RAFAEL RODRIGUES PADOVANI

AUTOMATIC BACKGROUND MUSIC SELECTION FOR TABLETOP GAMES

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS - BRASIL
2018

Ficha catalográfica preparada pela Biblioteca Central da Universidade Federal de Viçosa - Câmpus Viçosa

T
P189a
2018

Padovani, Rafael Rodrigues, 1991-
Automatic background music selection for tabletop Games / Rafael Rodrigues Padovani. – Viçosa, MG, 2018.
ix, 52 f. : il. (algumas color.) ; 29 cm.

Texto em inglês.
Inclui apêndice.
Orientador: Levi Henrique Santana de Lélis.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f. 45-49.

1. Processamento de linguagem natural (Computação).
2. Aprendizado supervisionado (Aprendizado de máquina).
3. Jogos para computador - Canções e musica. 4. Música.
I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação.
II. Título.

CDD 22. ed. 006.35

RAFAEL RODRIGUES PADOVANI

AUTOMATIC BACKGROUND MUSIC SELECTION FOR TABLETOP GAMES

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae.*

APROVADA: 09 de abril de 2018.

Cláudio Fabiano Motta Toledo

Luiz Chaimowicz

Levi Henrique Santana de Lelis
(Orientador)

I dedicate this work to my beloved parents Gilberto and Vânia.

Acknowledgments

First of all, I thank to God because he is the light, strength, fortress and wisdom that gives sense to my life. For all that He has accomplished for me.

To my adviser, Prof. Levi, for his dedication, patience and teachings. For contributing to my professional growth and for being also an example to be followed.

To CAPES for funding my studies. To all the professors and employees of the Department of Informatics of UFV, who somehow contributed to my professional growth

To my parents, Gilberto and Vânia, my infinite thank for the education and for all the support and encouragement in my studies. You are my examples of life, generosity, hospitality, honesty, responsibility and determination.

To my family for all the support.

To Luiza, for all love, companionship, understanding, encouragement and especially for always believing in me.

To my brother, Thiago, for always being ready to help me.

Finally, I thank all the friends of the master's degree by the companionship and support.

Contents

List of Figures

List of Tables

Abstract

PADOVANI, Rafael Rodrigues, M.Sc., Universidade Federal de Viçosa, April, 2018. **Automatic Background Music Selection for Tabletop Games.** Adviser: Levi Henrique Santana de Lelis.

System accuracy is a crucial factor influencing user experience in intelligent interactive systems. Although accuracy is known to be important, little is known about the role of the system's error distribution in user experience. In this dissertation we show, in the context of background music selection for tabletop games, that supervised learning algorithms can make the system "indecisive" by performing errors that are sparsely distributed in a game session. We then introduce Bardo, a real-time intelligent system to automatically select the background music for tabletop role-playing games. Bardo selects and plays as background music a song representing the classified emotion. With variants of Bardo we also introduce an ensemble approach with a restrictive voting rule that instead of erring sparsely through time, it errs consistently for a period of time. We show that our ensemble approach is able to make the system decisive. We hypothesize that sparsely distributed errors can harm the users' experience and it is preferable to use a model that is somewhat inaccurate but decisive, than a model that is accurate but often indecisive. A user study in which people evaluated edited versions of the D&D videos suggests that Bardo's selections can be better than those used in the original videos of the campaign. A second user study was performed following the same process and the results suggest that understanding how different error distributions affect user experience is key to develop intelligent systems able to successfully interact with humans.

Resumo

PADOVANI, Rafael Rodrigues, M.Sc., Universidade Federal de Viçosa, abril de 2018. **Seleção Automática de Música de Fundo para Jogos de Mesa.** Orientador: Levi Henrique Santana de Lelis.

A precisão de sistemas é um fator crucial que influencia a experiência do usuário em sistemas interativos inteligentes. Embora se saiba que a precisão é importante, pouco se sabe sobre o papel da distribuição de erros do sistema na experiência do usuário. Nesta dissertação é mostrado, no contexto da seleção de músicas de fundo para jogos de mesa, que algoritmos de aprendizado supervisionado podem tornar o sistema "indeciso" ao executar erros que são distribuídos de forma esparsa em uma sessão de jogo. Em seguida, é apresentado Bardo, um sistema inteligente em tempo real que seleciona automaticamente músicas de fundo para jogos de RPG de mesa. Bardo seleciona e toca como música de fundo uma música representando a emoção classificada. Com variações do Bardo também foi apresentada uma abordagem de conjunto com regra de votação restritiva que, ao invés de errar esparsamente através do tempo, erra consistentemente por um período de tempo. Foi mostrado que a abordagem conjunta é capaz de tornar o sistema decisivo. Foi gerada a hipótese de que erros distribuídos esparsamente podem prejudicar a experiência dos usuários e é preferível usar um modelo que seja um pouco impreciso, mas decisivo, do que um modelo que seja preciso, mas muitas vezes indeciso. Um estudo com usuários no qual as pessoas avaliaram versões editadas de vídeos de D&D sugere que as seleções feitas por Bardo podem ser melhores do que aquelas usadas nos vídeos originais da campanha. Um segundo estudo com usuários foi realizado seguindo o mesmo processo e os resultados sugerem que entender como diferentes distribuições de erro afetam a experiência do usuário é fundamental para desenvolver sistemas inteligentes capazes de interagir com sucesso com seres humanos.

Chapter 1

Introduction

The development of artificial agents capable of acting in accordance with human emotions is key in the development of socially acceptable agents. Tabletop games offer an excellent testbed for the development of such agents as they provide controlled scenarios in which humans display a large variety of emotions. In particular, tabletop role-playing games (RPGs) offer a rich set of scenarios for research in socially acceptable agents.

In RPGs the players interpret characters, known as player characters (PCs), in a story told by the dungeon master (DM). The DM tells the story and interprets all other characters, which are known as the non-player characters (NPC). PCs have ability scores and skills that determine if actions performed in the game (e.g., attacking an opponent) are successful.

Our work is motived by the application domain of automatic background music selection for tabletop games. For that reason we introduce Bardo, a system that uses supervised learning to automatically identify through the players' speech the emotion in the story being told in sessions of Dungeons and Dragons (D&D),[1] a storytelling-based tabletop game. Bardo chooses a background song to be played according to the identified emotion. The domain of background music selection for tabletop games is interesting because it allows one to test intelligent systems interacting with humans in a complex and yet controlled scenario.

Aiming at enhancing their immersion, D&D players often manually select the game's background music to match their story (Bergström and Björk, 2014). Unless one of the players constantly selects the background music according to the story's context, the music might not match the emotional state of the game (e.g., the PCs could be battling a dragon while a calm music is being played). Having one

[1] http://dnd.wizards.com

of the players constantly selecting the background music is not ideal, as that might distract the player from the actual game. The problem of background music selection for D&D is excellent to evaluate the quality of our emotion identification system as Bardo has to understand the emotion in the story well enough to select a proper background music. For example, if the players are going through a suspenseful moment in the game, then Bardo should play a music to match that moment.

We evaluated Bardo with an online D&D campaign (i.e., a story played through multiple sessions of the game) available on YouTube.[2] This online campaign offers a great environment for our research because the videos allowed us to use YouTube's speech recognition (SR) system to automatically convert voice to text (Harrenstien, 2009). Moreover, the songs played as background music in the original videos offer a baseline in our experiments.

We tested two supervised learning algorithms with Bardo: a lexicon-based approach that considers the density of emotions in sentences (D) and a Naive Bayes classifier (NB). We show empirically that despite the text translated by the SR system being noisy and sometimes even incomprehensible, a NB classifier is able to obtain the good overall accuracy of 64%.

Although the NB approach presented good accuracy, its emotion classifications could break the player's immersion in the game due to NB's "indecisiveness", i.e., frequent erroneous changes of the background music caused by classification uncertainty. We call these errors *short misclassifications*. We hypothesize that short misclassifications can harm user experience and that it is preferable to use classification models that are somewhat inaccurate but decisive, than models that are accurate but often indecisive. So, rather than just looking at the system accuracy, it is important to known about how the distribution of system errors affects user experience. For example, if system errors are inevitable, would we prefer a system whose errors are distributed uniformly during the system's execution or a system whose errors occur in a period of time and is accurate for the rest of its execution?

In this dissertation we study how the distribution of system errors affects user experience in the context of background music selection for tabletop games. We show that supervised learning algorithms can make the selection system "indecisive" by performing sparsely distributed errors during a game session. We also show empirically that an ensemble containing variants of NB outperforms all approaches tested and is able to substantially reduce the number of such changes while maintaining the system's overall accuracy.

[2] https://www.youtube.com/playlist?list=PLQOB_yCwC5J2m1lKNxYQfwyAhp2MRihAb

We conducted a first large-scale user study with 61 participants in which we use Bardo to select the background music of excerpts of the original videos of the D&D campaign. We compare Bardo's selections with those made by the video authors. The background music selection made by the video authors was performed as a post-processing step, while editing the videos. By contrast, Bardo performs its selection in real time, as the video is played. Thus, the video authors have the advantage of being able to select the background music knowing what will happen next in the story, thus allowing for a better selection. Despite this disadvantage, the results of our study show a clear preference by the participants for Bardo's selections.

A second user study with 37 participants was conducted following the same process of the first user study. In this study people watched videos of D&D sessions with the background music selected by an ensemble of classifiers and by Bardo's original model. We used videos in which the original model was more accurate but performed more short misclassifications than the ensemble approach. The results of this user study support our hypothesis and suggest that understanding how different error distributions affect user experience is key to develop intelligent systems able to successfully interact with humans.

1.1 Contributions

The contributions of this work are listed as follows:

- Bardo is the first system able to select background music for a real-life social event through emotion classification of people's speech.
- Our work is the first to use tabletop games as a testbed to study the interaction of an artificial agent with humans.
- A labeled dataset of sentences of the D&D sessions used in our experiments. The dataset contains the sentences of almost 5 hours of gameplay. This dataset will be made available for other researchers interested in music selection and emotion identification in speech.
- A systematic empirical study of the system's emotion classification accuracy as well as an evaluation of Bardo through a large-scale user study.
- A second user study testing our hypothesis that short misclassifications can harm user experience.

- The main contribution of this work is to show how the distribution of system errors affects user experience in the context of background music selection for tabletop games.

1.2 Dissertation Structure

This work is organized as follows. In Chapter 2 we describe Bardo, our supervised learning system. In Chapter 3 we evaluate the system with accuracy tests. In Chapter 4 we describe the empirical methodology used in a first user study and discussions about the results. In Chapter 5 we investigate, based on user study results, that the accuracy is not the only evaluation metric that can influence user perception. In Chapter 6 we introduce our ensemble approach in order to reduce the short misclassifications. In Chapter 7 we perform a second user study evaluating the improvements made by the ensemble approach in user experience. In Chapter 8 we review some relevant related work and finally, in Chapter 9 we state the conclusions of this work.

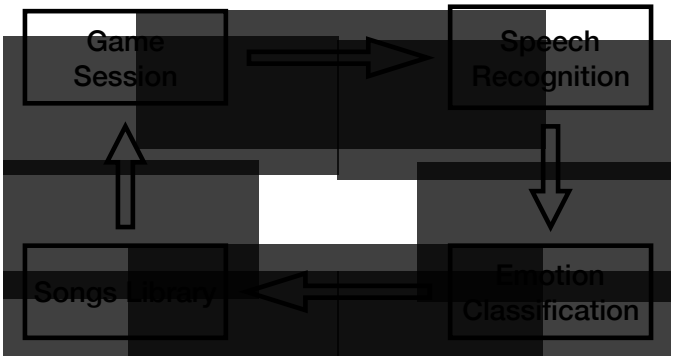

Figure 2.1: Overview of Bardo.

