 0 axis. These facts can already predict that certain songs will be confused by the classifier.

One solution to try to differentiate the classes further would be to remove the songs at the boundary between quadrants. However, given the small size of the dataset, this possibility was discarded.

4.2 Music encoding

In order to process the musics in the LSTM and in the classifier, it is necessary to convert the MIDI files to numeric values. To do this, first we create a mapping of each note in the different musics in the unlabelled dataset into a word [3].

- **n_[pitch]:** *play a note with a given pitch number: any integer from 0 to 127;*
- **d_[duration]_[dots]:** *change the duration of the following note to a given duration type with a given amount of dots. Types are breve, whole, half, quarter, eighth, 16th and 32nd. Dots range from 0 to 3;*
- **v_[velocity]:** *change the velocity of the following notes to a given loudness number. It ranges from 4 to 128 with intervals of 4;*
- **t_[tempo]:** *change the tempo of the piece. Tempo is also discretized in bins of size 4, so it can be any integer in the set T from 24 to 160;*
- **. :** *end of the time step. Each step is one sixteenth note long.*
- **/n:** *end of a piece.*

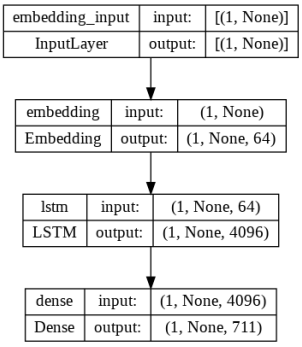

Figure 4: LSTM model

At the same time, it is attributed to each encoded string, a numeric value, representing the order where the encoded string appears.

Since the LSTM, only processes numeric values, the vocabulary dictionary was inverted so that the key becomes the index. To generate the musics, we do the reverse process, first we map the predicted integer note into the encoded string, and to write the new midi file, we convert all the encoded strings into their real value.

4.3 Models

LSTM

In a first attempt, the LSTM was created with 4 hidden layers, where each layer has 512 neurons. The model also contains an embedding layer initially to handle the input. To prevent overfitting, a dropout of 0.05 was added to each hidden layer.

Lastly, the model was trained for 3 epochs, again due to computational capacity, and with a learning speed of 0.001. Additionally, the model was improved with an Adam optimization.

After observing that the results were poor (see Table 1), and following Ferreira [3]and Radford's [5] models, the model was retrained. In this new attempt, only one hidden layer with 4096 neurons was used, while preserving the other configurations, as shown in image 4.

Classifier

To classify the emotions induced by the songs, a logistic regression model was trained. Several tests were elaborated to reach the best possible model configuration. In the classifier part is when there are more discrepancies from Ferreira's previous work. His work focused on a binary problem, between

positive and negative songs. However, in this work we have one more dimension, turning the problem into a multi-class problem. Since it is a multi-class problem, and by the documentation [1], it was decided to use a multinomial approach instead of One-versus-Rest because the latter would confound some songs that are on the borderline between quadrants. To penalize the inaccuracies, an L1 penalty was used, since we have independent classes. In addition, it was used a SAGA solver once it works well with multi-class problems.

The results with the different tests are shown in table 3.

4.4 Melody generation

After training both models, LSTM and classifier, the activated neurons of the LSTM hidden layer are calculated. These neurons are the ones that, according to the classifier, had more influence in the decisions.These neuron weights will be used to get the LSTM to generate songs according to the desired emotion.

To optimize the weights of the neurons for each emotion, it was used a genetic algorithm (GA), being necessary to perform it for each type of emotion. The GA starts with an initial population of 20 individuals, where each one is constituted by X genes, being X the number of neurons activated by the classifier.

The fitness function of each individual is calculated by how well it can "fool" the classifier. First, the gene values are overwritten in the original LSTM, and this new LSTM is used to generate melodies based on the desired emotion. Once generated, the melodies are evaluated with the previously trained classifier and the error of the predictions is calculated.

