UNIVERSITY OF TORINO

M.Sc. in Stochastics and Data Science

Final dissertation

Computer-Assisted Evaluation of Story-Driven Interactive Storytelling Systems

Supervisor: Roberto Esposito
Co-supervisor: Vincenzo Lombardo

Candidate: Matteo Silvestro

ACADEMIC YEAR 2016/2017

Summary

The evaluation of the narrative emerging from storytelling systems is not a trivial task, not even for linear ones. For interactive stories, the space of possible developments may grow very quickly and become easily unmanageable. The story designer has hardly any assistance at all in this process. We propose, expanding on previous attempts, a general methodology to evaluate story-driven interactive storytelling systems via clustering, tension curve extraction and user surveys. This procedure outputs a set of clusters, each with its own specific tension curve shape and average quality score. The story designer may inspect the resulting clustering and iterate over his/her storytelling system using the new knowledge acquired. This may also lead to an association between tension curves and quality of a story. We apply this methodology to our story-driven interactive storytelling system to show its current and potential achievements, respectively. Our results indicate that clusters, even if not well-formed, display different quality scores and that some tension curves seems to be associated with better stories. While the method has proven to be valid, there is room for improvements.

Acknowledgements

I would first like to thank my supervisor Dr. Roberto Esposito and my co-supervisor Prof. Vincenzo Lombardo, who supported me throughout my thesis with their patience and knowledge.

I would also like to mention all professors and researchers of the University of Torino that attended the seminar for the presentation of the topics debated in my thesis. Their suggestions were very useful.

Prof. Antonio Pizzo, Prof. Vincenzo Lombardo, and the students of the course *Interactive storytelling* made possible the interactive storytelling system DoppioGioco and the interactive story *Hot Bread*, providing an interesting subject to analyze in my thesis. Moreover, Prof. Antonio Pizzo's knowledge of dramatic structures was incredibly helpful, helping to model the methodology in a meaningful way.

An invaluable assistance in the creation of the user survey was given by Prof. Rossana Damiano, that I would like to thank for all her contributions and the promptness of her replies. I am also indebted to all people that took their time to complete the survey, without data these results would not have been possible.

Finally, I would like to express my profound gratitude to Constanza Amanda Pelissero for her helpful corrections and unfailing support, and to my family, that was always there when I needed anything.

Contents

Chapter 1

Introduction

Stories drive our lives in many different ways. We tell stories to entertain ourselves, to explain things in a more enjoyable way, to pass knowledge through generations, to overcome traumas and make us feel better. People are fundamentally storytellers, they tell narratives about their experiences and what such experiences meant to them [9, 10, 13].

At the same time, people love to hear good stories. A trivial list of events may be very boring, but if we wrap all those events in an engaging narrative everything changes and becomes entertaining, if not fascinating. Being able to tell a good story is a crucial matter in a lot of entertainment forms, be it a book, movie or video game. Not only, even advertisements are more effective if there is a good narrative behind [49].

That being said, it is easy to understand how important is to have some tools to evaluate the quality of stories. We want the majority of people to enjoy them and be engaged by them. Extensive studies have been made on linear stories, such as those you find on books or movies. Quite often the simple yet effective approach of the Freytag's pyramid [18] fits very well for successful stories [37]. However, new types of narratives are arising, demanding new approaches. One of them, namely interactive stories, can become quite difficult to manage and, hence, storytellers willing to explore them need tools to confront their quest. We take the first steps into the mostly unexplored field of the evaluation of interactive storytelling systems (from now on IS systems).

1.1 Interactive stories

Interactive storytelling is a form of digital entertainment in which the story plot is not predetermined. The author creates the world, the characters and some possible events, but the user interacts with the story and changes its development, living each time a different experience.

Developing an IS system is a challenging task involving multi-disciplinary knowledge. Since IS systems require artificial intelligence, at least to some extent, and a device with which to interact, usually a computer or a mobile phone, they are a fairly new type of entertainment. Nonetheless, the first examples of interactive stories can be found in the gamebook era.

A *gamebook* is a work of printed fiction that allows the reader to participate in the story making choices. The first gamebook in the modern sense was published around 1945 by Alan George (probably a pseudonym) under the name of *Treasure Hunt*. The reader must help two siblings to find a castle said to contain a hidden treasure. The book is composed of 24 numbered units (blocks of text), with a third-person narrative on the left page and an illustration of the current location on the right. Each illustration contains indications of possible directions of travel, identified by a unit number to go to [24].

However, they did not become popular until the children gamebooks series *Choose Your Own Adventure* was published, starting from 1979 until 1998. These stories are written from a second-person point of view[1] and involve the protagonist - that is the reader - taking on a role that is relevant to the adventure, like a mountain climber, a doctor or a deep sea explorer. After reading a page, the reader is asked to take a decision, that will make him turn to another page and so on, until one of the many possible endings is reached [22].

Later on, the influence of role-playing games such as the widely popular *Dungeons & Dragons* led to more complex gameplay mechanics. In well-known series such as *Fighting Fantasy* or *Lone Wolf* player statistics and dice-based role-playing elements were added. Not surprisingly, more recently the series *Lone Wolf* was converted into computer games, such as *Joe Dever's Lone Wolf HD Remastered* (2014).

In fact, the diffusion of personal computers gave rise the transition of

[1]The second-person point of view is a form of storytelling in which a narrator relates all the action of their work using a second-person pronoun such as "you." For instance: "You are about to begin reading Italo Calvino's new novel, *If on a winter's night a traveler*." [7].

interactive stories from books, that were anyway limited, to computers, with endless possibilities. One of the first efforts in this direction are text adventure games. One of the earliest examples is *Zork* (1980), in which you are a treasure-hunter venturing into a dangerous land in search of wealth and adventure [29]. The player interacts with the world by typing verbs in a command prompt-like interface.

With increasing graphics processing abilities, more elaborated IS systems were possible. One example is that of *open world role-playing games*, such as *Fallout: A Post Nuclear Role Playing Game* (1997) and its sequels, in which each decision made by the player strongly influences the world and the possible endings. In Japan, *visual novels* are widely popular. They feature mostly static graphics, but the interaction with various characters leads towards different endings. Visual novels encourage the player to start the story again over and over to get all possible endings, eventually unlocking a final *true ending*. Some notable examples are *Clannad* (2004) and *Nine Hours, Nine Persons, Nine Doors* (2009). Finally, independent video game developer *Telltale Games* brought to life a whole new genre, taking its roots from adventure games. Their first critically acclaimed title, *The Walking Dead* (2012), is an episodic interactive drama graphic adventure game taking places in the wold of *The Walking Dead* comics. The player is forced to take heart-breaking decisions that may decide upon the life of a person or the survival of the entire group, interacting mainly through quick time events.

Recently, tools to create interactive stories in an easier way, even for non-programmer story designers, have begun to spread. One of them, *Twine* (2009), makes it possible to create interactive stories using a visual interface, with no coding required. The resulting story is published in HTML format, easy to share almost everywhere. The site itself has a collection of works published using this tool.

So, IS systems have been becoming increasingly popular and sophisticated. Nowadays, it is possible to integrate a strong narrative (storytelling skills) with highly-complex interaction systems (artificial intelligence skills).

To be clear about the terminology we use from now on,

- *interactive storytelling system* refers to the objects that make possible to experience an interactive story, for example a book (in the case of gamebooks) or a computer software (in the case of interactive video games) that is realized in such a way that interactions with the story world are possible;

- *interactive story* refers to a story that can take different paths, according to the interaction of the user with the story world, that is experienced through an IS system;
- *linear story* refers to a story that is static and hence stays the same every time the user experiences it.

It is also important to note that if the interaction of the user with an interactive story (e.g. the set of choices) is fixed, then we obtain a linear story and we say that we extracted a linear story from the interactive story. In fact, every interactive story can be converted to the set of all possible linear stories.

1.2 Classification of interactive stories

IS systems can be divided into two main types [30]:

- *character-driven*, if the story emerges from interaction between characters;
- *story-driven*, if the story is based on narrative units connected one to another.

In character-driven interactive stories, the story designer must create the story world and a set of characters. The interaction of characters generates the story, that is every time different. Most of the times the user impersonates one of the characters, while the others are controlled by an artificial intelligence agent. This requires a very well-tuned artificial intelligence, since this is what drives the story. One big advantage of this approach is that the possibilities are infinite, since a lot of different events may be generated by interaction between different characters at different times and in different situations. On the other side, the story designer has a limited control over the resulting story. In fact, while the story designer decides how the story world and the characters behave, their interaction can trigger unexpected events and it is almost impossible to predict them all. Moreover, most of them may not even be interesting or engaging. In this category fall open world role-playing games, for instance.

In story-driven interactive stories the story designer creates a *story graph*, that is composed of two parts:

- *units*, that are static texts or pre-scripted sequences of events;

- *connections*, that manages how units are connected one to another.

Hence, the units are the building blocks of the whole story. They are entirely predetermined and written by the story designer itself. This means that they are interesting and engaging, provided the story designer has good storytelling skills. What discriminates them from linear stories is the way these units are connected between each other. In fact, for each unit the user is faced with a choice that will, in turn, lead to another unit, until a final unit (an ending) is finally reached. Being composed of human-made units and hopefully well designed connections, the resulting interactive story will be of higher average quality with respect to character-driven ones. However, the interaction is much more limited and the possibilities are fewer. This may not be a problem, because it means having better control over the story. In fact, it is not uncommon to limit the story branches so that they converge to a limited number of endings at a certain point. This is particularly common in commercial products, for instance *Clannad* (2004) is limited to fourteen endings. Other methods may be used, like giving the user the illusion to have choices, while making the narrative converge anyway to a fixed point, a sort of bottleneck. Some good examples are *The Walking Dead* (2012) and *The Walking Dead: Season Two* (2013–2014), that have respectively one and five endings. Other examples that fall in this category are gamebooks and visual novels.

Summarizing, character-driven interactive stories reward interactivity over narrative, while story-driven ones reward narrative over interactivity.

1.3 Evaluation of interactive stories

The main issue about the evaluation of interactive stories resides in managing all possible experiences the user may have depending on the choices made. The set of all possible linear stories extracted from an interactive story is called *story space*. Specifically,

- in character-driven IS systems, the story space is literally infinite. In fact, the interaction between characters may give rise to countless events, possibly unexpected. One way to tackle this problem is to resort to a finite number of play sessions of the interactive story, either by human or random agents [6].
- in story-driven IS systems, the story space is limited but usually has a combinatorial explosion that can become quickly impossible to manage.

In fact, even a seemingly harmless amount of 107 units can give rise to an astonishing number of 55,888 linear stories (as it is in our case). One way to work around this issue is a random sampling of the story space.

We have now a smaller and hopefully more manageable story space in both cases. At this point, we can apply well-known evaluation methods to the linear stories in the reduced story space. However, we must also take into account that all these stories were extracted from a unique interactive story and hence have some connections one to another. A simple yet effective way to further reduce the story space and give meaning to it is to divide linear stories in groups, such that all members of each group are somehow similar and share some features, like the genre or the emotions generated in the user. Story designers can then have a better overall look at their IS system, possibly even hints at what parts require enhancements or modifications.

1.4 Our goal

Story designers with a fully functional IS system can benefit of tools to help them evaluate and hopefully improve their work. We build our researches upon previous efforts in this direction. In the article *Towards Automatic Story Clustering for Interactive Narrative Authoring* [6], Bída, Černý and Brom propose a methodology to evaluate a generic character-driven IS system.

In fact, they point out that the state of the art evaluation of IS systems involves either user surveys, e.g. [42], or technical evaluations. The former has the advantage of actually capturing the user perceived quality of the story, if well designed, and this is exactly our aim. In fact, we want the user to enjoy our stories. However, they are costly and take a lot of time. The latter has the advantage of being inexpensive and fast. Technical evaluations may be based on story properties like length and complexity. However, they usually have little to do with the actual enjoyment of the story by the user.

They proposed a computer-assisted evaluation methodology of stories generated by a character-driven IS system. This involves the separation of stories extracted from their own IS system into well-formed groups, so that the story designer needs to inspect directly only a few stories per group.