Chapter 2

Bardo

An overview of Bardo is shown in Figure 2.1. An SR system translates into text the players' speech signal captured from the game session. The text is then classified into one of the four emotions used in our model. Bardo selects a song from its library for the current classified emotion, which is then played as background music for the players in the game session.

Since we assume the emotional state of the game to always be classified into one of the four emotions of our model, we can treat the problem of emotion classification in tabletop games as the problem of deciding when to change states in a finite state machine (FSM). In our FSM each emotion is represented by a state and a transition from a state to any other state is possible. We assume the Calm state to be the starting state for D&D sessions. The state changes whenever the supervised learning model classifies a sentence to be of a state different than the current. Bardo requires as input a set of songs expressing each of the four emotions. Bardo plays a song provided as input for a given emotional state, and it changes the music being played whenever an emotional state change occurs.

In the next sections we describe Bardo's speech recognition step, the process used to label the dataset employed to train supervised learning models, the emotion model we introduce for D&D, and finally the supervised learning models used in our experiments.

2.1 Speech Recognition System

Since we perform our evaluation with D&D sessions from YouTube videos, the SR system we use is YouTube's system that automatically generates subtitles from speech signals from the videos. Thus, what we refer as a sentence is in fact a subtitle generated by YouTube's SR system. YouTube generates captions automatically by combining Google's automatic SR technology with its caption system (Harrenstien, 2009).

In contrast with most works that deal with text classification, the sentences Bardo produces and later classifies are often grammatically incorrect, lack structure, and are often incomprehensible. This is due to two reasons. The first is the SR system not being able to properly translate what is being said by the players. The second reason relates to our application domain, which is the fact that the SR system can add to the same sentence words said by multiple players. For example, one player could say *"I unsheathe my longsword and move toward the dragon."*, while another player says *"I run away looking for shelter."*. The SR system could capture parts of the two sentences, providing to Bardo a mixture of the two, e.g., *"I run away looking move toward the dragon."*. Due to the lack of structure in the sentences provided by the SR system, Bardo classifies the emotion of the game based on a bag of words, it does not use the structure of the sentences. Whenever referring to a sentence, we are referring to a bag of words returned by the SR system. Also, it is important to say that we use text provided by the SR system instead the audio signal itself, which contrasts with other works. This is due the fact that D&D is a funny game and the point is to entertain the players. It is very common to find moments when they are laughing even if they are in a suspenseful or agitated moment. In theses cases our classification task could be biased.

2.2 Emotion Model for D&D

Bardo's SR system generates text from what is being said by the players and the text is grouped into sentences. Each sentence is mapped by a supervised learn-

ing algorithm to an emotion in our categorical emotion model, which considers the emotions: Happy, Calm, Agitated, and Suspenseful. We note that some of the emotions in our model does not necessarily reflect a human emotion (e.g., Suspenseful), but rather the emotions that appear in stories. An example of a categorical emotion model that contains only human emotions in that introduced by Ekman 1999, which considers the following emotions: Anger, Disgust, Fear, Happiness, Sadness, and Surprise. We believe our model of "story emotions" to encompass most of the emotions that appear in D&D narratives.

2.3 The Dataset Annotation Process

In order to train the supervised learning algorithms used with Bardo, we create a labeled dataset from a D&D campaign named *Call of the Wild* (CotW) available on YouTube. We translated into text with YouTube's SR system what the PCs and DM spoke in the first 9 of the 12 episodes of CotW. We did not use the last 3 episodes because they did not have the automatically generated subtitles available. The first 9 episodes have 5,892 sentences and 45,247 words (4,614 distinct words), which result in 4 hours, 39 minutes, and 24 seconds of D&D gameplay—each episode is approximately 30 minutes long. CotW is played with D&D's 5th edition, and in addition to the DM, the campaign is played by 3 PCs, all male. One of the players was playing D&D for the first time, all the other players had played the game before. In our annotation process we label each sentence generated by YouTube's caption system according to our four-class emotion model.

The process of emotion annotation is hard in general due to its subjectivity. For example, the κ metric for inter-annotator agreement ranges from 0.24 to 0.51 and the percentage annotation overlap ranges from 45 to 64% for children's fairy tales (Alm et al., 2005). We hypothesize that emotion annotation for tabletop games is easier as the games' mechanics might offer signs of the story's emotion. Moreover, our emotion model is simpler than the model used by Alm, Roth, and Sproat Alm et al. (2005) for fairy tales—while they use 7 emotions, we use 4. We enlisted three annotators to label the CotW campaign according to our emotion model. In addition to testing our hypothesis, we expect to produce a more accurate dataset by having three instead of one annotator.

Each annotator watched all 9 episodes and labeled all sentences produced by the SR system. The annotation process was done by assuming the PCs' perspective in the story, as there could be moments that PCs and NPCs were experiencing

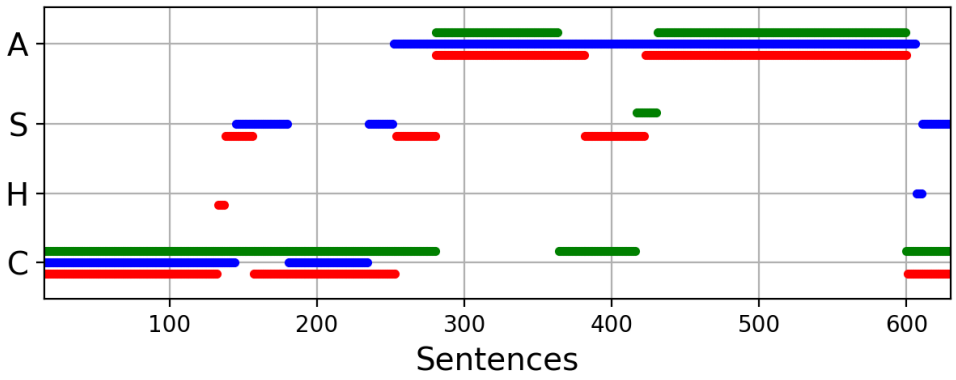

Figure 2.2: Inter-annotator agreement of episode 1 of CotW. The x-axis shows the sentences in the episode, ordered by appearance and the y-axis the four emotions: Agitated (A), Suspenseful (S), Happy (H), Calm (C).

different emotions. For example, in one of the episodes the PCs are trying to break into their enemy's house, which results in a set of suspenseful scenes. By contrast, until the NPCs guarding the house discover that the PCs trying to break in, it is just another day at work for them.

Should two annotators agree on the label of a sentence s, then s is labeled according to the two annotators. One of the annotators watched the videos again to break the ties (sentences for which each annotator attributed a distinct label).

The inter-agreement κ metric of our three annotators was 0.60 and the percentage overlap of the annotations was 0.79. These numbers are higher than what is usually observed in the literature (e.g., emotions in children's fairy tales). This result support our hypothesis that emotion annotation in tabletop games might be easier due to the clues offered by the game mechanics and to the simpler emotion model. Figure 2.2 shows details of the annotated emotions of all sentences in the first episode of CotW. The x-axis shows the sentences of the episode, ordered by appearance, and the y-axis the four emotions: Agitated (A), Suspenseful (S), Happy (H), Calm (C). Each color shows the emotion attributed by one of the annotators to a sentence. For example, in the beginning of the episode all annotators agreed that the emotion Calm represents well that moment of the story. A disagreement among the three annotators is observed around sentence 400, where one annotator believes it to be an Agitated moment of the story, while others believe it to be either Suspenseful or Calm. After analyzing the video again, the tie-breaker annotator decided that the moment around sentence 400 is Agitated.

#	Sentence	Emotion
s_{30}	word on the street is that there are	Calm
s_{31}	some people who want your blood just	Suspense
s_{32}	like you wanted theirs in the last part	Suspense
...	...	...
s_{131}	it again but hold on rorey shot once oh	Agitated
s_{132}	okay I get two shots another 28 to it	Agitated
s_{133}	yeah it's 12 inch the arrow second arrow	Agitated
s_{134}	hits him and he falls crumples hits a	Agitated
s_{135}	spacing it's a stone it dies on the spot	Agitated

Table 2.1: An excerpt of our D&D dataset.

The dataset contains 2,005 Agitated sentences, 2,444 Suspenseful, 38 Happy, and 1,337 Calm. Although Happy sentences are rare in CotW, the emotion may appear more frequently in other D&D campaigns. For example, it is common for the story to start with the PCs meeting in a tavern, and such scenes might require Happy songs.

Table 2.1 shows an excerpt of our dataset, from sentence s_{30} to s_{32} and from sentence s_{131} to s_{135} of an episode (see column "#"). We use the subscript of a sentence to denote the index in which the sentence appears in an episode. For example, s_{134} occurs immediately after s_{133}. The second column of the table shows the sentences and the third the sentences' emotion according to the annotators. We highlight the lack of grammatical structure and meaning in some of the sentences, e.g., s_{131}: *it again but hold on rorey shot once oh.*

Next, we present the two supervised learning algorithms we used with Bardo.

2.4 Density-Based Classifier (D)

Our Density-based method (D) classifies a sentence s by counting the number of words associated with an emotion—the emotion with largest count is chosen as the emotion of s. The associations between words and emotions are provided by the NRC Emotion Lexicon (Mohammad and Turney, 2013), which has approximately 14,000 English words annotated with eight emotions: Anticipation, Anger, Joy, Fear, Disgust, Sadness, Surprise and Trust. Since the NRC lexicon uses a emotion model that is different than ours, we need to find a mapping between NRC's emotions and Bardo's. For example, a mapping would count all words in the Joy, Surprise, and Trust emotions of NRC as words of Bardo's Happy class.

We devised a method to choose the emotion mapping to be used. Since we

need to map 7 NRC emotions into 4 Bardo emotions, there are $4^7 = 16{,}384$ different mappings to be tested. We use a brute-force search procedure that tests each one of the 16,384 possible mappings and returns the mapping that performs best in a leave-one-episode-out cross-validation procedure in the training set. The training set contains 8 episodes and the test set the remaining episode. This search procedure for the best-performing mapping on the training set is D's supervised learning phase.

2.4.1 Classification Threshold

In addition to the automatic mapping of NRC emotions into Bardo's emotions, another enhancement we use is a density threshold d_t. Bardo only switches states in its FSM when D's emotion with the largest count exceeds d_t. This is because emotion transitions are somewhat rare in D&D sessions (for example, see Figure 2.2). Thus, ideally Bardo switches states only when there is a clear sign of emotion change. We expect the density threshold d_t to avoid unnecessary and incorrect state changes. The value of d_t is also determined altogether with the emotion mapping in the leave-one-episode-out cross-validation.

2.4.2 Sliding Window

Since the number of words in a sentence is usually small (frequently smaller than 10), there is not much information within a sentence to accurately identify the story's emotion. Thus, to calculate the emotion density of sentence s_i, we use a sliding window of size k that is composed by all sentences from s_{i-k-1} to s_i. The sliding window allows D to account for more data while identifying the story's emotion. As an example, consider the task of predicting the emotion of sentence s_{135}, shown at the bottom of Table 2.1. Instead of performing D's counting procedure in sentence s_{135} alone, if using a sliding window of size 3, we perform the counting procedure in the union of words of sentences s_{135}, s_{134}, and s_{133}. Note that sentences from s_0 to s_{k-1} are classified with a window of size smaller than k.