For each emotion, 5 epochs were used to optimize the neurons. At each iteration, the best 4 individuals were selected through elitism to be the parents. Then, these were recombined with crossover. In addition, a mutation rate of 20% was performed in order to introduce diversity in the population. In the end, the genes of the best individuals for each emotion are saved.

To generate new melodies, the neurons optimized for the desired emotion are used to replace the original LSTM neurons. With the resulting LSTM, new melodies are generated with 256 words each. To start the generation, an initial symbol ".", which represents the end of a time, was given to the model.

4.5 Evaluation

The second evaluation will be done by people through a survey. First, some data will be collected from the participants to characterize them, such as age, gender and musical knowledge. Then, each participant was asked to identify, for each song presented, the best emotion that the melody transmits to him/her. Finally, the respondents were also presented with some questions related to the creativity of the AI system, based on the success criteria used by Liapis Et al. [4]. This criterion involves novelty, quality and typicality, however, they have been adapted for this context resulting in:

- **Novelty**: How does the resulting melody differ from what the user is accustomed to?
- **Quality:** To what extent the user would listen to the produced melody?

Epoch	Cross entropy loss
1	4.4455
2	4.4379
3	4.4411
Average	4.4198

Table 1: Table with the results of training a LSMT with 4 hidden layers and 521 neurons in it

Epoch	Cross entropy loss
1	3.0546
2	2.8492
3	2.7404
Average	2.8814

Table 2: Table with the results of training a LSMT with 1 hidden layer and 4096 neurons in it

These categories were measured using a Likert scale to estimate the model's accuracy using a survey. The survey created can be accessed here.

5 Experimentation

5.1 Train the LSTM

As it can be observed by the results of table 1 and 2 of training the generative LSTM, using only one hidden layer with more neurons achieves better results than using multiple hidden layers with fewer neurons, despite the final number of neurons is equal in both. Therefore, in the following tests, only the second model was used, i.e. an LSTM with only one hidden layer with 4096 neurons.

5.2 Train the classifier

As expected, the results obtained (Table 3) were not good. The problem was mainly due to the small size of the dataset. The dataset only contains 200 songs, so, as there are four classes, the model does not have enough examples of each class to perform a good identification. Another important problem, as described before, was the imbalance present in the dataset. These two problems combined leaded to the low accuracy of the classifier.

To improve the results, one solution would be to generate synthetic data in order to reduce the imbalance of the dataset or use a larger and balanced dataset

Solver	Approach	Penalty	Neurons	Accuracy
Saga	M	L2	4096	47.5
Sag	M	L2	4096	47.5
Newton-cg	M	L2	4096	47.5
Lbfgs	M	L2	4096	47.5
Saga	M	L1	38	47.5
Liblinear	OvR	L2	4096	47.5
Liblinear	OvR	L2	34	45.0
Saga	OvR	L1	455	47.5

Table 3: Results of training the classifier with different parameters. M: multinomial; OvR: One-versus-Rest

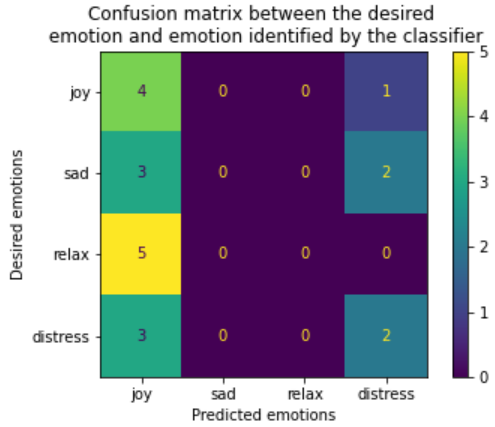

Figure 5: Confusion matrix between the emotion given to the system and the emotion predicted by the classifier on the generated songs