We build on their results. We adapt their methodology to story-driven IS systems and extend it, so that it also outputs an average quality score per

group and possibly find a connection between stories traits and engagement by the user. Furthermore, we show its application to an IS system.

1.5 Methodology outline

We propose a general methodology to evaluate an IS system and obtain some insights about the quality of the resulting narrative, pointing to possible areas of improvement.

In Bída et al. [6], the target IS system is character-driven and the methodology works as follows:

1. the story designer generates a large number of linear stories extracted from the IS system;
2. the story designer runs the clustering algorithm dividing the stories into groups;
3. the story designer now needs to check only several stories from each group to evaluate the IS system.

The main goal is to let the story designer reduce the story space in a meaningful way, so that each group contains stories with similar features and only a few of them must be inspected directly. The others are assumed to behave similarly to the ones checked. This way, the story designer can have a decent overview of the whole possible narratives experienced by users of the interactive story.

We improve on this methodology, tailoring it for story-driven IS systems. This works as follows:

1. using a suitable explorer, extract all linear stories (or a significant sample of them, if not possible) from the story graph;
2. associate each linear story to a tension curve;
3. run a clustering algorithm on the linear stories;
4. design a user survey, so that every linear story has a human score associated that expresses the quality of the story;
5. use the quality scores to evaluate the clustering;
6. the story designer can inspect the clusters to have insights into the IS system.

While we address the details later, we do not just want the story designer to reduce the story space. In fact, we also want to output groups of stories with, hopefully, their own specific tension curve, that is a sort of dramatic trend, and quality score.

We first review research in the field of IS systems and describe our IS system of choice. Then, we explain what is clustering and what algorithms we use, paying particular attention to the choice of distances. Finally, we talk about the design of the user survey and how to get information from the output of this methodology. We conclude with a discussion and possible future developments.

Chapter 2

Interactive stories state of the art

We present an overview of the actual state of IS systems from an academic point of view and then review related work on evaluation of IS systems. Finally, we also introduce our own IS system, DoppioGioco.

2.1 Milestones of interactive storytelling systems

We review some of the most important IS systems emerging from past researches. This list is by no means exhaustive, but it helps to understand the recent achievements in this field. Furthermore, each IS system chosen has its own peculiarity:

- Façade is one of the most complex and complete IS systems published;
- PaSSAGE is a very interesting approach in making video games stories more interactive;
- FearNot! is a pedagogical IS system aimed at anti-bullying education;
- Madame Bovary is an immersive IS system.

2.1.1 Façade

A reference point for most research done in the field of interactive storytelling is Façade [34], an artificial intelligence-based art/research experiment in electronic narrative. In Façade the player plays the character of a longtime friend of Grace and Trip, an attractive and materially successful couple. During an evening get-together at their apartment that quickly

Figure 2.1: Façade, when you meet Trip and Grace.

becomes ugly, you get entangled into their marriage issues. Your decisions can change the course of Grace and Trip's lives. It is one of the few IS systems that are publicly released, free to be played[1]. The authors made a novel architecture that integrates emotional, interactive character behavior, drama-managed plot and shallow natural language processing. The player interacts by moving in the apartment, clicking on objects or persons and writing text at any moment, appearing at the bottom like subtitles.

Façade tries to find a middle ground between character-driven and story-driven IS systems. In fact, on a moment-by-moment basis there is a simulated world, with autonomous behaviour-based characters and the player itself. This is what defines a character-driven IS system and makes possible a high degree of freedom. However, there is also an additional invisible agent called *drama manager*. The drama manager continuously monitors the situation and proactively modifies the behaviors of Grace and Trip. These updates are organized into *story beats*, each a collection of behaviors tailored to a particular situation or context but still offering a non-trivial

[1] http://www.interactivestory.net/

simulation space. Beats are annotated by the author with preconditions and effects on the story state, instructing the drama manager when they make sense to use, in the interest of creating an overall dramatic narrative (a plot). These preconditions and effects serve to specify a partial ordering of beat sequences. By choosing beat sequences with appropriate tension values, the resulting narrative follows the Freytag pyramid. This way, it is possible to have a coherent and engaging plot, according to the intention of the story designer, while allowing for high freedom and continuous interaction.

2.1.2 PaSSAGE

PaSSAGE [45] (Player-Specific Stories via Automatically Generated Events) is an artificial intelligence system designed to dynamically select elements of content in a video game based on an automatically learned model of its current player's preferences. PaSSAGE operates based on a set of designer-supplied preference annotations, that are composed of available player actions and potential story events. They use the player types (from Robin Laws' rules [28]) as the basis for their model: Fighters (who prefer combat), Power Gamers (who prefer gaining special items and riches), Tacticians (who prefer thinking creatively), Storytellers (who prefer complex plots) and Method Actors (who prefer to take dramatic actions). Annotations on player actions are encoded by how strongly they are a representation of each of this five play styles. Whenever the player executes an action, the model of the player (consisting of a vector of weights for each play style) is updated. The higher the weight, the stronger the model's belief that the player prefers that style of play. Annotations on story events encode the suitability of that event for each play style. That way, the next event in the story is chosen by PaSSAGE calculating the overall suitability of each available event, taking into account both the event annotations and the current model of the player. For instance, if a player is mostly a Fighter he/she is more likely to be ambushed by some enemies.

PaSSAGE is primarily a character-driven IS system. In fact, it is well-suited for role-playing video games by construction: not surprisingly, the authors demonstrated their IS system in a custom module of *Neverwinter Nights* (2002). Anyway, it carries some elements from story-driven IS systems, since it has annotated events. Both the events and annotations are crafted by the story designer itself.

In the article the authors present the results of a human user study designed to test PaSSAGE's ability to create interactive stories against two narratives with predetermined structures and their IS system produced stories that were more fun and that afforded better agency.

2.1.3 FearNot!

FearNot! (Fun with Empathic Agents Reaching Novel outcomes in Teaching) is an IS system designed to allow children (from now on players) to explore what happens in bullying in an nonthreatening environment in which they took responsibility for what happened to a victim, without themselves feeling victimized [3]. The creation of an empathic relationship between player and character was seen as the mechanism through which this sense of responsibility would be achieved, so that the child player would really care what happened to the victimized character. The player is asked to act as an invisible friend, and to give advice which would influence the behavior of the victim without undermining its autonomy of action and the player's ability to believe in it as a character with an independent inner life. The story starts with introducing the school and the characters, then a bullying episode occurs. The victim then asks the player for advice in dealing with this and the player suggests a coping behavior. This structure is repeated twice.

FearNot! is, again, mostly a character-driven IS system. In fact, the authors previously made a scripted version, that was not very responsive to player interactions – it did not seem realistic enough. Then, they implemented each character as an autonomous artificial intelligence agent, with its own emotions and goals following the Ortony, Clore and Collins (OCC) theory of emotions (emergent version) [35]. This choice did not however removed the need for a drama manager, that is a typical tool of story-driven IS systems. In fact, the choice about where the new episode must be set or where characters are present must be made, as well as deciding other initial conditions. Moreover, some scenes may occur off-screen, as if the player suggests the victim to tell a parent or a teacher. Hence, the drama manager is in charge of the beginning and the end of the story, as well as the transitions in-between.

A small-scale evaluation was run in which eleven children had to experience the emergent version of FearNot! and then answer some questions. These results were compared to a previous survey on the scripted version. In the end, the new emergent version was felt as more realistic and with

more believable characters.

2.1.4 Madame Bovary

More of a proof-of-concept than a complete work, a very ambitious project is the implementation of the classic French novel *Madame Bovary* by Gustave Flaubert [16] as an immersive IS system, where the user takes the role of one of the characters and influence the unfolding of the story [8]. The background focuses on one episode of the novel, which consists on the love affair between Emma Bovary and Rodolphe. In particular, the scene in which Emma and Rodolphe (whose role is played by the user) meet after they started their affair offers good opportunities for interaction, since Rodolphe's response will determine the outcome of their affair and the subsequent unfolding of the narrative. What makes this IS system immersive is the type of interaction that offers. In fact, the IS system consists in a CAVE-like environment, in which the user stands in a cubic room with screens on the walls, and the narrative unfolds as a real-time stereoscopic 3D animation. The user has a hand and head tracking system, so his/her position and direction of sight (and hence that of Rodolphe) is tracked and its story impact is analyzed. Furthermore, the user can speak in natural language, since there is an advanced speech recognition software that listens to what the user says. The interaction is then multimodal. For instance, when Emma proposes Rodolphe to runaway together, the user can either turn its back to her or say something like: "You should not be one of those frivolous women" to express criticism, generating embarrassment and commiseration in Emma.

This is a character-driven IS system, since all characters are autonomous artificial intelligence agents with their own planner to achieve their goals. This is possible mainly because the whole story is quite short and hence a complex drama manager is not necessary. The main novelty of this IS system is the possibility to let the users live their experience in a very immersive environment, not only allowing to dive into a full 3D world but also interacting in a very natural way, by speech and movements. Many IS systems are going towards this direction, even the aforementioned Façade with AR-Façade [14].

Since this is a proof-of-concept, it is not possible to apply traditional user evaluation methods. The challenge is to implement this IS system to a level of technical performance (real-time response, accuracy) that will allow future evaluations.

2.2 Related work on evaluation

Our main inspiration comes from *Towards Automatic Story Clustering for Interactive Narrative Authoring* [6], one of the first papers with the idea of story clustering. They proposed a semi-automatic narrative analysis by a meaningful clustering of narratives into groups with similar stories.

Figure 2.2: SimDate3D Level Two screenshot showing Thomas and Barbara in the park with comic baloons above their heads having a conversation.

To evaluate their methodology, they conducted experiments on their character-driven IS system SimDate3D (SD) level one and two. SD level one is a simple dating 3D game where the goal of the player is to achieve that a couple – Thomas and Barbara – gets to the cinema. The game comprises a sketchy conversation through comic-like bubbles and the player partially controls Thomas, deciding the way he walks (near/far from Barbara, in a weird/normal way). There are three possible endings:

1. characters get to the cinema safely;

2. characters get angry and part;

3. characters interaction is too positive, so they decide to skip the cinema and go home.

The sequel SD level two comprises an extended scenario, where Thomas is dating two girls at the same time, Barbara and Nathaly. The player still controls Thomas, having the possibility of choosing which girl to hang out with and where to go. Eventually, all three characters meet and engage in an argument, that can have four possible outcomes:

1. Thomas staying with Barbara and breaking up with Nataly;
2. Thomas staying with Nataly and breaking up with Barbara;
3. both girls breaking up with Thomas;
4. Thomas staying in the relationship with both girls.

All characters emotions are modeled through the already mentioned OCC emotion theory [35]. They are autonomous agents and there is no drama manager, hence the IS system is purely character-driven.

The authors first generated a number of play session from their IS system, either by human players or random players. Then, using a clustering algorithm (in their case the very simple yet effective k-means) they divided the play sessions into groups, using appropriate story distance measures (they tried with three string-based distances and a tension curves-based one). This way, even if the story space is infinite, the output is a predetermined number of clusters, that allow for only a few play sessions per cluster to be inspected directly. Furthermore, they evaluated the resulting clustering using various methods, such as the proportion of play sessions with the same final. We hugely rely on their methodology, with some improvements.