The size of the window is also found in the cross validation procedure we perform in the training set. We test 16,384 different emotion mappings, 20 threshold values, and 5 sliding window sizes, resulting in 1,638,400 parameter values tested during D's training.

2.5 Naive Bayes Classifier (NB)

Another algorithm we consider for classifying sentences for Bardo is Naive Bayes (NB) (McCallum and Nigam, 1998). In NB one computes the probability of a sentence s being of an emotion e, for all possible e. NB returns for s the e with highest probability. Let E be the set of all four emotions considered by Bardo. Instead of using the probability of a sentence s being of emotion $e \in E$, denoted $P(e|s)$, NB uses a value that is proportional to $P(e|s)$,

$$P(e|s) \propto \log P(e) + \sum_{w \in s} \log P(w|e)\,.$$

Here, $P(e)$ is the probability of encountering a sentence of emotion e in a D&D session. $P(e)$ can be approximated by $\frac{N_e}{N}$, where N_e is the total number of sentences of emotion e in our training set, and N is the total number of sentences in our training set. $P(w|e)$ is the probability of encountering the word w in a sentence of emotion e. $P(w|e)$ is approximated by $\frac{count(w,e)}{count(e)}$, where $count(w, e)$ is the total number of times the word w appears in sentences of emotion e and $count(e)$ is the total number of words that appears in sentences of emotion e in our training set. We use the add-one smoothing scheme to avoid over-fitting and the sum of logs to avoid underflow issues (Manning et al., 2008).

We also use a threshold parameter with NB. That is, Bardo only switches states in the FSM if NB's classification exceeds a threshold. Also, similarly to the D classifier, NB also uses a sliding window. We find both the threshold and the sliding window values in a leave-one-episode-out cross validation procedure in the training set, as described for the D. We test 5 window sizes and 26 threshold values with NB, resulting in 130 values tested.

2.5.1 Feature Selection

We perform feature selection to remove from our dataset noisy words that might reduce the classification accuracy. We use only the k most discriminative words for each emotion during classification. That is, words whose presence/absence in a sentence provide useful information about the actual emotion of the sentence. We use mutual information (MI) (Manning et al., 2008) to measure how discriminative a word is. Similarly to the other parameters, the value of k is also chosen during the cross-validation procedure. Since NB performed better than D in preliminary experiments, we use feature selection only with variants of NB. When performing feature selection, in addition to the 130 values tested with NB, we test other 100

values of k, resulting in a total of 13,000 values for the Naive Bayes approach that uses all enhancements: threshold, sliding-window, and feature selection.

Once NB and D finish training, they are able to instantaneously classify the story's emotion for a given bag of words provided by Bardo's SR system.

Chapter 3

Classification Evaluation

In this section we present the accuracy results of the following variants of D and NB: D with no enhancements (denoted as D), D with sliding window (DS), D with threshold (DT), D with sliding window and threshold (DST), NB with no enhancements (denoted as NB), NB with sliding window (NS), NB with threshold (NT), and NB with sliding window and threshold (NST). We also use the NB variants with feature selection, denoted with a letter "K", i.e., NBK, NSK, NTK, and NSTK. We separate each episode to be tested and train the algorithms on the other episodes (e.g., when testing an algorithm on episode 1, we train it with episodes 2–9 and the resulting model is applied to episode 1).

Table 3.1 presents the accuracy obtained by the algorithms ("Alg.") in each episode in terms of the percentage of sentences correctly classified by the algorithms. The "Avg." column shows the algorithm's average percentage accuracy across all episodes. Numbers highlighted in bold indicate the best performing variant amongst D variants, NB variants, and NB variants with feature selection for a given episode. For example, row D has bold numbers for episodes 1 and 7, indicating that amongst all D variants, D performed best in these two episodes. We highlight the background of a cell if the number in the cell represents the best result across all algorithms. For example, NS and NST were the best-performing algorithms in episode 9, while NSTK was the best-performing algorithm in episode 3.

3.1 Accuracy Analysis

Overall the Naive Bayes variants perform better than the density-based ones. NS and NSK perform best on average, with NST and NSTK being slightly less accurate. The feature selection approach improves the accuracy only of the simplest NB

Alg.	Episodes									Avg.
	1	2	3	4	5	6	7	8	9	
D	**39**	38	30	42	26	24	**35**	39	35	34
DS	30	**63**	46	**57**	**63**	**53**	25	**47**	**57**	**49**
DT	34	17	35	40	38	06	04	40	35	27
DST	34	**63**	**49**	**57**	62	22	26	46	54	46
NB	46	29	61	28	24	23	0	42	46	33
NS	**64**	**62**	**76**	71	54	**69**	**44**	**59**	**79**	**64**
NT	41	48	44	49	46	50	34	44	52	45
NST	61	61	**76**	**72**	**56**	61	43	58	**79**	63
NBK	41	48	47	51	44	53	36	40	57	46
NSK	**71**	**57**	79	**69**	**56**	59	55	**59**	**76**	**64**
NTK	41	48	47	51	44	53	36	40	57	46
NSTK	**71**	56	**80**	**69**	**56**	**66**	**59**	38	75	63

Table 3.1: Prediction accuracy in % of variants of NB and D.

classifier. NB has an average accuracy of 33% while NBK has an average accuracy of 46%. Feature selection does not improve the classification accuracy if the threshold and/or the sliding window approaches are used. The sliding window approach improves the prediction accuracy in all scenarios tested: DS is more accurate than D, DST is more accurate than DT. Similar results occur with Naive Bayes with or without feature selection and with or without threshold.

The threshold approach substantially improves the prediction accuracy of the simplest NB (NT is substantially more accurate than NB), and it does not reduce prediction accuracy of the other Naive Bayes variants. The threshold approach can decrease the prediction accuracy for density-based approaches. D is more accurate than DT and DS is slightly more accurate than DST. This is because in some of the episodes the threshold value returned by the cross-validation procedure does not generalize well. For example, D is more accurate than DT in episodes 2 and 7, and DS is more accurate than DST in episode 6.

Episode 7 is difficult to all algorithms, with NS being only 44% accurate in that episode and NB being wrong in all its prediction attempts. In episode 7 the PCs talk to different groups of NPCs trying to make an alliance for an imminent war. This episode contrasts with the others because the DM and PCs role play their characters without using the game mechanics. By contrast, the other episodes have scenes in which the PCs are stealthy or aggressive, which require the use of the game mechanics. The NB-based algorithms often rely on identifying the use of game rules to make accurate predictions. For example, episodes 3, 4, and 9 have a mixture combat and stealthy scenes and NS is able to accurately detect them, as

can be observed by the algorithm's accuracy in those episodes.

3.2 Discriminative Words for NB

	Agitated		Calm		Happy		Suspenseful	
#	Word	MI	Word	MI	Word	MI	Word	MI
1	damage	16.84	name	7.41	comfort	4.48	door	4.64
2	hit	10.16	tribe	4.92	jeez	3.26	inside	2.73
3	fire	5.16	find	4.39	dare	3.26	open	2.53
4	11	4.29	tree	2.87	O'Meara	3.26	stealth	2.14
5	roll	3.89	campfire	2.61	relieve	3.26	slaves	1.45
6	arrow	3.86	barbarian	2.43	rebuilding	3.26	guard	1.41
7	attack	2.94	bird	2.35	coat	3.26	keys	1.39
8	14	2.89	fish	2.33	Restless	3.26	wagon	1.30
9	points	2.82	okay	2.27	foresees	3.26	hot	1.28
10	5	2.80	well	2.26	Darian	3.01	gifts	1.04
11	turn	2.69	bond	2.09	jacket	2.83	watch	1.04
12	13	2.62	all	1.91	carriage	2.48	slip	1.04
13	initiative	2.58	wife	1.75	too	2.32	casting	1.04
14	sword	2.46	love	1.71	slippers	2.07	charm	1.04
15	4	2.36	drew	1.71	replace	2.07	ji	1.04
16	head	2.07	day	1.67	comforting	2.07	friday	1.04
17	guys	1.99	wise	1.59	comforted	2.07	CJ	1.04
18	seven	1.94	short	1.59	chocolate	2.07	OMG	1.04
19	tries	1.94	destiny	1.59	friends	1.87	tj's	1.04
20	shot	1.92	wine	1.59	wrap	1.87	fridays	1.04
21	Wow	1.92	gonna	1.58	check	1.62	small	0.93
22	7	1.92	over	1.55	Oh	1.60	rations	0.84
23	gets	1.81	20	1.54	tent	1.56	backwards	0.84
24	takes	1.77	been	1.49	ok	1.56	line	0.84
25	2	1.74	party	1.47	alive	1.49	wrong	0.84
26	spear	1.71	alright	1.47	kids	1.49	snow	0.84
27	rolled	1.68	start	1.45	thank	1.38	cat	0.84
28	1	1.56	more	1.35	battle	1.33	believe	0.84
29	10	1.55	campaign	1.34	our	1.33	lift	0.84
30	six	1.51	warm	1.34	boy	1.25	scream	0.84
31	arm	1.49	build	1.34	best	1.25	210	0.84
32	dang	1.49	hunt	1.34	god	1.22	smell	0.84
33	that's	1.46	sensed	1.34	son	1.19	knives	0.84
34	what's	1.43	trying	1.26	rolling	1.19	baller	0.84
35	everyone	1.41	matter	1.21	need	1.01	perception	0.81

Table 3.2: The 35 words with largest MI values (multiplied by 10^3) for each class.

We inspected the set of most discriminative words for the NB classifier. The notion of discriminative words can be formalized in terms of mutual information (MI) (Manning et al., 2008), which attributes a score of how discriminative a word is for a given class. Higher MI values indicate more discriminative words. Table 3.2 shows the 35 most discriminative words for each emotion in our model. The table shows the words order by MI values, which were multiplied by 10^3 and truncated to two decimal places for ease of presentation.

The Agitated class is easily identified by words related to the game mechanics. For example, the very large values of 16.84×10^{-3} and 10.16×10^{-3} for *damage* and *hit*, which are words related to combat scenes, demonstrate the importance of game mechanics while identifying the emotion of a scene. In addition to *damage* and *hit*, NB learned that words such as *roll*, *initiative*, and *attack*, which are also related to D&D's mechanics, are important to identify an Agitated scene. Interestingly, Agitated scenes are also marked by many rolls of dice, which are required to verify if the PCs intended actions are successful. The classifier also learned that numbers are in general a good marked of the Agitated class.

The Suspenseful scenes are usually related to the PCs being stealthy (e.g., trying to break into someone's house), which require the PCs to perform dice rolls. However, Suspenseful scenes can also include tense negotiations between PCs and NPCs, which do not necessarily required rolling dice, but only role playing the characters. This justifies numbers being discriminative for Agitated but not for Suspenseful. We still observe words related to the game mechanics such as *perception* in the Suspenseful column.

The words with largest MI-values for the Calm emotion are related to the scenes in which the PCs are safe and planning their next step in the campaign (e.g., *campfire* and *party*) or when the PCs are amicably learning the *names* of companions and allies. The words with largest MI-values for Happy are related to scenes in which the PCs are with their families and friends (e.g., *friends* and *comfort*). We also note that names of people are discriminative for the Happy emotion (e.g., *O'Meara* and *Darian*).

It is interesting to note that the classifier learns how different interjections are discriminative for different emotions. For example, if a player says *Wow*, then it is likely that the sentence is related to an Agitated scene. By contrast, *OMG* ("Oh my God") indicates a Suspenseful and tense scene, and *Oh* a Happy scene. The use of interjections can certainly vary from a group of players to another. However, these results suggest that the interjections are used consistently for different emotions by a fixed group of players.