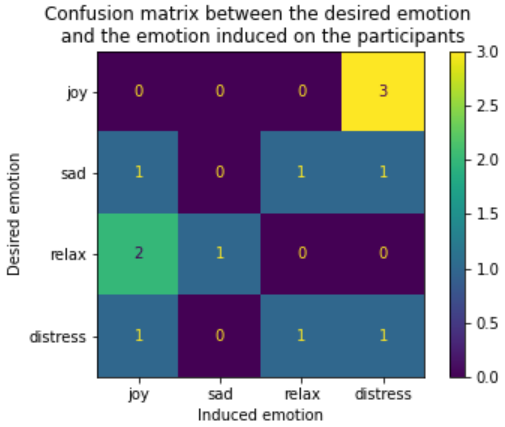

Figure 6: Confusion matrix between the emotion given to the generated songs and the emotion predicted by the participants

However, the results weren't great, the process will continue, using the configurations:

- Solver: Saga
- Approach: multinomial
- Penalty: L1

This configuration was chosen because of all the possible ones, it was the one that used a smaller number of neurons (that will be used as weights for the LSTM, and we want to change the smallest number of neurons possible to not alter the training) and that has a higher accuracy.

5.3 Generated melodies

Five melodies were generated for each type of emotion. The generated musics were given to the previously trained classifier in order to confirm the emotion of the generated songs.

The image 5 shows the confusion matrix between the emotion supposedly created by LSTM and the emotion identified by the classifier.

By the results obtained, it becomes clear that the classifier has some difficulty in classifying the correct emotions of the melodies, achieving an overall accuracy of 30%. After an analysis of the generated melodies, it can be inferred that this is caused by two main factors: (i) the melodies generated by the LSTM are indeed confusing, many of them being similar, although they should belong to different classes; (ii) the classifier has a low accuracy, i.e., both when changing the values of the LSTM neurons and when classifying the generated melodies, the error was high.

Comparing the results obtained by the model, and analysing the distribution of the dataset used for training, we can see that the model has some difficulty in distinguishing melodies with positive arousal value, i.e., more energetic songs like joy and distress. At the same time, it can also be seen that the model tends to generate musics with positive arousal, as can be seen by the fact that all the *relax* songs are all classified as *joy*, and the *sad* ones are classified as *joy* or *distress*. This may be due to the reasons mentioned above, or to the fact that the dataset used to train the LSTM (with

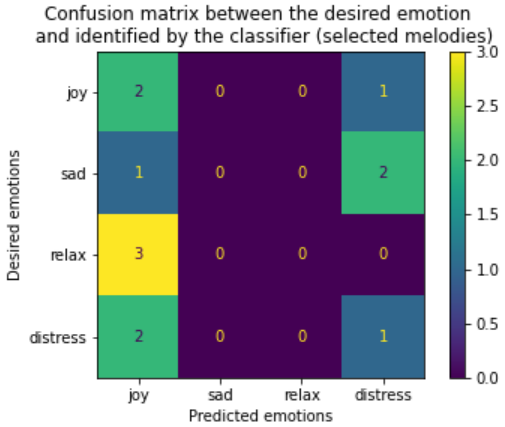

Figure 7: Confusion matrix between the emotion given to the system and the emotion predicted by the classifier on the generated songs (songs selected to present to the users)

unannotated songs) may also be biased, and contain essentially energetic songs. This may be true since the dataset is of music used in video games, which tend to be peppy songs.

5.4 Annotators evaluation

To confirm the results obtained, three individuals evaluated the generated melodies. In order not to bore the participants, only three melodies were chosen at random for each emotion, from all those generated, to be evaluated. The survey used can be accessed here.

As can be seen, when presenting the generated melodies to people, the system evaluation worsens, having an accuracy of only 8.33%, since the evaluators only agreed on one song in how that the emotion conveyed was equal to the emotion given to generate the song. However, it should also be noted that it was only tested with 3 people, being a small sample of people to draw conclusions. Nevertheless, the survey results are considered more accurate than the classifier evaluation, due to the problems previously described with the classifier and to the fact that the final goal of the project is to create a

system for people.