Another interesting paper, even if not directly related to interactive storytelling, is *The emotional arcs of stories are dominated by six basic shapes* [38]. The authors wanted to objectively test aspects of the theories of folklorists, specifically the commonality of core stories within societal boundaries. Hence, they analyzed a corpus of books and extracted from each of them an emotional arc, inspired by Kurt Vonnegut's *Shapes of Stories* [46]. To do that, they created LabMT, a database that associates English words with an happiness score ranging from 1 to 9. Then, they computed an average happiness value for overlapping blocks of words from a book. The succession of values denotes what they called an emotional arc. They found a connection between spikes in the emotional arc and

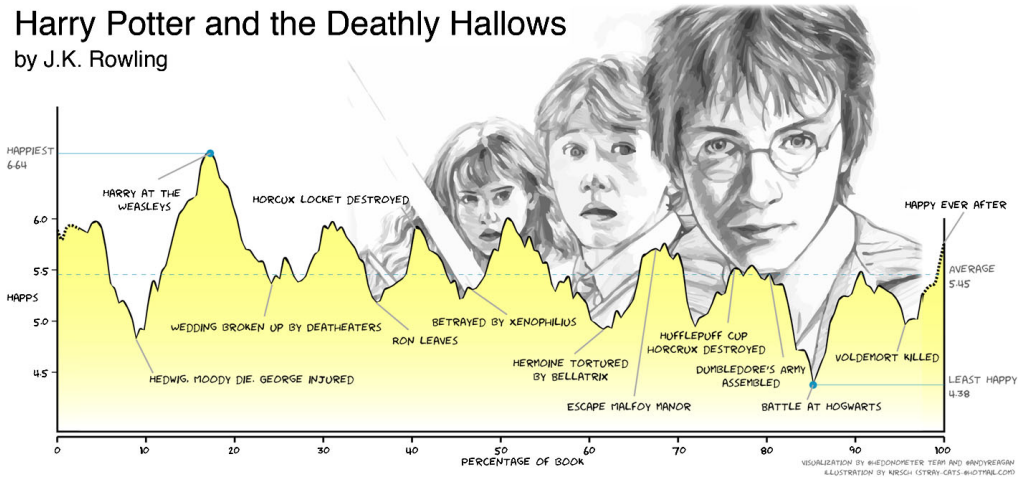


Figure 2.3: Annotated emotional arc of *Harry Potter and the Deathly Hallows* by J.K. Rowling [40].

important plot points. Using various clustering algorithms, such as hierarchical clustering, they divided the stories into similar groups and found that in each group stories would have emotional arcs with similar shapes, that roughly approximated the six emotional arc archetypes. For each of these six core emotional arcs, they also examined the closest characteristic stories in publication today and found that particular emotional arcs enjoy greater success, as measured by downloads.

We also use curves, in our case tension curves, to divide stories into groups and use them as a possibly defining feature for each group. Furthermore, we try to find particular tension curves that enjoy greater success, as measured by quality given by user surveys.

2.3 GEMEP emotional system

We rely on emotions to build tension curves. Different computational systems have been proposed over the last two decades, for example to generate corpora consisting of emotion expressions, that is a collection of audio, image or video contents of actors expressing a particular set of emotions.

All corpora contain most if not all of the six basic emotions: anger, disgust, fear, happiness, sadness, and surprise. They are considered of major importance because they are expressed through specific prototypical facial expressions that are universally recognized [15].

In GEMEP more subtle differentiations within emotion families were introduced. The GEneva Multimodal Emotion Portrayals (GEMEP) is a multimodal corpus of emotion expressions featuring audio and video recordings of actors portraying several affective states [5]. Thanks to its syncretic and methodologically robust design, this model is especially suitable to annotate the affective content of media, that is exactly what we need.

In GEMEP there are twelve emotions: anxiety (worry), amusement, cold anger (irritation), despair, hot anger (rage), interest, joy (elation), panic fear, pleasure, pride, relief, sadness (depression). These are divided into different groups based on two emotional dimensions: *polarity* (positive or negative) and *intensity* (high or low). The combination of these two dimensions give rise to four groups of three emotions each, as we see in Table 2.1.

Positive polarity	Negative polarity
High intensity	High intensity
• joy	• hot anger
• amusement	• panic fear
• pride	• despair
Low intensity	Low intensity
• relief	• irritation
• interest	• anxiety
• pleasure	• sadness

Table 2.1: The four groups of emotions in the GEMEP emotional system.

As we will see, our IS system, DoppioGioco, relies on this emotional system for reasons that will become apparent later.

2.4 DoppioGioco and Hot Bread

DoppioGioco is a story-driven IS system targeted at creators of narrative contents [12]. Its peculiarity is that the story designer may also be a user of the system: in the offline mode he/she creates the interactive story, while in the online mode he/she tests the flow of the story with a simulated audience. This way the user acts like an improvising actor, deciding at each time the continuation of the story based on the audience reaction.

2.4.1 Functioning of the interactive storytelling system

As already stated, the IS system is composed of an offline mode and an online mode.

The *Story Manager* is the tool for editing the story units offline and organizing them in a plot. Each unit consists of an audiovisual clip and a set of metadata elements describing it, such as title and textual description. For each unit, the author has to provide the information needed to the story engine to create a consistent story at runtime: the precedence relations with the other units, needed to generate a causally motivated story, and the emotions attached to the unit, needed to account for the response of the audience. This is the tool used exclusively by the story designer and will be used by him/her when redesigning the interactive story.

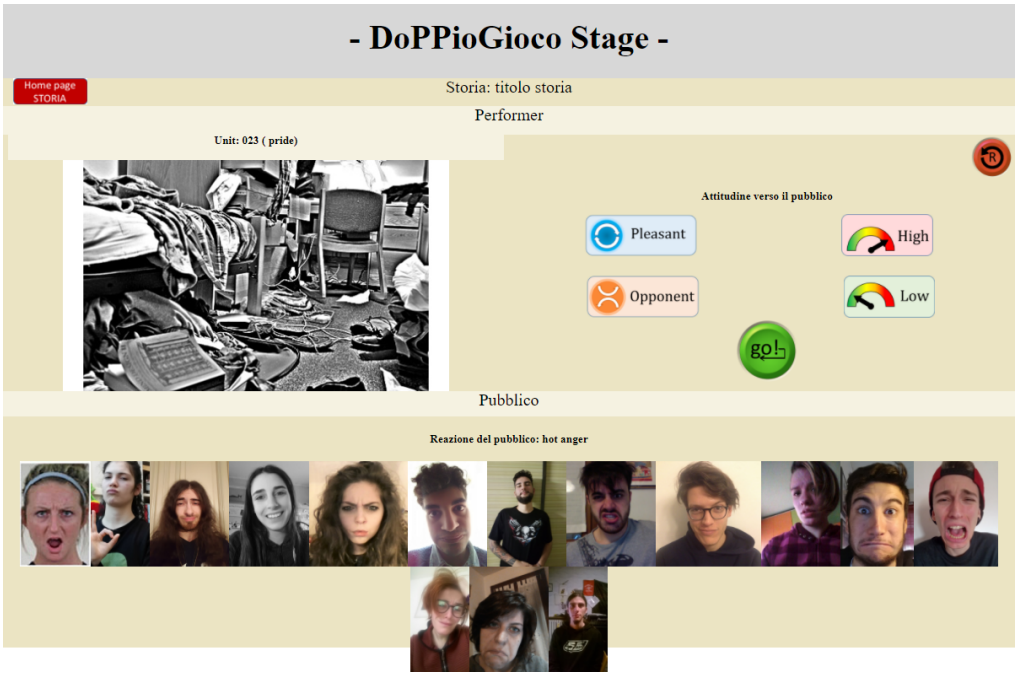


Figure 2.4: Screenshot of the DoppioGioco Stage Manager.

The *Stage Manager* lets the user experience the interactive story. Starting from an initial unit, the story begins to be told, unit by unit. After each unit, the audience reacts by displaying a set of emotions: like a real audience, the individuals who compose the public may react differently from each other, so the IS system computes the most frequent emotion. At this point, the author can decide to please or oppose the audience with high or low intensity. After selecting the reaction type, the IS system chooses the next unit accordingly. If several continuations are compatible with the current story unit, one is chosen randomly. The reaction of the audience is simulated by a random mechanism.

Figure 2.4 shows a screenshot of the Stage Manager. On the upper left

corner the unit clip is displayed, showing also the unit number and its associated emotions. On the upper right corner the user can select whether to be pleasant/opponent with high/low intensity with respect to the audience reaction. At the bottom, a button (not visible in the figure) allows to generate the audience reaction that will be displayed as a collection of faces representing different emotions. The most frequent one is written on top of them.

2.4.2 Annotating emotions

DoppioGioco relies on the GEMEP emotional system. The story units and the reaction of the audience are both annotated with the twelve emotion categories contemplated by GEMEP. Not only it is an emotional system designed for describing emotions in performance but, most importantly, it relies on polarity-based account on emotions. Hence, it is perfectly suited to deal with the polarity of the reaction to the audience's responses: the user can decide to please the audience (same polarity) or oppose them (opposite polarity). The reaction rule simply works as follows:

1. detect the polarity of the audience overall emotion, that may be positive or negative (according to GEMEP emotion groups);
2. if the user chose to be pleasant, select the same polarity; otherwise select the opposite polarity;
3. tune the intensity level of the reaction emotion to the intensity (low or high) selected by the user;
4. select the GEMEP group with the selected polarity and intensity (positive/negative polarity, high/low intensity).

In order to emphasize the elements of arbitrariness that characterize a live, interactive performance, a random element was introduced at the selection of the emotions within the selected group: given the available units, the IS system randomly selects the next unit among the available ones, so that the user does not have complete control on the selection.

For example, in Figure 2.4 the overall emotion of the audience is hot anger, which belongs to the "negative, high intensity" group. Suppose that the user decides to be pleasant and to respond with a low intensity emotion: then the selected group will be "positive, low intensity". Hence, the IS system will choose randomly from all the following units annotated with

emotions belonging to the "positive, low intensity" group, namely relief, interest and pleasure.

2.4.3 Hot Bread and the story graph

While DoppioGioco is the IS system, we can build a number of different interactive stories with it. The one provided here is called *Hot Bread*, a romance set in an American small town in the '70s, Stob. In this story the characters go through changes in their personal lives as a result of their professional and relational crossroads. The story is more inspired to serial formats (such as TV fictions) than to the classical reverse U-shaped stories that characterizes traditional storytelling.

We extracted the story graph from the database of the IS system.

The story graph is represented in Figure 2.5. There is a total of 107 units, with a single initial one (000) and 35 endings. This give rise to a total of 55,888 possible linear stories. All units have exactly four continuations. Each unit (except the initial one) has a color corresponding to the group at which the unit emotion belongs to, that is blue for negative polarity, red for positive polarity and varying saturation for intensity, namely light variant for low intensity and dark variant for high intensity.

Further statistics about the data set of all possible linear stories are shown in Table 2.2. In Figure 2.6a we can see how all linear stories are distributed among all possible endings, i.e. for each final unit how many stories have as ending that unit. In Figure 2.6b we can see how many linear stories have a certain length as an histogram. We can see that most stories have a length of 11 units.

Number of units	107
Number of linear stories	55,888
Story beginnings	1
Story endings	35
Story length average	10.652
Story length mode	11
Story length minimum	5
Story length maximum	12

Table 2.2: Some statistics about *Hot Bread*.