Chapter 4

User Study 1: an Accuracy Evaluation

The results presented so far show how accurate different learning models employed by Bardo can be, but it offers no insights on how humans perceive the selections made by the algorithm—for that we conduct a user study. In this first user study we are mainly interested in verifying how humans perceive the background music selection performed by Bardo in video excerpts of CotW for which Bardo's classification algorithm has different levels of accuracy. We are also interested in comparing the background music selection performed by Bardo with that performed by the video authors.

4.1 Empirical Methodology

We perform our study online with edited videos from the CotW D&D campaign. Namely, we replace the original background songs with the songs selected by Bardo employing NS, which was the best performing algorithm together with NSK in our accuracy experiments.

Instead of using the original background music, we used Bardo to select to background music of 5 excerpts of the CotW campaign videos. NS is trained with episodes different than the one from which the excerpt is extracted. For example, if an excerpt is extracted from episode 1, then NS is trained with all episodes but 1. Each excerpt is approximately 2 minutes long and contains representative scenes of a typical D&D session. Also, we selected the excerpts for which NS's accuracy varied considerably. This way we are able to relate the participants preferences with NS's accuracy. Finally, to avoid possible biases in our excerpt selection, the average

Table 4.1: We used three different instrumental music tracks for three emotions observed in the experiment.

Emotion	Song	Artist
Calm	Call of the Raven	Jeremy Soule
Suspenseful	Hurricane Suite	Naruto Shippuden OST I
Agitated	Open the Gates of Battle	Casey Martin

Table 4.2: Each video in the experiment corresponds to an excerpt of the episodes that was randomly distributed, so we have the interval where they happen as well.

Excerpt	Episode	Starts	Finishes
1	1	15:37	17:13
2	5	20:20	21:33
3	1	11:37	12:36
4	3	12:45	14:45
5	4	10:04	11:41

accuracy of NS in the 5 excerpts is similar to NS's average accuracy in the entire dataset, which is approximately 64% (see Table 3.1).

We use as baseline the background music selection made by the video authors while editing the CotW videos. However, instead of using the videos with their original background music, to perform a fair comparison, we edit the videos to use the same set of songs Bardo uses. That is, we replace what we consider to be the Calm songs in the original videos by our Calm song, and so on. We use a set of songs different than the ones used in the original videos because we do not have easy access to the original songs. The songs we use are similar to the originals in the sense that they all fit in the D&D's fantasy genre. Also, we chose a set of songs that clearly represent different emotions so the participants can easily notice the background music changes. One caveat of using the original videos edited with our songs as baseline is that the video authors might have selected different emotions in their editing process if they had used our songs instead of theirs. In this experiment V1 is an excerpt starting at 15:37 and finishing at 17:13 of episode 1 of CotW; V2 starts at 20:20 and finishes at 21:33 of episode 5; V3 starts at 11:37 and finishes at 12:36 of episode 1; V4 starts at 12:45 and finishes at 14:45 of episode 3; V5 starts at 10:04 and finishes at 11:41 of episode 4 (see Table 4.2).

Another caveat of this approach is that it requires us to label the emotions of the background songs of the video excerpts so that they could be replaced by our songs. The labeling of the original songs was performed by two independent annotators. That is, two annotators watched the 5 CotW excerpts and identified

the emotion of the background music chosen by the video authors. This annotation process was straightforward as only one disagreement occurred while deciding the emotion of a background song. This disagreement was resolved by a third independent annotator.

Note that the authors of the videos had much more information than Bardo to make their background music selection. This is because the video authors selected the background music as a post-processing step, after the videos were recorded. As such, they could spend as much time as needed to select the background songs. Moreover, the video authors knew what was going to happen in the story and could use information such as the story's tension arc to bias their background music selection. By contrast, Bardo did not know what was going to happen in the story and it had to make its decisions in real time.

The video excerpts we use have no sentences of the Happy emotion, thus we use three songs in our experiment, one for the Agitated scenes, one for the Suspenseful, and one for the Calm. We used the song *Call of the Raven* by Jeremy Soule for Calm, *Hurricane Suite* by Naruto Shippuden OST I for Suspenseful and *Open the Gates of Battle* by Casey Martin for Agitated, as shown in Table 4.1. Each participant listened to excerpts of all three songs after answering our consent form and before evaluating the videos. The participants were told that they would evaluate the background music used in several video excerpts of a D&D campaign, and those three songs were the only songs that would be used as background music. We believe that we reduce the chances of a participant evaluating the quality of the songs instead of the song selection procedure by telling them which songs will be used as background music.

After listening to the songs each participant watched two versions of the same video excerpt, one with the video authors' emotion selection of background music and another with Bardo's. The order in which the videos appeared was random to prevent ordering biases. We included a brief sentence providing context to the participant, to ensure they would understand the story being told in each excerpt. The participants could watch each video as many times as they wanted before answering the following multiple choice question.

- Which video has the most appropriate background music according to the context of the story? (choose one of the following options).

    1. Video 1.
    2. Video 2.

Method	Video Excerpts				
	V1	V2	V3	V4	V5
Bardo	60.60	55.73	14.75	73.77	62.30
Baseline	16.42	22.95	60.65	11.48	9.84
Tie+	16.42	9.84	21.32	6.55	13.11
Tie-	6.56	11.48	3.28	8.20	14.75
NS	80.55	20.00	52.63	100.00	86.66

Table 4.3: Comparison between accuracy and user preference

3. The background music used in both videos are equally appropriate.
4. The background music used in both videos are equally inappropriate.

After marking their answer, the participants would evaluate another pair of excerpts. The order in which the pairs of excerpts appeared was also random. The participants answered a demographic questionnaire after evaluating all excerpts.

Our experiment was advertised in D&D communities in the social media.[1] In total we had 66 participants, 57 males, 8 females, and 1 other, with average age of 26. All participants had at least some experience playing RPGs. We have removed the answers of 3 participants who reported to have only basic proficiency in English (the language used in the videos and in the study), and of 2 participants who reported problems with their audio system during the experiment. We report the results of the remaining 61 participants, which resulted in 305 answers (5 pairs of videos for each participant).

4.2 User Study Results

The videos edited by Bardo were preferred 163 times by the participants, while the Baseline was preferred 74 times, and the approaches tied 68 times. The difference between Bardo and Baseline is significant according to a two-sided binomial test ($p = 7.325 \times 10^{-9}$). Table 4.3 shows the detailed results for all 5 excerpts used in our study. The upper part of the table shows the percentage of times the participants chose the videos edited by Bardo, by the Baseline, and the percentage of times the participants thought the videos to be equally appropriate (Tie+), and equally inappropriate (Tie-). For example, for the second excerpt (V2), the participants preferred Bardo's video in 55.73% of the answers, and for the third excerpt (V3) the participants preferred the Baseline video in 60.65% of the answers. The last row of

[1] https://goo.gl/forms/llNqkLSwUTuhSU4h1

the table shows NS's accuracy in each excerpt. The highlighted cells show the best performing approach (Bardo or Baseline).

4.3 Discussion

The results of our user study shows a clear preference for the video editing provided by Bardo over that provided by the video authors. Despite the caveats of our Baseline, this result demonstrates the potential of Bardo for enhancing the players' experience in tabletop games. We observe that there is some correlation between NS's accuracy with people's preferences. For example, NS has 100% accuracy in V4, which is also the excerpt Bardo performed best: it was preferred in 73.77% of the answers for V4. If we add Tie+ to this percentage, we obtain a positive answer in 80.32% of the cases for Bardo. On the other hand, NS has an accuracy of only 20% and Bardo a preference of 55.73% in V2, while NS has an accuracy of 52.63% and Bardo a preference of only 14.75% in V3.

The results for V2 and V3 suggest that factors other than NS's accuracy can affect the participants preferences. In V2 Bardo mistakenly chooses Agitated instead of Suspenseful for most of the excerpt. However, since the excerpt depicts a Suspenseful scene with some action, most of the participants were fine with the Agitated song selected by Bardo. The excerpt V3 depicts a scene in which a group of barbarians (PCs) go out on a hunt for food, and after killing an elk, a bear sneaks up upon them. Bardo selects the Agitated emotion due to the action of hunting the elk, and it eventually switches to Suspenseful due to the mechanics of the game used before the bear appears (the DM asked the PCs to *roll their perception*, which is usually related to suspenseful moments). Although a brief switch from Agitated to Suspenseful is correct according to the dataset, Bardo's timing is not good and it only switches to Suspenseful after the PCs had engaged in combat with the bear, which is wrong according to the dataset, as the combat indicates another Agitated moment. The Baseline appears to use its knowledge of what is going to happen in the story and selects Calm for the combat with the elk and switches to Agitated once the bear appears—the Baseline raises the stress of the background music as the DM raises the danger the PCs face. Since Bardo has no knowledge of the future, it has no means of using similar technique.

We had a serendipity moment when analyzing the results of excerpt V1. V1 starts at sentence 352 and finishes at sentence 388 of episode 1. Figure 2.2 shows that there was a disagreement amongst the annotators during the labeling process

of V1. It was eventually decided that all sentences in V1 are Agitated. However, Bardo made a transition from Agitated to Suspenseful, as originally suggested by one of the annotators. After watching the video edited by Bardo, all annotators were convinced that the system made a better selection than the annotators themselves. By investigating the bag of words used to classify this transition, we discovered that Bardo performs the switch from Agitated to Suspenseful when the sliding window stops considering the words *"comes the damage does 14 damage to it"* and considers the last words spoken by the DM, which were *"we're starting to feel sick ill"*. By not considering the word *damage* and the number *14* the probability of classifying the bag of words given by the sliding window as Agitated decreased considerably, and the words *sick* and *ill* contributed to increase the probability of the Suspenseful class.

	Predicted			
Labeled	Agitated	Calm	Happy	Suspenseful
Agitated	1,596	28	0	381
Calm	230	493	0	614
Happy	4	0	0	34
Suspenseful	484	274	0	1,686

Table 5.1: Confusion matrix of the NS classifier in the CotW campaign.

Chapter 5

Beyond Classification Accuracy

While the results of our first user study indicate that Bardo's classification accuracy influences how humans perceive Bardo's background music selections, the results also suggest that there are other evaluation metrics that can influence people's perceptions. In this section we discuss two of such evaluation metrics. Namely, we discuss the different types of short misclassifications and the distribution of short misclassifications across a game session.

5.1 Sparsely Distributed Errors

Depending on the application domain, some classification errors might have different costs than others (Provost and Fawcett, 2001; Drummond and Holte, 2006). For example, in our application domain, one can argue that Bardo will break the player's immersion by mistakenly selecting a Happy song for a Suspenseful scene. However, maybe the players will not be so negatively affected if Bardo selects a Calm song for a Suspenseful scene.