When listening to the songs, it is possible to notice that the sound generated is very strident, and the melodies are very energetic, two characteristics associated with tense / stressful songs. This fact may have led people to identify the emotion induced by most of the songs as *distress*, since most people are sensible to this kind of sound.

Also, the resulting melodies have a very electro beat, and the respondents are not used to this kind of music and have low musical knowledge, as the average musical knowledge was 2.33 on a scale of 1 to 5. These facts also contributed to the fact that most of the songs were considered as distress, when previously they had been considered as joy by the classifier.

This contrast between songs previously considered as joy, being now considered as distress by the participants, highlights the error that the classifier has, and that influences both the creation of the songs and the identification of emotions.

In some songs, there was a divergence between the participants, showing that the number of people is low considering the number of songs presented, that is, in certain songs each participant voted for different songs. In the situations where this happened, it was used a fourth person to hear the songs and unbalance the results.

In terms of the creativity of the AI system, the participants say they are surprised by the songs generated by the system, perhaps because of their low knowledge of the field of artificial intelligence and the advances it has made. On a scale of 1 to 5 to evaluate the creativity of the system, respondents gave the system a score of 3.67. When asked about how different the songs were from what they usually listen to, they rated the system with 4.67. The interviewees also answered that "Maybe" they would listen to the tunes again and "Yes" they would, when asked about their willingness to listen to more songs generated by the system.

6 Conclusions

The present work demonstrated how a modified LSTM can generate melodies that induce certain feelings in humans. The project focused on the most basic feelings according to Russell, having one feeling per quadrant. The LSTM was controlled by a logistic regression classifier trained to identify music's induced emotions.

One of the biggest limitations of this work was the computational capacity, the results could be better if the dataset to train the LSTM was larger, but as the computational power was limited, it was necessary to reduce the size of the datasets and the number o epochs used to train the models.

Although the results were not the best, we can draw some conclusions from this work. The most important is related to the previous point, in which the size of the datasets are crucial to obtain good results. From the previous facts it can also be concluded that not having a balanced dataset influenced the classifier, which in turn led to the fact that by altering the LSTM neurons to generate songs with certain sentiments, the resulting values did not generate the desired emotions.

Despite all the technical conclusions, regarding the size of the datasets and the epochs used to train the models, we can conclude that inducing emotions through the creation of melodies is not as straightforward as initially thought when we decided to go ahead with this project. Generating melodies by itself is already a complicated task, and adding emotions makes the task even more difficult.

The results obtained by the system were poor, due to all the reasons already mentioned and also due to the fact that the emotions that the songs induce in people differ greatly from person to person, and the number of people used to evaluate the model is too small to take conclusions.

For future work, it will be to use more powerful machines with the ability to train, and thus increase the size of the datasets and the epochs to train the models. Another solution would also be to try to balance the datasets. Before that, perhaps test the generated songs with a larger number of people, in order to measure more accurately the actual precision of the model. It would also be interesting to investigate how more complex emotions, inserted in specific positions of Russell's model, could be introduced in LSTM to generate melodies based on them.

References

[1] Linear Models - Logistic regression.

[2] Lucas N. Ferreira. lucasnfe/vgmidi: Dataset of piano arrangements of video game soundtracks labelled according to sentiment.

[3] Lucas N. Ferreira and Jim Whitehead. Learning to generate music with sentiment. 2019.

[4] Antonios Liapis and Georgios N. Yannakakis. Boosting computational creativity with human interaction in mixed-initiative co-creation tasks. 2016. Accepted: 2019-10-18T09:23:08Z Publisher: Digital Games Research Association (DiGRA) & Foundation of Digital Games (FDG).

[5] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to Generate Reviews and Discovering Sentiment, April 2017. arXiv:1704.01444 [cs].

[6] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, December 1980.

[7] Isaac Tham. Generating music using deep learning, August 2021.

[8] Elliot Waite et al. Generating long-term structure in songs and stories. *Web blog post. Magenta*, 15(4), 2016.