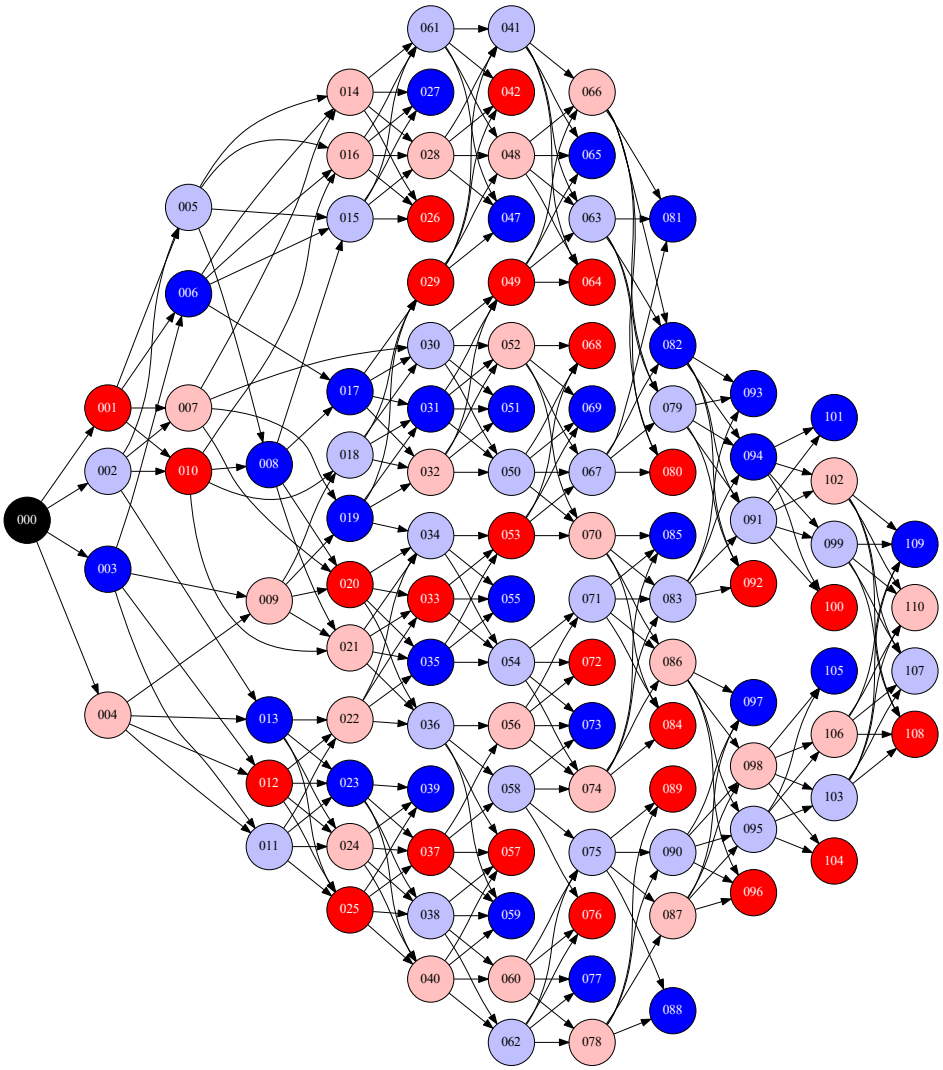
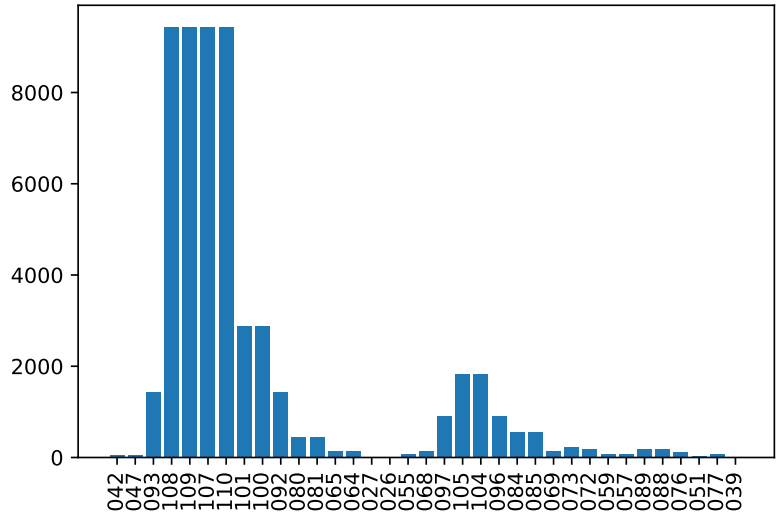


Figure 2.5: Story graph of *Hot Bread*. The color of the unit indicates at which emotion group its annotated emotion belongs to: blue for "negative, high intensity"; light blue for "negative, low intensity"; light red for "positive, low intensity"; red for "positive, high intensity".

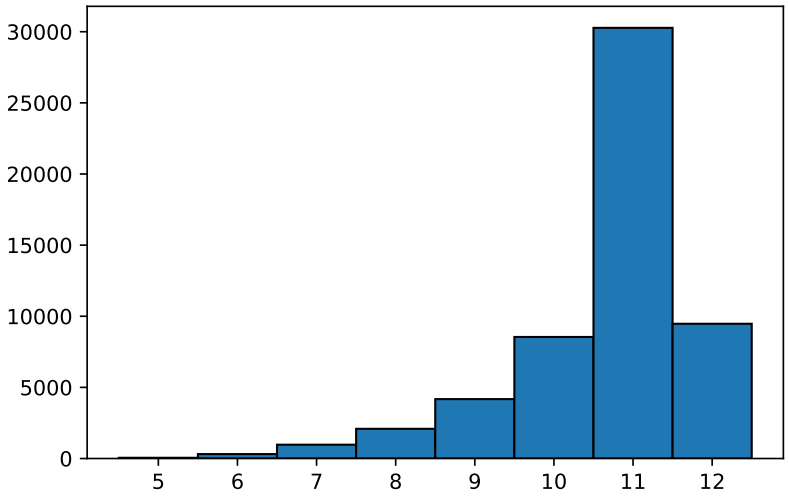
2.5 The next steps

We reviewed some notable IS systems, each with its own distinctive trait. Finally, we also introduced the IS system that is the object of our analyses, DoppioGioco, and its interactive story, *Hot Bread*. We showed some statistics and its story graph.

As we have seen, there are several ways to evaluate an IS system, but we



(a) Distribution of stories among all possible endings.



(b) Histogram of the story length distribution.

Figure 2.6: Statistical plots about linear stories extracted from *Hot Bread*.

rely on the one proposed by Bída et al. [6]. This needs to be adapted to DoppioGioco, that is a story-driven IS system, and be integrated with the results of a user survey with the aim to obtain a quality score for stories.

In this perspective, in the following chapters we investigate a proper

clustering algorithm, a method to measure distances between stories and a way to create a well-designed user survey.

Chapter 3

Clustering

Clustering is the task of grouping a set of objects into groups, called clusters, in such a way that the average similarity of objects in the same group is higher than the group average similarity of objects in different groups. The concept of similarity may be expressed in different ways. Distance, density or distribution-based similarities are all in common use.

We use as clustering method *k-medoids*, that is related to the well known *k-means*. We also introduce the *silhouette* as a way of deciding how many clusters to choose.

3.1 k-means

k-means [33] is a clustering method that aims to partition n objects into k clusters. The number of clusters k is a user-given parameter.

3.1.1 Mathematical background

Given a set of d-dimensional real vectors $(x_1, x_2, \ldots, x_n)$, k-means aims to partition this n objects into $k \leq n$ clusters $\boldsymbol{S} = (S_1, S_2, \ldots, S_k)$ to minimize the within-cluster sum of squares (WCSS), i.e. the distance between objects in the same cluster. Formally, we want to find a clustering solution $\hat{\boldsymbol{S}}$, i.e. a vector of clusters, over the set $\mathcal{S}$ all possible clustering solutions, such that

$$\hat{\boldsymbol{S}} = \arg\min_{\boldsymbol{S} \in \mathcal{S}} W(\boldsymbol{S}),$$

with the WCSS $W(\boldsymbol{S})$ defined as

$$W(\boldsymbol{S}) = \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 = \sum_{i=1}^{k} |S_i| \operatorname{Var}(S_i),$$

in which $||\cdot||$ denotes the Euclidean norm, μ_i the mean of the objects in the cluster S_i, and $|\cdot|$ the cardinality of a set.

Since the total variance, that is

$$T = \frac{1}{n}\sum_{i=1}^{n} ||x_i - \mu||^2 = \mathrm{Var}(x_1, x_2, \ldots, x_n),$$

depends only on the objects and not on the clustering solution, we can define the between-cluster sum of squares (BCSS), i.e. the distance between objects in different clusters, as

$$B(\boldsymbol{S}) = T - W(\boldsymbol{S}),$$

and note that minimizing the WCSS is equivalent to maximizing the BCSS, hence obtaining well-formed clusters.

This problem is computationally hard, hence a straightforward implementation of this method is not possible. However, there are efficient heuristic algorithms that converge quickly to a local optimum, such as Lloyd's algorithm.

3.1.2 Lloyd's algorithm

When people think of k-means they usually refer to its most famous implementation, Lloyd's algorithm [17, 32].

Given an initial set of k cluster means $\mu_1, \mu_2, \ldots, \mu_k$, the algorithm alternates between two steps: an assignment step and an update step.

In the assignment step, every object is assigned to the cluster whose mean is the least distant (in Euclidean sense) from the object, specifically

$$x \in S_i^{(t)}, \quad i = \underset{1\le j\le k}{\arg\min}\, ||x - \mu_j^{(t)}||^2,$$

where S_i is the cluster the object x is assigned to, while t indicates the iteration we are in.

In the update step, compute the new means based on the newly formed clusters,

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|}\sum_{x\in S_i} x.$$

The algorithm converges when, in the assignment step, clusters no longer change. It can be shown that an iteration of Lloyd's algorithm can never increase the within-cluster sum of squares, hence we are assured to reach a

stationary point, that is a local optimum, where no further improvements are possible. However, this is not guaranteed to be the global optimum.

Furthermore the algorithm may stop converging if one uses a different distance other than the Euclidean one. Various modification have been proposed to use arbitrary distances, one of them being k-medoids.

3.1.3 Initialization problems

We have seen that Lloyd's algorithm needs an initial set of k cluster means. One simple yet effective way of doing that is to randomly assign a cluster to each data object, a method sometimes called *random partition* [21]. An implementation is available in Algorithm 1.

The algorithm is very sensitive to initial means, since there may be different local optima. To overcome this problem a common practice is to run it multiple times and then average the results, taking advantage of its speed.

As for the number of clusters k, its value must be known beforehand and there is no straightforward way to choose it. We tackle this problem later using the silhouette value.

Algorithm 1: Lloyd's implementation of k-means with random partition initialization.

Input: data objects $\{x_i\}_{i=1}^n$; number of clusters $k \in \mathbb{N}$.
Output: cluster means $\mu_1, \mu_2, \ldots, \mu_k$.

randomly initialize $\mu_1, \mu_2, \ldots, \mu_k$;
repeat
  assign each x to S_i, $i = \arg\min_{1 \le j \le k} ||x - \mu_j||^2$;
  for $j = 1$ *to* k **do**
    $\mu_j = \frac{1}{|S_j|} \sum_{x \in S_j} x$;
until *no change in* $\mu_1, \mu_2, \ldots, \mu_k$;
return $\mu_1, \mu_2, \ldots, \mu_k$.

3.1.4 Computational complexity

For each iteration of Lloyd's algorithm, each of the n data object must be compared with each of the k clusters. This means that, if data objects are d-dimensional, the computational complexity of the algorithm is $\mathcal{O}(knd)$ per iteration.

The number of iterations changes from run to run, but the algorithm is expected to converge quickly and usually a maximum number of iterations

is chosen to avoid excessive run times.

3.2 k-medoids

k-medoids [25] is a clustering method that aims to minimize the distance between objects labeled to be in a cluster and an object designated as the center of that cluster. In contrast with k-means, the center of a cluster is always an object belonging to that cluster (called a *medoid*), moreover an arbitrary distance function may be used.

3.2.1 PAM algorithm

The most common implementation of k-medoids is the Partitioning Around Medoids (PAM) algorithm [25].

In the initialization step, a set of k random data objects $m_1, m_2, \ldots, m_k$ are chosen as medoids, i.e. centers of clusters.

In the assignment step, every object is assigned to the cluster whose medoid has the least distance d (that can be either the Euclidean distance or an arbitrary one), hence

$$x \in S_i^{(t)}, \quad i = \arg\min_{1 \le j \le k} d(x, m_j^{(t)}),$$

where S_i is the cluster the object x is assigned to, while t indicates the iteration we are in.

In the update step, compute the new medoids of the new clusters,

$$m_i^{(t+1)} = \arg\min_{x \in S_i^{(t)}} \frac{1}{|S_i^{(t)}|} \sum_{y \in S_i^{(t)}} d(x, y),$$

i.e. choose as new medoid the cluster member that has the minimum average distance with respect to all other members. Note that minimizing the mean or the sum of distances does not change the result. A full implementation

is shown in Algorithm 2.

Algorithm 2: PAM implementation of k-medoids with random initialization.