We present in Table 5.1 the NS classification results in terms of a confusion

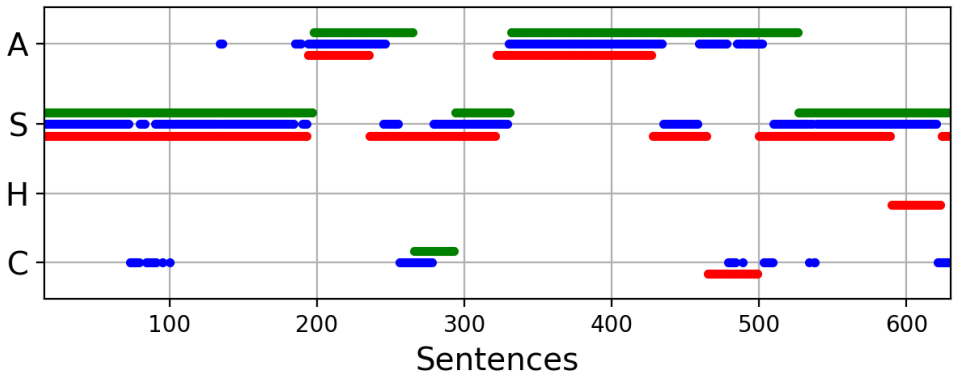

Figure 5.1: Comparison between the labels as provided by the annotators (red lines), NS classifier (blue lines), the ENS (green lines) for an excerpt of episode 6.

matrix to verify the most common short misclassifications NS performs. Although we present the confusion matrix only for the NS classifier, the other classifiers have similar matrix (including the methods we introduce in Chapter 6 below). The columns represent the number of sentences classified by NS for each class, while the rows represent the number of sentences with a given label. For example, NS classified 1,596 sentences correctly for the Agitated class, and it predicted 230 Calm sentences to be Agitated.

A common short misclassification observed in Table 5.1 is the prediction of Agitated or Calm for Suspenseful sentences; this type of error occurred $484 + 274 = 758$ times. Another common short misclassification is the prediction of Suspenseful for Agitated and Calm sentences; this type of error occurred $381 + 614 = 995$ times. Note that the short misclassification of Agitated for Calm and Calm for Agitated are much less common; this type of error occurred only $230 + 28 = 258$ times. Almost no short misclassifications occur with the Happy class.

We believe the distribution of short misclassifications shown in Table 5.1 to be beneficial to our task. This is because the Suspenseful class is involved in most of the classification errors. We believe that playing Calm or Agitated songs in Suspenseful scenes or playing Suspenseful songs in Calm and Agitated scenes to not negatively affect the players' immersion as much as other types of short misclassification errors. Suspenseful scenes often occur as a transition between Calm and Agitated ones in the CotW campaign. As a result, Suspenseful songs are often acceptable in scenes of other classes and songs of other classes are often acceptable in Suspenseful scenes. This intuitive analysis is supported by our user study (Chapter 4).

5.2 Short Misclassifications

In addition to not accounting for the impact different misclassification types can cause on the players' immersion, the accuracy of a system does not provide information about the distribution of short misclassifications across the game session. The distribution of short misclassifications is important because even an accurate system can disturb the player's immersion if the system switches the background music "too often".

For example, assuming a classification accuracy of 70% and an excerpt of 100 sentences, if the errors are uniformly distributed across the sentences, Bardo will mistakenly switch the background music for a few seconds and then switch back to a song that reflects the actual emotion of the story after every 4 sentences. This frequent and abrupt changes can harm the players' immersion. It is likely that Bardo performs better if using a classification system with lower accuracy that errs systematically in an entire scene instead of erring sparsely throughout the game session.

This idea of having the classification errors sparsely versus compactly distributed in a game session is illustrated in Figure 5.1. The red lines show the labels of sentences of episode 6, the blue lines show NS's classifications, and the green lines show the classification of an algorithm we introduce in Chapter 6, which we call ENS. Although NS is more accurate than ENS in episode 6 (NS has an accuracy of 69% and ENS an accuracy of 59%), we hypothesize that NS is more likely to harm the players' immersion than ENS. This is because ENS is consistent in the sense that it does not change its classification as often as NS does.

NS quickly and abruptly switches between Suspenseful and Agitated around sentence 130 of the episode, which will most likely harm the players' immersion due to the frequent and erroneous changes of the background music that the misclassifications imply. Perhaps even worse for the players' immersion, NS quickly and abruptly switches between the Agitated and Calm emotions around sentence 500. We call these quick and abrupt misclassifications such as the ones performed by NS in episode 6 as *short misclassifications*.

The problems caused by short misclassifications can be mitigated by having the current song fading out (slowly reducing its volume until it is inaudible) and then having the song to be played next to fade in (starting with song being inaudible and slowly increasing its volume). This way, if the classifiers switches back to the previous class during the fade-out process, the previous song could fade back in. While audio engineering techniques such as audio fade might reduce the problems

caused by short misclassifications, such techniques can still interfere with the players' immersion by frequently fading the songs in and out.

In the next section we discuss how to develop a ensemble of classifiers, i.e., a classifier that minimizes short misclassifications.

Chapter 6

Ensemble Approach (ENS)

We implement an ensemble approach (Hansen and Salamon, 1990; Dietterich, 2000), which we call ENS, with the goal of reducing the short misclassifications. Given a set of classifiers C, ENS classifies the emotion of the first sentence in the episode according to a majority vote rule of the classifiers in C; ties are broken randomly. For any other sentence represented by a bag of words w, Bardo with ENS only transitions from one emotion to another emotion e if all classifiers in C agree that w is of emotion e. That is, for the first sentence in the episode ENS uses a majority vote rule and for every other sentence ENS uses a unanimity rule, which is a special case of the voting rule introduced by Xu et al. 1992. In Xu et al.'s unanimity voting rule a sample is "rejected" by the ensemble if all classifiers do not agree on the sample's label. In our case, if the classifiers do not agree on the sample's label, we assume the emotion has not changed in the game's story.

In Section 6.1 we perform a theoretical analysis of ENS under some simplifying assumptions. In Section 6.2 we perform an empirical study of two variants of ENS.

6.1 Theoretical Analysis

Our analysis is divided into two parts. First, we show that ENS is expected to induce fewer emotion transitions than a single classifier. Then, we derive the sufficient conditions for ENS to be more accurate than a classifier in expectation.

6.1.1 Reduced Number of Transitions

Let E be a set of emotions, C the set of classifiers used with ENS, and $n = |C|$. We assume the probability of classifying a bag of words w of emotion e_j as being

of emotion e_i to be the same for all w of emotion e_j. We denote such probability as $p_c(e_i|e_j)$. Similarly, we denote as $p(e_i|e_j)$ the probability of all classifiers in C classifying any w of emotion e_j as being of emotion e_i. We write p_c and p instead of $p_c(e_i|e_j)$ and $p(e_i|e_j)$ whenever e_i and e_j are clear from the context.

Two events can occur in our problem: (i) ENS correctly classifies the current emotion e_j or (ii) ENS incorrectly classifies the current emotion e_j. In this part of the analysis we assume these events to be independent (*i.e.,* the chances of a bag of words w_{t+1} being of emotion e is independent of the emotion of w_t). Assuming independence, the expected number of trials ENS performs for event (i) to occur is given by

$$B(C, e_j) = p(e_j|e_j)^{-1}.$$

Similarly, the expected number of trials ENS performs for event (ii) to occur is given by

$$R(C, e_j) = b(e_j)^{-1}.$$

Here, $b(e_j) = \sum_{\substack{e \in E \\ e \neq e_j}} p(e|e_j)$. $R(C, e_j)$ is the expected number of trials until all classifiers agree on an emotion different than e_j. We write b instead of $b(e_j)$ whenever e_j is clear from the context. Note that Bardo using ENS with $C = \{c\}$ is equivalent to Bardo using c alone.

The following observation states that Bardo using ENS using C with $n > 1$ is expected to change emotions less frequently than Bardo with any of its classifiers individually.

Observation 1 *For $C = \{c_1, c_2, \cdots, c_n\}$ and a subset of size one $C' = \{c\}$ with c being any classifier in C, we have that $B(C, e_j) \geq B(C', e_j)$ and $R(C, e_j) \geq R(C', e_j)$.*

If all classifiers in C are identical, $B(C, e_j) = B(C', e_j)$ and $R(C, e_j) = R(C', e_j)$, as the classifiers will always agree on the emotion transitions. If the classifiers in C are independent, then $p(e_i|e_j) = \prod_{c \in C} p_c(e_i|e_j)$ and the values of B and R will grow quickly with the size of C. Large B and R values mean that Bardo switches the background music less often (*i.e.,* small $T(C, W)$ values). Since the number of emotion transitions $T(W)$ is small in practice, by reducing $T(C, W)$ one is expected to reduce the value of $SM(C, W)$, our surrogate for short misclassifications.

Example 1 *Consider an example in which all classifiers in C are independent, the probability of each classifier correctly identify the current emotion is 65%, and the probability of incorrectly identify each of the other emotions is 15%, 15%, and 5%. For $n = 4$, $R \approx 980$ and $B \approx 6$. If facing the event of misclassifying an emotion, Bardo will misclassify the emotion for an average of 6 bag of words before switching to the correct emotion. Then, the system is expected to correctly classify the next 980 bag of words. Since an episode is approximately 700 bag of words long, it is more likely that the emotion in the story changes or the game session finishes before the system misclassifies the current emotion.*

6.1.2 Improved Overall Accuracy

As one adds distinct classifiers into C, ENS will require an increasingly large number of trials before detecting an emotion transition. In particular, if the number of trials is larger than the number of sentences in a *scene*, then ENS might miss the emotion transition entirely. A scene is an excerpt of a game session composed of bag of words with the same emotion.

Definition 1 (Scene) *Let $S = \{w_i, w_{i+1}, \cdots, w_{j-1}, w_j\}$ be a subset of W with $i \geq 1$ and $j \leq m$. Also, all $w \in S$ have the same emotion e and the emotions of w_{i-1} and w_{j+1} are different from e (if $i > 1$ and $j < m$). We call S a scene.*

Next, we derive the sufficient conditions for ENS to be more accurate than a single classifier in a scene S in expectation. The execution of ENS within S can be modeled with two states: X and Y. ENS is in X at time step t if it correctly identified the emotion of w_{t-1}. Since w_{t-1} and w_t have the same emotion (they belong to the same scene), if the classifiers do not agree on an emotion, then ENS correctly classifies w_t by assuming it has the same emotion as w_{t-1}. ENS is in Y at time step t if ENS classified w_{t-1} as being of an emotion different from w_t's actual emotion. We assume ENS starts in Y as ENS is expected to start a scene in Y much more often than in X. ENS starts in X if the classification performed by ENS correctly identifies the emotion of the first bag of words in the first scene of an episode, or if it misclassifies the last bag of words of a scene and the predicted emotion is the emotion of the next scene.

We define as q the size of a scene S and model the expected number of bag of words correctly classified by ENS in S as $F_Y(q)$, which can be written with the

following recurrence,

$$F_Y(q) = p(F_X(q-1)+1) + (1-p)F_Y(q-1) \tag{6.1}$$

$$F_X(q) = bF_Y(q-1) + (1-b)(F_X(q-1)+1) \tag{6.2}$$

Here, $F_Y(0) = 0$ and $F_X(0) = 0$. Function $F_Y(q)$ reads as "the number of bag of words ENS is expected to correctly classify in the remaining q bags of the scene, given that ENS is in state Y". Function $F_X(q)$ can be read similarly, except that it computes the expected number of bag of words classified correctly if ENS is in state X.

Once a scene starts in Y, ENS correctly classifies the current bag of words with probability p, thus adding one to the summation and transitioning to state X with $q-1$ bag of words remaining in the scene (see first term of $F_Y(q)$). ENS misclassifies the current bag of word with probability $1-p$ and remains in state Y with $q-1$ bag of words remaining in the scene (see second term of $F_Y(q)$). Once in X, ENS correctly classifies the remaining bags of words if the classifiers do not agree on an incorrect emotion (probability b). Equations 6.1 and 6.2 assume p and b to be the same for all w.

The following lemma shows that $F_Y(q)$ can be written as a closed-form equation. $F_X(q)$ can also be written with a similar equation, but since our analysis assumes ENS starts in Y, we omit the closed-form equation of $F_X(q)$. The lemma can be proven by induction and the proof is in the Appendix.