Input: data objects $\{x_i\}_{i=1}^n$; number of clusters $k \in \mathbb{N}$.
Output: medoids $m_1, m_2, \ldots, m_k$.

```
randomly initialize m_1, m_2, ..., m_k;
repeat
    assign each x to S_i, i = arg min_{1≤j≤k} d(x, m_j);
    for j = 1 to k do
        foreach x ∈ S_j do
            swap x and m_j and recompute the cost, i.e. sum of
             distances between each cluster members and x;
            if the cost increases, undo the swap;
until no change in m_1, m_2, ..., m_k;
return m_1, m_2, ..., m_k.
```

3.2.2 Time complexity

Even if k-medoids has a lot of benefits over k-means, namely the possibility to use an arbitrary distance function and the greater robustness to noise and outliers, it has a major drawback: the time complexity of its implementation PAM is bigger than that of Lloyd's algorithm.

In fact, if data objects are d-dimensional, Lloyd's algorithm is $\mathcal{O}(knd)$ per iteration, while PAM is $\mathcal{O}(k(n-k)^2 d)$ per iteration. The reason is that to compute a new medoid the algorithm must compare every pair of objects in a cluster, as opposed to Lloyd's algorithm in which, to compute the mean, a single loop over all objects is enough. Hence, the time increases quadratically as the size of data n increases and this may be a problem with big data sets. As a way to improve performance, it is common to store all pairwise distances between data objects in a matrix, saving a lot of computational time at the expense of memory. In fact, as a result the time complexity is reduced to $\mathcal{O}(k(n-k)^2)$ and hence is faster than the standard implementation as the dimensions increase.

3.3 Silhouette

The silhouette is a method of interpretation and validation of clusters consistency.

The silhouette value is a measure of how similar a data object is to its own cluster (coherence) compared to other clusters (separation). The value ranges from -1 to $+1$, with a high value indicating that the object is well matched to its own cluster and is far from neighbor clusters. If most objects have a high value the clustering is a consistent one, while if most objects have a low value the clustering is poor and it may have too many or too few clusters.

It can be applied to any type of distance-based clustering technique, since the only needed inputs are data objects, cluster membership and distance measure used in the clustering.

3.3.1 Definition

Assume the data has been assigned to k clusters. Let x_i be an arbitrary data object, $j(i)$ be the index of the cluster it belongs to,

$$j(i) = k \iff x_i \in S_k,$$

and $d(x_i, S_k)$ be the average distance of x_i to the data objects in the cluster S_k,

$$d(x_i, S_k) = \frac{1}{|S_k|} \sum_{y \in S_k} d'(x_i, y),$$

with d' being the distance function used in the clustering.

Let $a(x_i)$ be the average distance between x_i and the objects in its own cluster,

$$a(x_i) = d(x_i, S_{j(i)}),$$

and $b(x_i)$ the average distance between x_i and the objects in its nearest cluster (also called neighbor cluster),

$$b(x_i) = \min_{k \neq j(i)} d(x_i, S_k).$$

We would expect $a(i)$ to be considerably smaller than $b(i)$, but this is not guaranteed, especially if the clustering is poor. Hence, an appropriate measure of the goodness of a clustering is $b(x_i) - a(x_i)$, which is then normalized so that the value ranges in the interval $[-1, 1]$.

We define the *silhouette value* of a data object x_i as

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}$$

so that

$$-1 \leq s(x_i) \leq 1.$$

While this is related to a single data object, it may be desirable to have a value concerning the whole clustering, hence we can simply take the *average silhouette value*

$$\overline{s} = \frac{1}{n} \sum_{1 \leq i \leq n} s(x_i).$$

3.3.2 Silhouette plot

The average silhouette value is a good indicator of the overall consistency of the clustering, but individual silhouette values may provide additional insights into each cluster. Indeed, a very powerful visualization technique is the *silhouette plot*.

In the silhouette plot every cluster is drawn separately from the others and usually has its own color. Then, for every object in a cluster its silhouette value is represented as a horizontal bar, ordered from the highest to the lowest value. This way, good clusters are the ones with a more regular shape, with little to no hollows. On the opposite, bad clusters have rapidly decreasing values, going fast to zero or, even worse, to negative values. In fact, a negative silhouette value means that, in average, that object is nearer to its neighbor cluster than to the one it actually belongs to.

In Figure 3.1 a silhouette plot of k-means clustering on sample data can be seen. On the right, a visual plot of the two-dimensional data is shown, so that it is possible to see in which way a clustering relates to its silhouette value.

3.4 Conclusions

We reviewed k-means and k-medoids, two well-known clustering algorithms, and some of their most efficient implementations, respectively Lloyd's algorithm and PAM algorithm. Both of them require to choose the number of clusters and we showed that a possible solution for this problem is to use silhouette scores. We use PAM algorithm to perform the clustering of linear stories, as Bída et al. did in their article [6]. By using k-medoids we cluster all linear stories around a specific tension curve, that is the one of the medoid of the cluster.

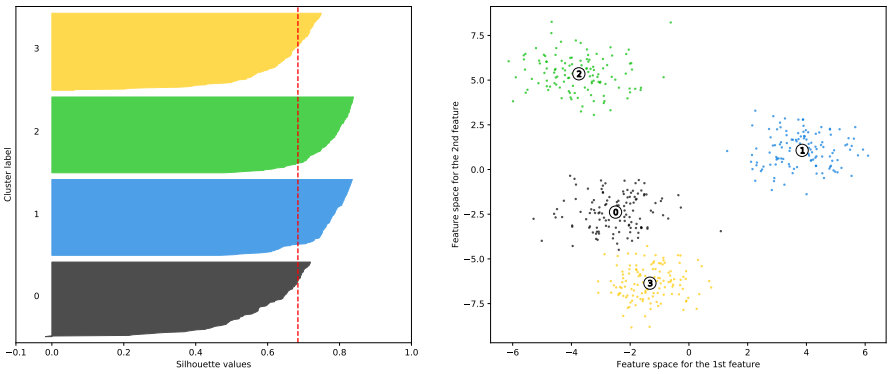


Figure 3.1: On the left, silhouette plot of k-means clustering on sample data. The red dotted line corresponds to the average silhouette value. On the right, visual representation of the sample data with matching colors.

Chapter 4

Tension curves and distances

As we have seen clustering methods such as k-means or k-medoids require a distance function to be defined. This is almost straightforward for geometric data sets, in which the distance is usually Euclidean. However, we want to cluster stories, hence the question: how to compute distance between two stories? This is a non-trivial task that can be addressed in several ways. We review the proposals made in Bída et al. [6] and go further in their direction.

4.1 String distances

A simple approach is to transform a story into *strings*, as proposed in Bída et al. [6] for action strings. The idea is to map each one of the twelve emotions considered in DoppioGioco into a different letter. We can associate anxiety with "a", amusement with "b" and so on.

The main advantage of this method is that string metrics have been well studied and many distances are available: Levenshtein distance [31], Jaro distance [23], Jaro-Winkler distance [47], just to name a few. Furthermore, since the sequence of emotions defines a story, the emotion string should do as well.

However, this may not represent accurately a story. In fact, as an example the distance between joy and amusement (both positive polarity, high intensity) would be the same as that between joy and sadness (negative polarity, low intensity). Even if we just associated a letter to each emotion group, rather than single emotions, we would expect to have different distances between "positive polarity, high intensity" and "positive polarity,

low intensity" emotion groups and between "positive polarity, high intensity" and "negative polarity, high intensity" emotions groups, for instance.

The limitations of this choice are evident, as already pointed out in Bída et al. [6], so we explore a more flexible and powerful technique, namely tension curves.

4.2 Tension curves

We define as *tension curve* a graphical representation of the dramatic structure of a story. A dramatist may argue that a tension curve should always be positive as it measures how intense is the emotion felt (negative or positive alike), but we preferred to follow the idea behind the shapes of stories proposed by Kurt Vonnegut [46] and maintained in Bída et al. [6].

Freytag published one of the most influential studies of dramatic structure of stories in his book [18], where he expanded Aristotle's analysis in his *Poetics*, asserting that a drama is divided into five parts (exposition, rising action, climax, falling action, and dénouement) that form the so called Freytag pyramid. Most narratives of mainstream entertainment systems follow this structure.

Not surprisingly, a lot of IS systems have been modeled after tension curves, such as Façade [34] that takes as reference Freytag curve. We want to extract tension curves from our own IS system, so to check whether successful stories follow a particular shape.

4.2.1 Tension evaluation

Since we already have annotated emotions for each story unit, the issue is how to convert emotions into numbers. Then, the tension curve is simply the piecewise linear function defined by those individual values.

The function that maps emotions into tension values, or *tension evaluation function*, is non-trivial to find and is IS system-specific. In our case, the GEMEP emotional system provides us with four very distinguished emotion groups. Inside each group, all emotions have roughly the same polarity and intensity, so each of them has the same tension value.

Our aim is now to associate a tension value to each of the four emotion groups. A very straightforward yet effective way is to map the positive/negative polarity to the sign $+$ and $-$, respectively, and the low/high intensity to the numbers 1 and 2, respectively, as can be seen in Table 4.1.

Emotion group	Emotions	Tension value
positive polarity high intensity	joy amusement pride	+2
positive polarity low intensity	relief interest pleasure	+1
negative polarity low intensity	irritation anxiety sadness	−1
negative polarity high intensity	hot anger panic fear despair	−2

Table 4.1: The tension evaluation function for the four groups of emotions in the GEMEP emotional system.

4.2.2 Moving average

We have now a way of building a piecewise tension curve for each story. However, it is possible to further improve the curve by smoothing considering the moving average of its values.

A *moving average* is a calculation to analyze data objects by creating series of averages of different subsets of the full data set. This is often used to smooth piecewise curves, as to remove random oscillations that may occur in the data. We go through three types of moving averages: cumulative, weighted and exponential. We always assume to have an ordered sequence of data objects $(x_i)_{i=1}^n$.

Cumulative moving average

In *cumulative moving average* for each object the average is taken over all previous objects until that object. Suppose we have as objects $(x_i)_{i=1}^n$, the corresponding cumulative moving average is defined as

$$\begin{aligned} \mathrm{CMA}_1 &= x_1, \\ \mathrm{CMA}_2 &= \frac{x_2 + x_1}{2}, \\ &\dots \\ \mathrm{CMA}_n &= \frac{x_n + x_{n-1} + \dots + x_1}{n}, \end{aligned}$$

so it is the simple average up until a certain object.

Weighted moving average

In *weighted moving average* the average is taken over all previous objects, each with a specific weight. Usually the weights decrease in arithmetical progression, i.e.

$$\begin{aligned}
\mathrm{WMA}_1 &= x_1, \\
\mathrm{WMA}_2 &= \frac{2x_2 + x_1}{2+1}, \\
&\dots \\
\mathrm{WMA}_n &= \frac{nx_n + (n-1)x_{n-1} + \dots + x_1}{n + (n-1) + \dots + 1}.
\end{aligned}$$

An arbitrary sequence of weights may be chosen.

Exponential moving average

In *exponential moving average* the average has weights that decrease exponentially, never reaching zero. A simple definition is given by recursion, i.e

$$\mathrm{EMA}_i = \begin{cases} x_1 & \text{if } i = 1 \\ \alpha x_i + (1-\alpha)\mathrm{EMA}_{i-1} & \text{if } i > 1, \end{cases}$$

where α is a constant smoothing factor between 0 and 1, that represents the degree of weighting decrease.

This type of moving average is related to weighted moving average, in fact from the recursive formula we obtain

$$\begin{aligned}
\mathrm{EMA}_n &= \alpha x_1 + (1-\alpha)\mathrm{EMA}_{n-1} \\
&= \alpha x_1 + (1-\alpha)(\alpha x_2 + (1-\alpha)\mathrm{EMA}_{n-2}) \\
&= \alpha x_1 + \alpha(1-\alpha)x_2 + (1-\alpha)^2\mathrm{EMA}_{n-3} \\
&= \dots \\
&= \alpha x_1 + \alpha(1-\alpha)x_2 + \dots + \alpha(1-\alpha)^{n-1}x_n \\
&= \frac{x_1 + (1-\alpha)x_2 + \dots + (1-\alpha)^{n-1}x_n}{1/\alpha},
\end{aligned}$$

and as $n \to \infty$, since $1/\alpha = \sum_{i=0}^{\infty}(1-\alpha)^i$, we finally get

$$\mathrm{EMA}_\infty = \frac{x_1 + (1-\alpha)x_2 + (1-\alpha)^2 x_3 + \dots}{1 + (1-\alpha) + (1-\alpha)^2 + \dots},$$

that is in fact a weighted moving average with infinite weights.

4.2.3 Tension curve smoothing

We want to use moving averages to smooth our piecewise tension curves. The advantage of moving averages for smoothing tension curves is twofold.

Spikes in the tension curves due to sudden changes in emotions from one unit to another are not desirable. Intuitively, a tension curve should be as smooth as possible, even if critical events may still produce large variations. Moreover, a smooth curve allows for better and more meaningful clusters.

Furthermore and most importantly, the moving average also makes sense from a dramatic point of view. In fact, it is reasonable to assume that previous felt emotions change the way we experience new units and their associated emotions [39]. If we feel sadness and then happiness we may have a bitter-sweet feeling, while if instead we go from a happy emotion to another happy emotion we may have a more pure form of happiness.

Because of this, we choose to apply a weighted moving average smoothing with exponentially decreasing weights. This way, the current emotion is the prevalent one, while previously felt emotions still influence our present emotional state in a way that quickly decreases as far as they get from the present emotion. More precisely, we choose an exponential decrease in base 2 just for simplicity.

Suppose, for example, to have a tension curve with values $x_1, x_2, \ldots, x_n$, then the smoothed value y_i of the tension curve at unit i is

$$\begin{aligned} y_i &= \frac{2^{i-1}x_i + 2^{i-2}x_{i-1} + \cdots + 2x_2 + x_1}{2^{i-1} + 2^{i-2} + \cdots + 2 + 1} \\ &= \frac{2^{i-1}x_i + 2^{i-2}x_{i-1} + \cdots + 2x_2 + x_1}{2^i - 1}. \end{aligned}$$

This way we obtain smoother tension curves, that are easier to cluster and still meaningful from a dramatic perspective.

4.3 Tension curves distance

After choosing a suitable tension evaluation function, the next step is to decide what type of distance to use. In fact, we have piecewise tension curves, that are essentially vectors of real numbers, but we still need to choose an appropriate distance, such as the standard Euclidean distance or the Manhattan distance.

4.3.1 Euclidean and Manhattan distance

The most common way of measuring the distance between vectors of real numbers $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ is the Euclidean distance, simply defined as

$$d_2(x, y) = ||x - y||_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}.$$

Intuitively, for two points in a two-dimensional space it simply measures how far they are as the crow flies. This is the main choice for geometric data sets as it is the most natural way of describing a distance between two points.

Another popular alternative is the Manhattan distance, defined as

$$d_1(x, y) = ||x - y||_1 = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|.$$

Here, for two points in a two-dimensional space it measures how far we must travel if we can only take directions along the axes.

We use the Manhattan distance as our distance of choice, since it is more robust. In fact, if two tension curves are exactly the same except for a few values which differ greatly, the Euclidean distance magnifies such differences more than Manhattan distance. Moreover, as the dimensions of the objects increase, the Euclidean distance becomes less meaningful than Manhattan distance and so the latter is usually preferred [1].

4.3.2 Expanded and warped curves

Another issue is that both distances may only be applied to vectors with the same length while, as we have seen, tension curves may have different lengths. In *Hot Bread* they range from 4 to 12 tension values each.

A straightforward but naive way of approaching this, as suggested in Bída et al. [6], is to simply expand the shortest tension curve by adding zeros until it matches the length of the longest tension curve. If we have the following tension curves

$$x = (x_1, x_2, \ldots, x_m),$$
$$y = (y_1, y_2, \ldots, y_n),$$

with $m < n$, the vector x is transformed into x'

$$x' = (x_1, x_2, \ldots, x_m, 0, \ldots, 0)$$

and the distance is simply $d(x, y) = ||x' - y||$.

However, we found this approach not realistic. In fact, since these tension curves represent stories, if we want to compare two of them, it would make more sense to compare the beginning of one story to that of the other, similarly for the ending. While the beginning of both stories is at position 1, the ending of the shorter story is at position $m < n$ and hence is compared to a middle unit of the longer story. This makes no sense from the dramatic point of view. Hence we want a method to normalize curves lengths, so that they have the same length but both beginnings and endings match.

There are however two issues:

1. how to choose the new values of the curve after changing the length, i.e. what type of *interpolation* to choose;

2. which length to choose.

Interpolation

Interpolation is a method of constructing new data points from within a range of discrete set of known data points. Several types exist, such as piecewise constant interpolation, linear interpolation, polynomial interpolation and spline interpolation. We review and use one the simplest ones, namely *linear interpolation.*

In linear interpolation, it is assumed that between each pair of points a straight line is drawn. In particular, if we have two points (x_1, y_1) and (x_2, y_2), the interpolant function is simply

$$y = y_1 + (x - x_1)\frac{y_2 - y_1}{x_2 - x_1}.$$

This type of interpolation is not very precise, however it is quick and easy. While a spline interpolation would probably allow for better results, it is safe to assume that between two units the tension does not change in odd and unpredictable ways and that it is not worth it to switch to a more sophisticated type of interpolation.

In our case, the independent variable is the unit index and the dependent variable the corresponding tension value. After interpolation, we have an interpolant function that allows us to generate another piecewise curve with arbitrary length, since any possible intermediate value can be computed.

Length choice

We still need to choose which is the final length of the tension curves.

The first approach would be to take the maximum length over all tension curves. Even if this is an immediate and apparently safe way, it may not be very accurate. In fact, as it is in our case, most of the tension curves have length 11, while the maximum is 12. Expanding a curve of length 11 to length 12 introduces a large error. In fact, since we have chosen linear interpolation, intermediate values may not be very precise and we should rely on them as little as possible. If we were to choose as target length 12, all 11 length stories (the majority of them) would be expanded to 12 values, with 10 of them intermediate ones (since 12 is not a multiple of 11 only the first and last values are exact ones) that are prone to error.

A possible solution is to take the least common multiple of all possible lengths, so that the resulting length is a multiple of all possible lengths and, hence, all tension curves are expanded. However, even this approach is far from ideal. In fact, we would still use a lot of intermediate values, even if in a less problematic way than before (since the target length is chosen to be a multiple of every possible story length), and, most of all, the resulting length could be a huge number, increasing too much the computational cost of the subsequent analysis. In our case, the set of possible lengths is $\{5, 6, 7, 8, 9, 10, 11, 12\}$, whose least common multiple is 27,720, that would become the target length. Furthermore, as we said before, an increase in dimensionality would make distances less meaningful.

Our approach was a different one, so that we could use the minimum possible number of intermediate values while still having a small length: picking the mode, i.e. the most frequent length over all possible tension curves, as target length. In our case, the mode is 11 and the full frequency table is displayed in Table 4.2. This method is as powerful as larger is the difference in frequency from the mode to all other values, as it is indeed for us, since that would in turn means that a large number of tension curves would not be neither squeezed nor expanded, minimizing the error.

Length	Number of stories
5	52
6	312
7	972
8	2088
9	4176
10	8544
11	30 272
12	9472

Table 4.2: Frequency table for story lengths of DoppioGioco's *Hot Bread* extracted linear stories.

Chapter 5

Evaluation of clusters quality

At this point we have a way to group together similar stories. However, it is not trivial to evaluate the resulting clustering. One simple approach is to use the aforementioned silhouette score, that assesses how much clusters are compact and well-formed. While it is undoubtedly desirable to have such features, we want a way to understand if clusters have a meaning also for IS system-specific purposes. In this case, we want to see if the clustering is able to separate between good and bad stories.

5.1 Quality of stories

Deciding whether a story is good or bad is not a trivial task. There is not actually a precise and universally recognized definition for this, but what we want to accomplish is the association of a score to a particular story, to express the engagement and the willingness of the reader to read more of the same or not.

For this purpose we design a user survey, so that we could have human feedback on the quality of stories. In the end, we want people to enjoy as much as possible stories and this is usually captured through engagement questionnaires, e.g. [43].

We decide to ask questions not only about a whole story but also about the single units it is made of. In fact, units are complete chunks of texts and it makes sense to evaluate them individually.

For each unit, we ask the user to give feedback about two main issues:

- what are the perceived main characters of the unit (to help us understand if the characters are well outlined);

- what emotional reaction is felt (to help us understand the quality of annotated emotions).

On the other side, at the end of the whole story we ask more specific questions, namely:

- what are the perceived main characters of the story;
- how much engaging, coherent and good was the story;
- whether the user is willing to read another similar story or not.

This way we have information about both single units and complete stories. In particular, we think that a good indicator of the quality of a story is whether the user enjoyed the story so much that he/she would like to read another similar one or not.

5.2 User survey design

Our user survey went through several iteration for refinement. In fact, it is not only important what questions to ask, but also the way they are asked, in order for the user to not be biased towards certain answers.

We use a custom web page that selects one story among the 55,888 possible ones and let the user read it unit by unit. We decide to limit the survey to a single story per user mainly to avoid the time to complete it to be too large - in fact, a single story can take as long as 40 minutes to be read completely.

After each unit, the user should answer the following questions as in Fig. 5.1:

- "Indica il personaggio o i personaggi più importanti di cui parla questa unità." ("Select the most important character(s) portrayed in this unit.") Thanks to a dropdown menu, users can select among a list of all characters mentioned in the unit (more than one selection is possible).
- "Valuta il tono generale della unità, se è negativo o positivo." ("Evaluate the general polarity of the unit, whether it is negative or positive.") Thanks to a slider, users can select one of the five allowed values, ranging from negative (1) to positive (5), expressing what is the perceived polarity of the emotions felt after reading the unit.

Figure 5.1: Unit rating page of the survey.

- "Valuta l'intensità delle emozioni che hai provato leggendo questa unità." ("Evaluate the intensity of the emotions you felt reading this unit.") Thanks to a slider, users can select one of the three allowed values, "Low", "Medium" and "High", referring to the intensity of the emotions felt.
- "Indica quali emozioni hai provato." ("Selects which emotions you felt.") Users can select up to three emotions from the list on the right and drag them into the list on the left, where they can also order them according to the perceived importance.

We decide to not to ask directly which emotion, between the 12 possible ones after which the IS system is built, is felt even if that would have been more straightforward for the comparison with the annotated ones. In fact, this could have confused the user introducing a bias, since the number of objects an average human can hold in working memory is 7 ± 2 [41] and 12

exceeds that limit. Instead, we opted for a separate rating for polarity and intensity, again according to our emotional system, and for the selection of up to three emotions in order of importance [48].