Lemma 1 *$F_Y(q) = p(F_X(q-1)+1) + (1-p)F_Y(q-1)$ and $F_X(q) = bF_Y(q-1) + (1-b)(F_X(q-1)+1)$ can be written as follows,*

$$F_Y(q) = \frac{p \cdot \Big((1-p-b)^{q+1} + p + b - 1 + q \cdot (p+b)\Big)}{(p+b)^2}$$

$$F_X(q) = \frac{-b(1-p-b)^{q+1} + p^2q + pbq - pb - b^2 + b}{(p+b)^2}$$

By using the formulation shown in Lemma 1, we derive the minimum size q of a scene to guarantee that ENS is expected to be more accurate than a single classifier with accuracy k.

Theorem 1 *Let S be a scene of size $q \geq 0$ and c a classifier with accuracy $k \in (0, 1]$*

in S. Assuming that the probability values $p, b \in (0,1]$ are fixed for all bag of words in S, ENS is more accurate than c if $q > \frac{p^2-p+pb}{(p+b)^2k-p^2-pb}$ and $k < \frac{p(p+b)}{(p+b)^2}$.

The proof of Theorem 1 is in the Appendix. Theorem 1 states that if S is long enough and c is not too accurate, then ENS is expected to be more accurate than c in S. Note that a regular classifier is a special case of ENS with an ensemble of size one. In that case, $b = p - 1$, which according to Theorem 1, $q > 0$ as long as $k < p$, as one expects.

Our theoretical results suggest that ENS is able to reduce short misclassifications and can be more accurate than a single classifier. On the other hand, ENS might miss the emotion transitions of short scenes. This is because short scenes might finish before ENS transitions from state Y to state X.

Although ENS uses the restrictive unanimity rule, our analysis holds for other voting rules such as the majority rule. In that case, p and b mean the probability of the majority of the classifiers in the ensemble classifying a bag of words correctly or incorrectly, respectively. We chose to use the unanimity rule because this rule is expected to result in larger values of B and R, which can potentially reduce the sparsely distributed errors. Also, note that one could also analyze ENS by treating it as a Markov Chain with states X and Y whose transition matrix is defined by p, $p - 1$, q, and $q - 1$.

6.2 Empirical Evaluation

In this section we evaluate variants of ENS and Naive Bayes (NB) on the 9 episodes of CotW. NB classifies a bag of words w according to the probability of each word in w belonging to a class and according to the a priori probability of a sentence belonging to a class (Manning et al., 2008).

Two of the NB models we use are created by choosing different sliding window sizes. We use a leave-one-episode-out cross-validation procedure to select the two sizes. In the leave-one-episode-out cross-validation procedure we remove one episode from the set of training episodes and train the model on the remaining episodes. The model is then evaluated on the held-out episode. This process is repeated for all possible episodes in the training set. One NB model is obtained by selecting the sliding window size that yields the model with largest average accuracy in the cross-validation procedure; we call this model NS. The other model, called NM, is defined similarly, but by selecting the sliding window size that yields the model with

lowest average SM. We tested windows with size: $\{20, 25, 30, 35, 40\}$. The classifier NS is identical to the one used in Chapter 3

We create two extra NB models by using a feature selection scheme. A NB model with feature selection uses only the h words with largest mutual information (MI) value (Manning et al., 2008) in its classification procedure. The MI value of a word measures how discriminative the word is to identify or to rule out a given class. We then create NHS and NHM, which are similar to NS and NM, except that in the cross-validation procedure we choose the value of z and h that maximizes accuracy (for NHS) and minimizes SM (for NHM). We tested the following values of h: $\{500, 600, \cdots, 1500\}$, resulting in a total of 55 values tested for NHS and NHM.

We test two versions of ENS, one with NS and NHS composing its ensemble (ENS(2)) and one with NS, NHS, NM, and NHM composing its ensemble (ENS(4)).

Since the number of transitions is small in CotW and ENS tries to minimize SM by reducing the number of emotion transitions, a reasonable baseline is to assume Bardo only plays as background music a song related to the Suspenseful emotion, which is the majority class in our dataset. We call this approach Baseline in our table of results.

We separate each episode to be tested and train the algorithms on the other episodes. For example, when testing a method on episode 1, we train it with episodes 2–9 and the resulting model is applied to episode 1.

Note that ENS is model independent and one could use ENS with richer models such as LSTM (Hochreiter and Schmidhuber, 1997) to further improve the results we obtain with an ensemble composed of NB models.

6.3 Accuracy and SM Results

Table 6.1 shows the percentage accuracy and SM values of the tested algorithms ("Alg.") in each episode. The "Avg." column shows the algorithm's average results. All numbers are rounded to the closest integer and the values in the "Avg." column were computed before rounding the numbers. We highlight the background of a cell if the number in the cell represents the best result across all algorithms for a given episode.

We also highlight the best overall averages for each episode.

The ENS approaches present a much lower SM than all classifiers. Namely, ENS(4)'s SM is 7 times lower than NHM's, the best performing individual classifier. ENS(4) also has an average SM lower than Baseline and a much higher accuracy.

Alg.	Episodes									Avg.
	1	2	3	4	5	6	7	8	9	
Accuracy										
Baseline	10	53	35	44	75	64	29	24	47	42
NS	64	62	76	71	54	69	44	59	79	64
NHS	71	57	79	69	56	59	55	59	76	64
NM	64	62	80	72	52	60	43	61	78	64
NHM	68	57	79	69	47	65	56	61	76	64
ENS(2)	71	60	78	69	54	59	56	59	81	65
ENS(4)	76	59	79	70	50	59	55	64	80	65
SM										
Baseline	7	8	5	12	8	9	4	6	4	7
NS	42	29	43	40	37	41	59	35	36	40
NHS	45	28	23	33	35	22	50	42	50	36
NM	40	24	39	27	45	30	58	31	36	36
NHM	47	28	19	29	33	37	49	25	50	35
ENS(2)	14	4	4	10	9	0	25	10	13	9
ENS(4)	5	3	4	6	5	0	12	2	11	5

Table 6.1: Accuracy and SM for different classification algorithms.

As explained in Section 6.1, ENS with a larger set of classifiers is expected to change the emotion less often, potentially further reducing the value of SM. ENS(2) and ENS(4) have the same average classification accuracy, but ENS(4)'s SM is nearly half of the SM value of ENS(2).

ENS reduces the SM values throughout all episodes. The SM reduction can be observed in Figure 5.1, which shows the ENS(4) classifications in green in an excerpt of episode 6. ENS(4) has a SM of 3 in the excerpt shown in Figure 5.1: there are 8 true emotion transitions shown in red while ENS performs 5 transitions. The SM value of ENS(4) is zero if one considers the entire episode (Episode 6 in Table 6.1). As anticipated in our theoretical analysis, ENS's misclassifications are not sparsely distributed, they usually happen at the beginning of a scene. For example, there is a change from Calm to Suspenseful at around sentence 500 (see red lines) and ENS only detects such a transition after approximately 20 sentences. Also,

ENS misses entirely some of the short scenes in the episode (*e.g.,* the transitions in between sentences 400 and 500). The presence of short scenes justify the individual classifiers being more accurate than ENS in this episode (see Table 6.1), as presented in Theorem 1.

Chapter 7

User Study 2: A Short Misclassifications Evaluation

At this point we can see that ENS is able to change the distribution of error of its base classifiers. Instead of erring sparsely through an episode, ENS tends to concentrate its misclassifications in the beginning of the scenes. In this section we test our hypothesis that the sparsely distributed errors can harm the user experience with a detailed user study.

7.1 Empirical Methodology

In this second user study we compare NS with ENS(4) (henceforth referred as ENS). Following the same process of the first user study we selected five excerpts of the CotW. NS and ENS are trained with episodes different than the one from which the excerpt is extracted. Each excerpt is approximately 2 minutes long. In order to test our hypothesis, we selected excerpts in which NS is more accurate than ENS but it has a larger SM value. The accuracy and SM values of the video excerpts (V1, V2, $\cdots$, V5) are shown at the bottom of Table 7.2.

The video excerpts we use have no sentences of the Happy emotion again, thus we use one song for each of the other emotions in our study. The songs used were the same as in the first user study shown in Table 4.1. This time V1 is an excerpt starting at 3:22 and finishing at 5:20 of episode 2 of CotW; V2 starts at 25:10 and finishes at 26:32 of episode 6; V3 starts at 23:26 and finishes at 24:55 of episode 4; V4 starts at 17:26 and finishes at 18:56 of episode 3; V5 starts at 20:00 and finishes at 21:31 of episode 7. The detailed data are shown in Table 7.1.

Table 7.1: Each video in the experiment corresponds to an excerpt of the episodes that was randomly distributed, so we have the interval where they happen as well.

Excerpt	Episode	Starts	Finishes
1	2	03:22	5:20
2	6	25:10	26:32
3	4	23:26	24:55
4	3	17:26	18:56
5	7	20:00	21:31

Each participant listened to excerpts of all three songs after answering our consent form and before evaluating the excerpts. We reduce the chances of a participant evaluating the quality of the songs instead of the song selection procedure by telling them which songs will be used as background music. After listening to the songs each participant watched two versions of the same video excerpt, one with the background music selected by NS and another by ENS. The order in which the videos appeared was random to avoid ordering biases. We included a brief sentence providing context to the participant, to ensure they would understand to story being told in each excerpt. The participants could watch each video as many times as they wanted before answering the question: "Which video has the most appropriate background music according to the context of the story?". The participant could choose one of the options: "Video 1", "Video 2", "The background music used in both videos are equally appropriate", and "The background music used in both videos are equally inappropriate". After marking their answer, the participants would evaluate another pair of excerpts. The order in which the pairs of excerpts appeared was also random. The participants answered a demographic questionnaire after evaluating all excerpts.

Our experiment was advertised in D&D communities in the social media.[1] We had 41 participants, 40 males and 1 female, with average age of 25. All participants had some experience playing D&D. We removed the answers of 4 participants who reported to have only basic proficiency in English (the language used in the videos and in the study). We report the results of the remaining 37 participants, which resulted in 185 answers (5 pairs of videos for each participant).

Method	Video Excerpts				
	V1	V2	V3	V4	V5
ENS	59.4	59.4	35.1	37.8	48.6
NS	10.8	10.8	40.5	37.8	24.3
Tie+	13.5	10.8	10.8	18.9	16.2
Tie-	16.2	18.9	13.5	5.4	10.8
Accuracy					
ENS	87.8	0.0	32.3	48.3	32.3
NS	85.7	20.0	41.9	69.0	38.7
SM					
ENS	0	0	0	1	1
NS	6	2	10	3	9

Table 7.2: User preference in emotion detected by ENS and NS.

7.2 User Study Results

The videos with background music selected by ENS were preferred 89 times by the participants, while NS was preferred 46 times, and the approaches tied 50 times. The difference between ESN and NS is significant according to a two-sided binomial test ($p < 0.001$).

Table 7.2 shows the detailed results for all 5 excerpts used in our study. The upper part of the table shows the percentage of times the participants chose the videos edited by ENS, by NS, and the percentage of times the participants thought the videos to be equally appropriate (Tie+), and equally inappropriate (Tie-). For example, for the first two excerpts (V1 and V2), the participants preferred ESN's selection of background music in 59.40% of the cases and for the fifth excerpt (V5) the participants preferred the NS selections in 48.60% of the answers. The last two rows of the table show ENS and NS's accuracy in each excerpt. The highlighted cells show the best performing approach (ESN or NS) on a given excerpt.