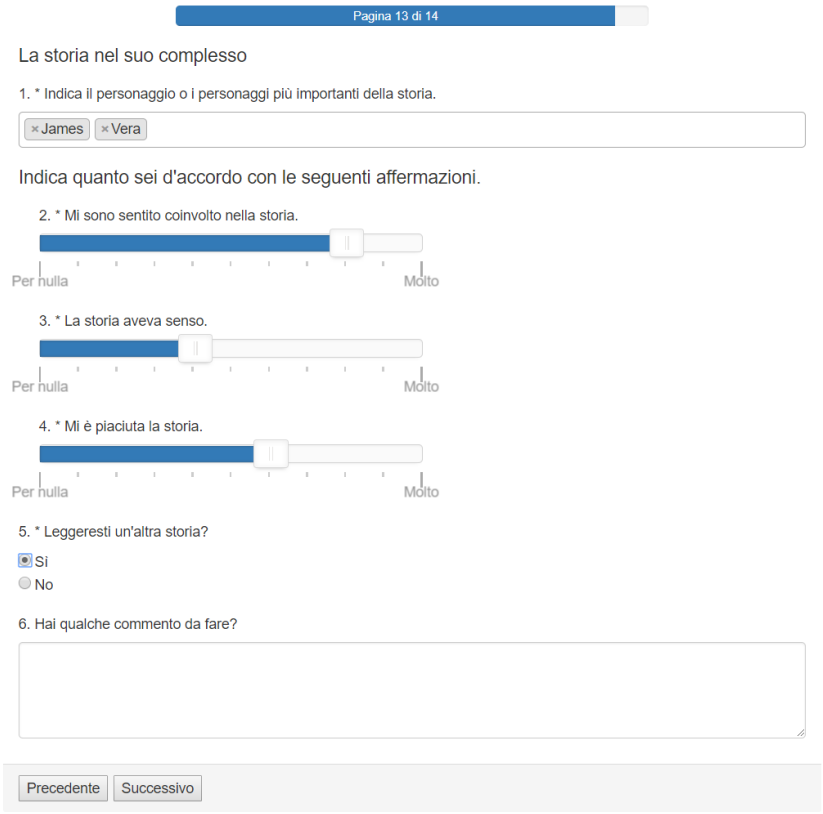


Figure 5.2: Story rating page of the survey.

After reading the whole sequence of units, users are asked about the story in its entirety as in Fig. 5.2:

- "Indica il personaggio o i personaggi più importanti della storia." ("Select the most important character(s) of the whole story.") Thanks to a dropdown menu, users can select among a list of all characters that appeared throughout the entire story, possibly selecting more than one.
- "Indica quanto sei d'accordo con le seguenti affermazioni." ("Please rate how much you agree or disagree with the following statements.") Users are provided with three statements: "Mi sono sentito coinvolto nella storia." ("I felt engaged by the story."), "La storia aveva senso." ("The story made sense.") and "Mi è piaciuta la storia." ("I enjoyed the story."). For each of them they could rate how much they agree

or disagree with the statement using a slider ranging from “Per nulla” (“Strongly disagree”) to “Molto” (“Strongly agree”).

- “Leggeresti un’altra storia?” (“Would you read another story?”) To evaluate how much users actually enjoyed the story, they are asked to state if they would like to read another story by picking the option “Sì” (“Yes”) or “No”.
- “Hai qualche commento da fare?” (“Do you have any comments?”) Users can write anything in a simple text area, to be able to address any problem they might have had during the survey.

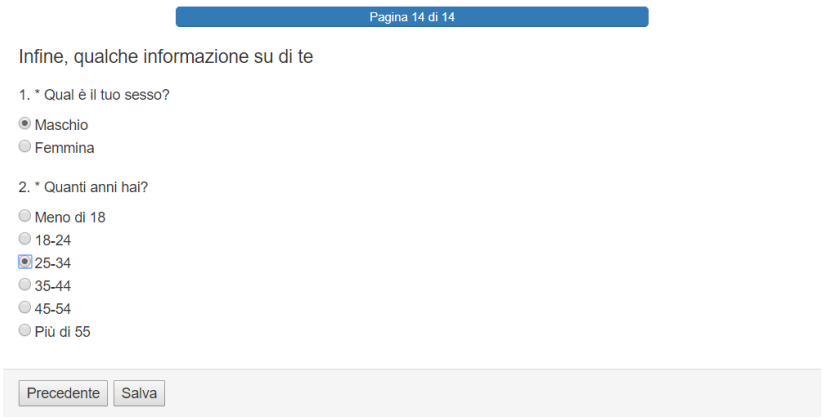


Figure 5.3: Demographics page of the survey.

Finally, users are asked about gender and age as in Fig. 5.3.

5.3 Survey results

The user survey run from February 4th, 2018 to March 2nd, 2018. A total of 97 users participated, with 29 males (29.90%) and 68 females (70.10%) and age distributed as seen in Figure 5.4. The sample size is not particularly large but there is a fair variety in both gender and ages.

We decide to use the unit ratings to assess how good the annotated emotions are for each unit. In fact each user gave an integer score from 1 (negative) to 5 (positive) for the polarity of the emotion and an integer score from 1 (low intensity) to 3 (high intensity) for the intensity of the emotion. We can compare this to the GEMEP emotion group to which it was assigned by the story designer.

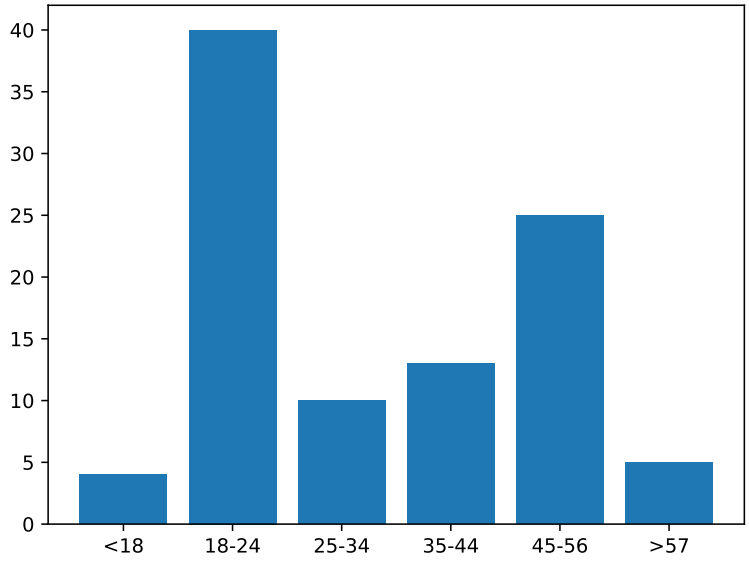


Figure 5.4: Age distribution of survey users.

As for the overall story rating, we decide to compute a quality score by making an average of the following values:

- *engagement score*, continuous value from 0 to 100 expressing how much users agreed to the statement "I felt engaged by the story.";
- *coherence score*, continuous value from 0 to 100 expressing how much users agreed to the statement "The story made sense.";
- *rating score*, continuous value from 0 to 100 expressing how much users agreed to the statement "I enjoyed the story.";
- *read-again score*, discrete value that can be either 0 ("No") or 100 ("Yes") expressing whether users would read another story or not.

It is possible to investigate if it would be more convenient to perform a weighted average of such values, but we feel that none of these has a major impact with respect to the others on the quality score and we decide to limit ourselves to a simple average, assuming all values to have the same weight.

Chapter 6

Applied method

After choosing a clustering method, a clustering distance and performing a well-designed user survey, we have fulfilled all the requisites necessary to apply our method. We now analyze the gathered data to get insights into the overall quality of the story of our choice, *Hot Bread*.

DUPIN (Detective UPon Interactive Stories), the software implementation of this method, is written in Python and its code is publicly available in a Git repository[1].

6.1 Annotated emotions consistency

As described in Section 5.2, in our survey users could give a value for the polarity and the intensity of the emotion felt while reading each unit. Ideally, this would coincide with the same emotion the unit is annotated with by the story designer. The reader may recall that in our IS system emotions are divided into four main groups based on the combination between two emotional dimensions: polarity (positive or negative) and intensity (high or low).

Each unit is annotated with a theoretical emotion, decided by the story designer. By assigning this emotion to one of the four emotion groups we get theoretical values for:

- polarity: $+1$ for positive and -1 for negative;
- intensity: 1 for low intensity and 2 for high intensity;

[1] https://github.com/msilvestro/dupin

that together form a pair. The polarity is defined over the interval $[-1, +1]$, while the intensity is defined over the interval $[0, 2]$. Note that we also allow the pair $(0, 0)$, useful to denote a unit with a neutral emotion or without annotations. For instance, unit 001 is annotated with the emotion despair, that belongs to the group "negative, high intensity" and hence has as theoretical value pair $(-1, 2)$.

From the survey we have multiple values for each unit, since different stories can share the same units. For each user and each unit we have five possible discrete values for the polarity (0, 1, 2, 3, 4) and three possible discrete values for the intensity (0, 1, 2). Then we compute the average of those values for each unit and normalize them so that they fit into the corresponding interval, i.e. $[-1, +1]$ for the polarity and $[0, 2]$ for the intensity. For instance, unit 001 has an average polarity of 1.39 and an average intensity of 1.13, that normalized become the pair $(-0.305, 1.13)$.

At this point we can use a distance to compute how much the survey values differ from the theoretical ones and obtain the *emotion inconsistency score*, as we call it. We use a simple custom distance, that is

$$||(p_1, i_1) - (p_1, i_2)||_c = 2 \cdot |p_1 - p_2| + |i_1 - i_2|,$$

because we are under the impression that the polarity should have a bigger impact than intensity. We assume that assigning the opposite polarity with respect to the one felt by the user is a bigger mistake than assigning a lower or higher intensity. Hence, we use the absolute value of the difference between each coordinate following the same simple approach of Manhattan distance, but with polarity having twice the importance of intensity.

Since we want the information to be more immediate to read, we transform this value into a percentage. We note that the custom distance has a minimum in 0, when two pairs are exactly the same, and a maximum in 6, when two pairs have both opposite polarity and opposite intensity. This implies that is suffices to divide distances by 6 to obtain the percentage relative to the maximum possible distance.

As an example, we can compute the inconsistency score for unit 001 in the following way:

$$\frac{||(-1, 2) - (-0.305, 1.13)||_c}{6} = \frac{2 \cdot |-0.695| + |0.87|}{6} = \frac{2.26}{6} = 0.37\overline{6},$$

so we can say that the user perceived emotion differs from the one chosen by the story designer of about 37.67%. It is possible to select a threshold

so that all values below it could be regarded as acceptable, while the others may require some inspection. We want the threshold to be high enough to allow a reasonable margin of error but low enough to capture badly annotated units. We arbitrarily choose 40% as threshold.

A complete table of resulting values can be seen in Table 6.1. For each unit we have: annotated emotion, theoretical polarity, theoretical intensity, survey polarity, survey intensity and inconsistency score, highlighted if it exceed the threshold of 40%.

Unit	Emotion	T. pol.	T. int.	S. pol.	S. int.	Score (%)
000		0	0	0.490	0.800	29.67
001	despair	−1	2	−0.305	1.130	37.67
002	sadness	−1	1	−0.450	1.100	20.00
003	joy	1	2	0.405	1.000	36.50
004	relief	1	1	0.350	0.850	24.17
005	sadness	−1	1	−0.125	1.080	30.50
006	pride	1	2	0.250	1.000	**41.67**
007	relief	1	1	0.310	0.880	25.00
008	joy	1	2	0.190	1.380	37.33
009	relief	1	1	−0.355	0.930	**46.33**
010	despair	−1	2	−0.190	0.880	**45.67**
011	sadness	−1	1	−0.335	0.830	25.00
012	hot anger	−1	2	−0.460	0.830	37.50
013	amusement	1	2	−0.175	1.060	**54.83**
014	relief	1	1			
015	irritation	−1	1	−0.500	1.500	25.00
016	relief	1	1	−0.625	1.500	**62.50**
017	joy	1	2	−0.190	1.500	**48.00**
018	anxiety	−1	1	−0.100	0.600	36.67
019	joy	1	2	0.800	1.000	23.33
020	hot anger	−1	2	−0.470	1.180	31.33
021	relief	1	1	0.100	1.200	33.33
022	pleasure	1	1	−0.325	0.880	**46.17**
023	pride	1	2	0.060	1.120	**46.00**
024	relief	1	1	−0.310	0.750	**47.83**
025	hot anger	−1	2	−0.440	1.380	29.00
026	hot anger	−1	2			
027	joy	1	2			
028	relief	1	1	−0.625	1.500	**62.50**