7.3 Discussion

The results of our user study show a clear preference for the music selected by ENS. In particular, the participants strongly preferred the selection performed by ENS in V1, V2, and V5. V1 is an excerpt with two scenes in which both methods are accurate. While NS's misclassifications are sparsely distributed in V1, ENS's

[1]https://goo.gl/forms/uAu38mwVMGbyjprx1

misclassifications occur in the beginning of one of the scenes due to ENS's late transition. The participants had a strong preference by the ENS's V1.

ENS classified all sentences in V2 as Agitated while the sentences were Calm. NS correctly selected the Calm song for part of the excerpt but switched a few times between Calm and Agitated. In this case, the participants preferred the selection that was inaccurate and decisive than the selection that was more accurate but indecisive. ENS performs similar misclassification in V3, where it selects the Agitated song for a Suspenseful scene. In contrast with V2, ENS's misclassifications in V3 were not well perceived by the participants. V3 depicts a scene in which one of the players is sneaking in their enemy's house. The use of an Agitated song instead of a Suspenseful is more harmful to the user's experience than a large SM value in this particular case. V3 is the only excerpt in which NS outperforms ENS in terms of user preference.

V4 is a case in which ENS's selections are perceived as good as NS's. ENS is less accurate than NS but has a lower SM value. The accuracy of ENS and NS are similar in V5, but the latter has a large SM value. In this case the participants have a strong preference by ENS's selections.

The study supports our hypothesis that large SM values can be harmful to the user's experience and that it might be preferable to be inaccurate and decisive than accurate and indecisive. The results also show that, depending on the scene, the lack of accuracy can outweigh the system's decisiveness.

Chapter 8

Related Work

In addition to emotion recognition from speech signals reviewed in the introduction of this dissertation, Bardo relates to several lines of research in the machine learning, affective computing and human-computer interaction literature. In this section we review relevant works on generation of music scores for audio stories, emotion recognition from text, emotion-based music recommendation systems, multi-objective optimization applied to machine learning, and ensemble methods for machine learning.

8.1 Generation of Music Scores for Audio Stories

Researchers have worked on the problem of score composition to induce an emotion response (Monteith et al., 2010) and also the match the emotion of audio stories (Monteith et al., 2011; Rubin and Agrawala, 2014). Monteith et al. 2011 introduce generative models to compose scores of different emotions. Their system uses local information about the speech to generate novel scores from these models. Rubin and Agrawala 2014 extend this line research by accounting for the entire structure of the speech to compose scores using a dynamic programming approach.

While these works relate to ours in the sense that they generate music through speech, they also present important differences. First, in these works one assumes that the speech can be reliably transcribed into text. For example, Rubin and Agrawala 2014 assume that the audio stories for which their system will generate music for is provided in text format. By contrast, in Bardo relies on a speech recognition system that captures the speeches of several people simultaneously. Second, these works assume that the emotion of the sentences of the audio stories as well as the emotion of segments of the music to be provided as input. By contrast, Bardo

identifies through supervised learning the story's emotion in real-time. Third, these systems have access to the entire story before starting it composition process. By contrast, Bardo performs its background music selection in real time. Nevertheless, several of the ideas introduced in these works can be applied in the context of Bardo. As future work, in contrast with selecting a background music, we are interested in composing music in real time for tabletop games, similarly to the work of Rubin and Agrawala 2014.

8.2 Emotion Recognition from Text

Methods for emotion recognition from text use either categorial or dimensional approaches to model emotion. Categorical models use discrete labels to describe affective responses (Ekman, 1999; Dalgleish and Power, 2000) and dimensional ones attempt to model an affective phenomenon as a set of coordinates in a low-dimensional space (Posner et al., 2005). Although we use a categorial approach with Bardo, it will be interesting to investigate the use of dimensional models for the problem of background music selection.

Text-based emotion recognition systems are usually lexicon-based or machine learned-based, and they are often applied to problems such as computer assisted creativity (Davis and Mohammad, 2014), sentiment analysis (Pang and Lee, 2008) and text-to-speech generation (Alm, 2008). Davis and Mohammad 2014 used a lexicon-based approach to emotion recognition similar to the one we use with Bardo. Davis and Mohammad's approach was used to classify emotions in novels and later generate music for the novels. Strapparava and Mihalcea 2008 used a similar method to classify the emotions in newspaper headlines and a Naive Bayes classifier to detect Ekman's emotions (Ekman, 1999) in blog posts. Ma et al. 2005 presented another lexicon-based approach that recognizes the affective textual content of a chat system using a keyword spotting technique. This method was used to animate a 2D agent that performs the emotional coloring of the message using synthetic affective speech and appropriate gestures.

Balabantaray et al. 2012 presented a classifier that is able to determine the emotion of a person's writing; their approach is based on Support Vector Machines. Suttles and Ide 2013 used a distant supervision approach (Mintz et al., 2009) to classify emotions considering the eight bipolar emotions defined by Plutchik 1980. This allowed Suttles and Ide to treat the multi-class problem of emotion classification as a binary problem for opposing emotion pairs. Albornoz and Milone 2017 proposed

an ensemble classifier for emotion recognition in different languages using Ekman's model of emotion. Their method was trained in one set of languages and tested in another, being able to classify emotions with an accuracy of 58%. Danisman and Alpkocak 2008 proposed another emotion classifier, called Feeler, which uses a vector space model and was evaluated on news headlines. Feeler achieved a accuracy of 67.5%, outperforming other classifiers based on Support Vector Machine and Naive Bayes approaches.

Liu et al. 2003 created a robust lexicon-based method that classifies emotions (using Ekman's model) of everyday situations (e.g., getting into a car accident) by evaluating the affective nature of the underlying story semantics. Their method used the corpus of 400,000 facts about the everyday world. Another approach for emotion detection from text was proposed by Gill et al. 2008, where they explored the use of computational linguistics techniques to derive and detect linguistic components that are correlated with expressions of emotion in short blog texts.

Bardo contrasts with these works because of our application domain and its features. For example, while most of previous works in emotion recognition from text is interested in obtaining high classification accuracy, we showed empirically that one has also to optimize for short misclassifications for the problem of background music selection.

8.3 Emotion-Based Music Recommendation Systems

Bardo also relates to emotion-based music recommendation systems, which recommend music to match the physiological state of the user (Song et al., 2012). This is because Bardo can be seen as a system that automatically plays its music recommendation for a given tabletop game session. Cai et al. 2007 proposed an emotion-based music recommendation approach called MusicSense to automatically suggest music when users read Web documents such as Weblogs. Andjelkovic et al. 2016 proposed an interactive system called MoodPlay to model the user profile based on a set of artists selected by the user. Deng and Leung 2012 considered the user's historical playlist and employed Conditional Random Fields (Lafferty et al., 2001) to predict the user's emotional state.

Bardo differs from previous approaches to emotion-based music recommendation because it performs emotion-based music recommendation considering the users' speech as input.

8.4 Multi-Objective Optimization in Machine Learning

Park et al. 1999 used an evolutionary multi-objective approach in a systems control problem. Similarly to our problem, their controller minimized two objective functions. In addition to practical applications, multi-objective optimization have been used to enhance machine learning models through feature selection (Emmanouilidis et al., 1999), regularization (de A. Teixeira et al., 2000), and generation of a diversified set of classifiers for an ensemble approach (Chandra and Yao, 2004). Multi-objective approaches have been also used to improve understandability of rule extraction systems (Ishibuchi et al., 1997). For a review on multi-objective optimization in machine learning, see the work by Jin and Sendhoff 2008.

Bardo differs from previous works as it is a human-centered approach, and as such, we test Bardo's multi-objective approach with a detailed user study.

8.5 Error Distribution of Autonomous Systems

The relation between system accuracy and user experience in terms of trust has been extensively studied, see the work of (Yang et al., 2017) (Yang et al., 2017) for a recent example. Most of the works on accuracy and trust involve one manipulating the error distribution of an autonomous system and measuring the user's trust on the system. Sanchez Sanchez (2006) controlled system errors to occur either in the first or the second half of a simulated task. They found that the users' trust was significantly lower if the errors occurred on the second half of the simulation. Desai *et al.* Desai et al. (2012) observed that users tend to switch an autonomous system to manual mode more often if the system errors occur in the middle of a task. In addition to studying the impact of system errors in user experience, we introduce a supervised learning model that alters the prediction error distribution of a music selection system. Also, we measure the impact of the error distribution on how the users perceive the background music, and not the user's trust on the system.

8.6 Ensemble Approaches to Supervised Learning

The combination of an ensemble of diverse classifiers that perform slightly better than random guessing can result in very accurate models (Hansen and Salamon, 1990; Schapire, 1990). Several algorithms were developed to create such ensembles. For example, in Bagging, one trains several classifiers with a different sampling of the training data (Breiman, 1996). AdaBoost also manipulates the training data by applying a weight to the training error of each training instance (Freund and Schapire, 1997). Another way to create a set of potentially diverse classifiers is by training models on different subsets of the instances' features (Cherkauer, 1996). We use an approach similar to Cherkauer's as we train a set of classifiers with and without a feature selection procedure to compose ENS's ensemble. Nonetheless, one could use any of the previous approaches to train a set of diverse classifiers to compose ENS's ensemble.

In contrast with other ensemble methods which primarily try to improve prediction accuracy, ENS is designed to alter the error distribution of its base classifiers. Also, our analysis differs from previous ones as we verify how long a given game "scene" has to be so that ENS is expected to be more accurate than a single classifier.

Chapter 9

Conclusion

We discussed Bardo, a real-time intelligent system to automatically select the background music for tabletop RPG games. Bardo was evaluated with a real-world online campaign of D&D. We evaluated the accuracy of two classifiers for our emotion classification task. Our results showed that a simple Naive Bayes variant is able to obtain good classification accuracy. Second, we conducted a user study in which people evaluated the background music selections performed by Bardo in the D&D campaign. We compared Bardo's selections with those performed by humans, and our results showed that the participants had a clear preference for the selections made by Bardo. Improvements for the first results were made by doing a feature selection where we have gotten a more balanced results and increased the accuracy. These results give a new way to provide a better game experience for the users by improving the distribution of error in a game session. As the main contribution of this work we showed that a system whose errors are sparsely distributed across a game session can harm user experience. We introduced an ensemble approach called ENS that errs consistently in the beginning of scenes as opposed to sparsely through the scenes. Theoretical results showed that ENS can reduce the sparsely distributed errors by performing fewer music transitions. We also showed that if a scene s is long enough and a classifier c is not too accurate, then ENS is expected to be more accurate than c in s. Empirical results showed that ENS is able to reduce the sparsely distributed errors without sacrificing accuracy. A user study in which people watched videos of D&D with the background music selected by ENS and by a classifier that does not account for the sparsely distributed errors supported our hypothesis. Another contribution of our work is a labeled dataset of sentences captured from almost 5 hours of D&D gameplay. Our dataset will be made available to other researchers interested in the problem of background music selection for

tabletop games. In the future we intend to evaluate Bardo in live sessions of D&D.

Bibliography

Albornoz, E. M. and Milone, D. H. (2017). Emotion recognition in never-seen languages using a novel ensemble method with emotion profiles. *IEEE Transactions on Affective Computing*, 8(1):43--53.

Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579--586. Association for Computational Linguistics.

Alm, E. C. O. (2008). *Affect in text and speech*. University of Illinois at Urbana-Champaign.

Andjelkovic, I., Parra, D., and O'Donovan, J. (2016). Moodplay: Interactive mood-based music discovery and recommendation. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 275--279. ACM.

Balabantaray, R. C., Mohammad, M., and Sharma, N. (2012). Multi-class twitter emotion classification: A new approach. *International Journal of Applied Information Systems*, 4(1):48--53.

Bergström, K. and Björk, S. (2014). The case for computer-augmented games. *Transactions of the Digital Games Research Association*, 1(3).

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123--140.

Cai, R., Zhang, C., Wang, C., Zhang, L., and Ma, W.-Y. (2007). Musicsense: contextual music recommendation using emotional allocation modeling. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 553--556. ACM.

Chandra, A. and Yao, X. (2004). Divace: Diverse and accurate ensemble learning algorithm. In *Intelligent Data Engineering and Automated Learning*, pages 619--625.

Cherkauer, K. (1996). Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. In *Working Notes of the AAAI Workshop on Integrating Multiple Learned Models*, pages 15--21.

Dalgleish, T. and Power, M. (2000). *Handbook of cognition and emotion*. John Wiley & Sons.

Danisman, T. and Alpkocak, A. (2008). Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, page 53.

Davis, H. and Mohammad, S. M. (2014). Generating music from literature. *arXiv preprint arXiv:1403.2124*.

de A. Teixeira, R., Braga, A., Takahashi, R., and Saldanha, R. (2000). Improving generalization of mlp with multi-objective optimization. *Neurocomputing*, 35:189--194.

Deng, J. J. and Leung, C. (2012). Emotion-based music recommendation using audio features and user playlist. In *Information Science and Service Science and Data Mining (ISSDM), 2012 6th International Conference on New Trends in*, pages 796--801. IEEE.

Desai, M., Medvedev, M., Vázquez, M., McSheehy, S., Gadea-Omelchenko, S., Bruggeman, C., Steinfeld, A., and Yanco, H. (2012). Effects of changing reliability on trust of robot systems. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 73--80, New York, NY, USA. ACM.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1--15, London, UK, UK. Springer-Verlag.

Drummond, C. and Holte, R. C. (2006). Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65(1):95--130.

Ekman, P. (1999). Basic emotions. In Dalgleish, T. and Power, M. J., editors, *The Handbook of Cognition and Emotion*, pages 45–60. John Wiley & Sons, Sussex, U.K.

Emmanouilidis, C., Hunter, A., MacIntyre, J., and Cox, C. (1999). Selecting features in neurofuzzy modelling by multiobjective genetic algorithms. In *International Conference on Artificial Neural Networks*, page 749?754.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119--139.

Gill, A. J., French, R. M., Gergle, D., and Oberlander, J. (2008). Identifying emotional characteristics from short blog texts. In *Proc. 30th Ann. Conf. Cognitive Science Soc., BC Love, K. McRae, and VM Sloutsky, eds*, pages 2237--2242.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 12(10):993--1001.

Harrenstien, K. (2009). Automatic captions in youtube. https://googleblog.blogspot.com.br/2009/11/automatic-captions-in-youtube.html.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735--1780.

Ishibuchi, H., Murata, T., and Turksen, I. B. (1997). Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems*, 89(2):135--150.

Jin, Y. and Sendhoff, B. (2008). Pareto-based multiobjective machine learning: An overview and case studies. *Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(3):397--415.

Lafferty, J., McCallum, A., Pereira, F., et al. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282--289.

Liu, H., Lieberman, H., and Selker, T. (2003). A model of textual affect sensing using real-world knowledge. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 125--132. ACM.

Ma, C., Osherenko, A., Prendinger, H., and Ishizuka, M. (2005). A chat system based on emotion estimation from text and embodied conversational messengers.

In *Active Media Technology, 2005.(AMT 2005). Proceedings of the 2005 International Conference on*, pages 546--548. IEEE.

Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41--48. AAAI Press.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003--1011. Association for Computational Linguistics.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436--465.

Monteith, K., Francisco, V., Martinez, T., Gervás, P., and Ventura, D. (2011). Automatic generation of emotionally-targeted soundtracks. In *Proceedings of the International Conference on Computational Creativity*.

Monteith, K., Martinez, T., and Ventura, D. (2010). Automatic generation of music for inducing emotive response. In *Proceedings of the International Conference on Computational Creativity*, pages 140--149. Department of Informatics Engineering, University of Coimbra, Department of Informatics Engineering, University of Coimbra.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1--135.

Park, S., Nam, D., and Park, C. H. (1999). Design of a neural controller using multiobjective optimization for nonminimum phase systems. In *IEEE International Fuzzy Systems Conference*, page 533?537.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. *Theories of emotion*, 1(3-31):4.

Posner, J., Russell, J. A., and Peterson, B. S. (2005). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17(03):715--734.

Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, 42(3):203--231.

Rubin, S. and Agrawala, M. (2014). Generating emotionally relevant musical scores for audio stories. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, pages 439--448. ACM.

Sanchez, J. (2006). *Factors that affect trust and reliance on an automated aid.* PhD thesis, Georgia Institute of Technology.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197--227.

Song, Y., Dixon, S., and Pearce, M. (2012). A survey of music recommendation systems and future perspectives. In *9th International Symposium on Computer Music Modeling and Retrieval*.

Strapparava, C. and Mihalcea, R. (2008). Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556--1560. ACM.

Suttles, J. and Ide, N. (2013). *Distant Supervision for Emotion Classification with Discrete Binary Values*, pages 121--136. Springer Berlin Heidelberg, Berlin, Heidelberg.

Xu, L., Krzyzak, A., and Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418--435.

Yang, X. J., Unhelkar, V. V., Li, K., and Shah, J. A. (2017). Evaluating effects of user experience and system transparency on trust in automation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 408--416, New York, NY, USA. ACM.

Appendix A

Appendix: Proofs

Proposition 1 *For a given set of classifiers $C = \{c_1, c_2, \cdots, c_n\}$ and a subset of size one $C' = \{c_k\}$ with $c_k \in C$, we have that $B(C, e_j) \geq B(C', e_j)$ and $R(C, e_j) \geq R(C', e_j)$.*

Proof. We start by showing that $B(C, e_j) \geq B(C', e_j)$,

$$\begin{aligned} B(C, e_j) &= \left(\prod_{c \in C} p_c(e_j|e_j) \right)^{-1} \\ &\geq \left(p_{c_k}(e_j|e_j) \right)^{-1} = B(C', e_j)\,. \end{aligned}$$

The inequality is because $p_c(e_j|e_j) \leq 1$ for any $c \in C$. Thus $\prod_{c \in C} p_c(e_j|e_j) \leq p_{c_k}(e_j|e_j)$. The equalities are due to the definition of B. As for $R(C, e_j) \geq R(C', e_j)$,

$$\begin{aligned} R(C, e_j) &= \left(\sum_{\substack{e \in E \\ e \neq e_j}} \prod_{c \in C} p_c(e|e_j) \right)^{-1} \\ &\geq \left(\sum_{\substack{e \in E \\ e \neq e_j}} p_{c_k}(e|e_j) \right)^{-1} = R(C', e_j)\,. \end{aligned}$$

Similarly to the B case, the inequality is because $p_c(e_j|e_j) \leq 1$ for any $c \in C$, thus $\sum_{\substack{e \in E \\ e \neq e_j}} \prod_{c \in C} p_c(e|e_j) \leq \sum_{\substack{e \in E \\ e \neq e_j}} p_{c_k}(e|e_j)$. The equalities are due to the definition of R. □

Lemma 1 *$F_Y(q) = p(F_X(q-1)+1) + (1-p)F_Y(q-1)$ and $F_X(q) = bF_Y(q-1) +$*

$(1-b)(F_X(q-1)+1)$ *can be written as follows,*

$$F_Y(q) = \frac{p\Big((1-p-b)^{q+1}+p+b-1+q(p+b)\Big)}{(p+b)^2}$$

$$F_X(q) = \frac{-b(1-p-b)^{q+1}+p^2q+pbq-pb-b^2+b}{(p+b)^2}$$

Proof. Our proof is by induction. Replacing $q=0$ in the equations above we obtain $F_Y(0) = p(1-p-b+p+b-1) = 0$ and $F_X(0) = -b+bp+b^2-pb-b^2+b = 0$.

We assume as inductive hypothesis that $F_Y(q-1) = p\Big((1-p-b)^q+p+b-1+(q-1)(p+b)\Big)(p+b)^{-2}$ and $F_X(q-1) = \Big(-b(1-p-b)^q+p^2(q-1)+pb(q-1)-pb-b^2+b\Big)(p+b)^{-2}$.

By replacing $F_Y(q-1)$ and $F_X(q-1)$ according to the inductive hypothesis in the recursive equation of $F_y(q)$ we obtain the following,

$$\begin{aligned}
F_Y(q) &= \Bigg(p\Big(-b(1-p-b)^q+p^2(q-1)+pb(q-1)-pb-b^2+b+(a+b)^2 \\
&\quad +(1-p)(1-p-b)^q+p-p^2+b-pb-1+p+(q-1-pq+p)(p+b)\Big)\Bigg)(p+b)^{-2} \\
&= \frac{p\Big((1-p-b)^{q+1}+p+b-1+q(p+b)\Big)}{(p+b)^2}
\end{aligned}$$

Similarly, we replace $F_Y(q-1)$ and $F_X(q-1)$ according to the inductive hypothesis in the recursive equation of $F_X(q)$ to obtain the following,

$$\begin{aligned}
F_X(q) &= \Bigg(pb\Big((1-p-b)^q+p+b-1+(q-1)(p+b)\Big) \\
&\quad +(1-b)\Big(-b(1-p-b)^q+p^2(q-1)+pb(q-1)-pb-b^2+b+(p+b)^2\Big)\Bigg)(p+b)^{-2} \\
&= \Bigg(pb(1-p-b)^q-pb+p^2bq+pb^2q \\
&\quad +(1-b)\Big(-b(1-p-b)^q+p^2q+pbq+b\Big)\Bigg)(p+b)^{-2} \\
&= \frac{-b(1-p-b)^{q+1}+p^2q+pbq-pb-b^2+b}{(p+b)^2}
\end{aligned}$$

□

Theorem 1 *Let S be a scene of size $q \geq 0$ and c a classifier with accuracy $k \in (0,1]$ in S. Assuming that the probability values $p, b \in (0,1]$ are fixed for all bag of words in S, ENS is more accurate than c if $q > \frac{p^2-p+pb}{(p+b)^2k-p^2-pb}$ and $k < \frac{p(p+b)}{(p+b)^2}$.*

Proof. ENS is expected to be more accurate than C if

$$\frac{p\Big((1-p-b)^{q+1}+p+b-1+q(p+b)\Big)}{(p+b)^2} > kq$$
$$\frac{(p+b)^2kq}{p} - p - b + 1 - q(p+b) < (1-p-b)^{q+1}$$

Since $(1-p-b)^{q+1} \geq 0$, the equation above holds if

$$\frac{(p+b)^2kq}{p} - p - b + 1 - q(p+b) < 0 \tag{A.1}$$
$$q\Big((p+b)^2k - p^2 - pb\Big) < p^2 - p + pb \tag{A.2}$$
$$q > \frac{p^2-p+pb}{(p+b)^2k-p^2-pb} \tag{A.3}$$

In Equation A.3, p^2-p+pb is negative as one needs $b+p>1$ for it to be positive, and $b+p \leq 1$. Thus, Equation A.3 holds if $(p+b)^2k-p^2-pb<0$, or $k < \frac{p(p+b)}{(p+b)^2}$. Suppose $(p+b)^2k-p^2-pb>0$, since $p^2-p+pb<0$, then one needs $q<0$ for Equation A.2 to hold, but $q \geq 0$. □