Unit	Emotion	T. pol.	T. int.	S. pol.	S. int.	Score (%)
029	despair	−1	2	−0.375	1.000	37.50
030	sadness	−1	1	0.000	1.500	**41.67**
031	joy	1	2	0.500	1.000	33.33
032	pleasure	1	1	0.335	1.000	22.17
033	hot anger	−1	2	−0.945	1.560	9.17
034	sadness	−1	1	−0.395	0.930	21.33
035	joy	1	2	0.375	1.080	36.17
036	irritation	−1	1	0.400	1.500	**55.00**
037	hot anger	−1	2	−0.390	1.000	37.00
038	sadness	−1	1	−0.555	1.220	18.50
039	joy	1	2			
040	interest	1	1	−0.165	1.330	**44.33**
041	anxiety	−1	1	−0.560	1.500	23.00
042	despair	−1	2			
047	joy	1	2			
048	relief	1	1	−0.085	0.830	39.00
049	despair	−1	2	0.250	1.750	**45.83**
050	sadness	−1	1	0.200	1.400	**46.67**
051	joy	1	2			
052	pleasure	1	1	0.500	1.400	23.33
053	hot anger	−1	2	−0.630	1.320	23.67
054	sadness	−1	1	−0.535	0.930	16.67
055	pride	1	2	1.000	1.000	16.67
056	relief	1	1	−0.375	0.880	**47.83**
057	hot anger	−1	2			
058	sadness	−1	1	−0.500	1.580	26.33
059	joy	1	2			
060	relief	1	1	−0.285	1.140	**45.17**
061	sadness	−1	1	−0.415	1.170	22.33
062	irritation	−1	1	−0.750	1.500	16.67
063	sadness	−1	1	−0.250	1.500	33.33
064	despair	−1	2	1.000	2.000	**66.67**
065	joy	1	2			
066	relief	1	1	0.555	1.000	14.83
067	anxiety	−1	1	−0.500	1.610	26.83
068	despair	−1	2			
069	joy	1	2	−1.000	1.000	**83.33**

Unit	Emotion	T. pol.	T. int.	S. pol.	S. int.	Score (%)
070	relief	1	1	−0.650	1.100	**56.67**
071	sadness	−1	1	−0.400	1.000	20.00
072	despair	−1	2			
073	joy	1	2	0.000	1.000	**50.00**
074	relief	1	1	0.210	1.050	27.17
075	sadness	−1	1	−0.450	1.300	23.33
076	hot anger	−1	2			
077	joy	1	2			
078	pleasure	1	1	0.165	0.890	29.67
079	anxiety	−1	1	−0.435	1.470	26.67
080	panic fear	−1	2			
081	joy	1	2			
082	joy	1	2	0.375	0.850	40.00
083	sadness	−1	1	−0.305	1.170	26.00
084	despair	−1	2	−1.000	2.000	0.00
085	joy	1	2	0.000	1.000	**50.00**
086	relief	1	1	−0.290	0.840	**45.67**
087	pleasure	1	1	−0.835	1.330	**66.67**
088	joy	1	2	0.500	2.000	16.67
089	despair	−1	2			
090	anxiety	−1	1	0.555	1.220	**55.50**
091	anxiety	−1	1	−0.480	1.040	18.00
092	hot anger	−1	2	−1.000	2.000	0.00
093	joy	1	2	0.000	1.000	**50.00**
094	joy	1	2	0.000	1.000	**50.00**
095	sadness	−1	1	0.175	0.820	**42.17**
096	hot anger	−1	2	−0.500	2.000	16.67
097	joy	1	2	0.335	0.330	**50.00**
098	relief	1	1	0.690	1.250	14.50
099	irritation	−1	1	−0.380	0.710	25.50
100	despair	−1	2	−0.750	2.000	8.33
101	joy	1	2	0.500	1.220	29.67
102	interest	1	1	0.240	0.910	26.83
103	sadness	−1	1	0.190	0.850	**42.17**
104	despair	−1	2	−0.835	0.330	33.33
105	joy	1	2	0.415	1.000	36.17
106	relief	1	1	0.225	0.910	27.33

Unit	Emotion	T. pol.	T. int.	S. pol.	S. int.	Score (%)
107	irritation	−1	1	−0.250	1.440	32.33
108	hot anger	−1	2	−0.375	0.850	40.00
109	joy	1	2	0.335	1.170	36.00
110	pleasure	1	1	0.720	1.380	15.67

Table 6.1: Frequency table for story lengths of *Hot Bread* extracted linear stories.

Only 30 units out of 107 have a score higher than 40% and hence, if our intuition regarding the threshold level is right, we may be satisfied of the overall emotion consistency. These units can however still be inspected and the annotated emotions tuned accordingly.

6.2 Quality of stories

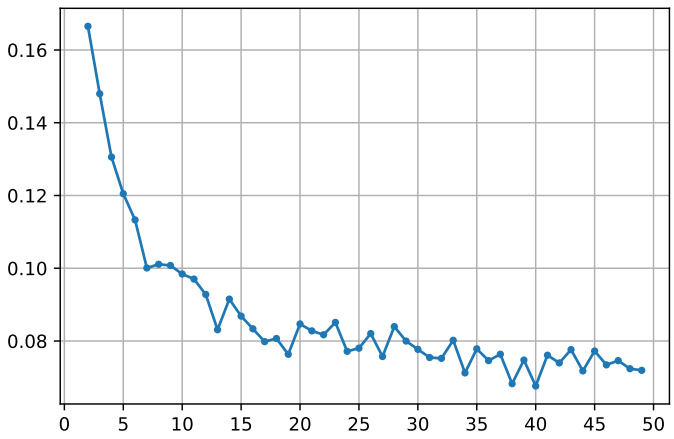
While manual inspection of each story provides useful information, it is a costly operation. As explained in Chapter 3 and Chapter 4 we try to overcome this problem by clustering stories using k-medoids with Manhattan distance on tension curves. If two stories are similar and their tension curves resemble each other they are more likely to be put in the same cluster. We use this fact to provide a better tool to investigate the set of all possible stories.

6.2.1 Number of clusters

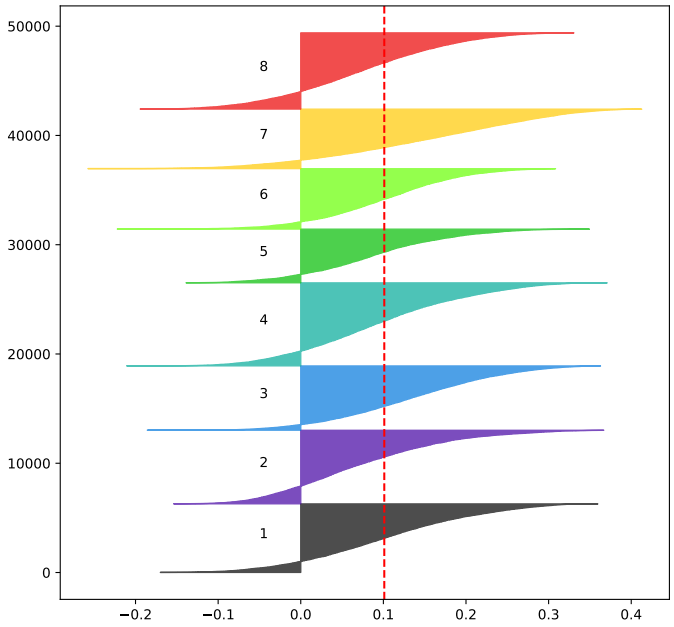
We start by deciding what is the number of clusters we are interested to obtain. Ideally we want each cluster to contain stories with tension curves that resembles a particular shape so to maximize cluster consistency. We already showed that a way to measure this property is the silhouette value.

We perform experiments with $k = 2, 3, \ldots, 50$ and, for each of them, we compute the average silhouette value. The plot in Figure 6.1a shows the average silhouette value for each number of clusters considered.

Ideally we would want to select the number of clusters that maximizes the silhouette value. However, we can see that it tends to decrease, hence if we were to choose the maximum we would end up with only two clusters. This behavior is quite strange and shows that there is no value for k that stands out and allows for an optimal clustering. We decide to choose $k = 8$, since it strikes, in our honest opinion, a good balance between having enough



(a) Average silhouette value for each number of clusters chosen.



(b) Silhouette plot for $k = 8$.

Figure 6.1: Analysis of average silhouette values and silhouette plot for $k = 8$.

clusters and not decreasing the silhouette value too much. In this point the silhouette score is 0.101, a value which is not particularly promising, and

its silhouette plot is shown in Figure 6.1b. We expect to have a reasonable clustering structure but not clear-cut clusters.

6.2.2 Clustering results

As a byproduct of k-medoids each cluster is associated with a medoid, whose tension curve is used as the basic shape of reference. Furthermore, thanks to the user survey we have quality scores for each story and, from that, we can compute a quality score for the whole cluster.

For each cluster we have the following information:

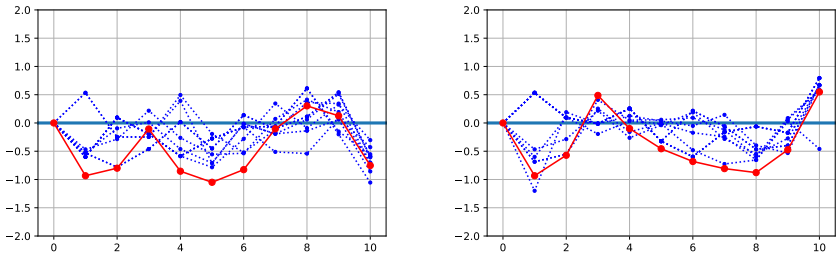
- number of stories contained (N);
- number of stories contained for which we have data from the survey (NS);
- average engagement score (ES);
- average coherence score (CS);
- average rating score (RS);
- average read-again score (RAS);
- average quality score, that is the average of the four aforementioned scores (QS);
- reference shape, that is the tension curve of the medoid.

In the best case scenario we shall expect clusters with strongly different reference shapes and with varying levels of quality scores.

Cluster	N	NS	ES	CS	RS	RAS	QS
1	6269	14	42.05	45.93	46.18	42.86	44.25
2	6727	13	59.96	53.12	51.62	69.23	58.49
3	5879	6	47.23	52.61	53.77	66.67	55.07
4	7593	12	39.01	51.22	44.34	66.67	50.31
5	4905	10	35.50	43.19	39.31	40.00	39.50
6	5522	14	55.31	56.82	58.12	50.00	55.06
7	5442	10	45.17	43.27	46.07	70.00	51.13
8	6971	17	59.10	57.57	60.69	52.94	57.57

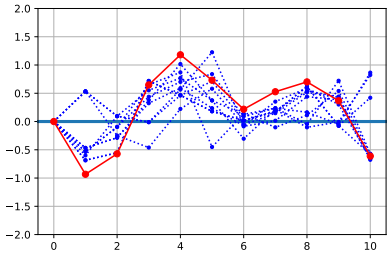
Table 6.2: Clustering results.

As we can see from Table 6.2, the quality of the clusters shows a large variability:

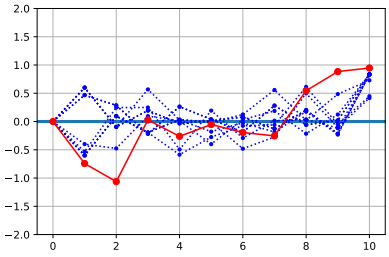


Cluster 1

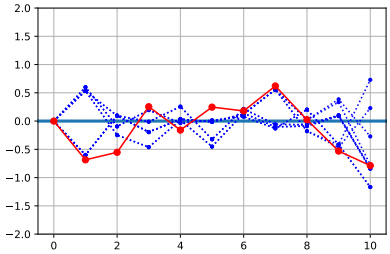
Cluster 2



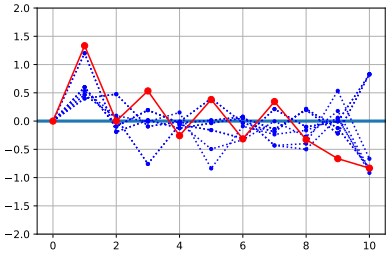
Cluster 3



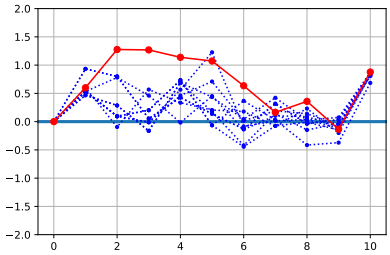
Cluster 4



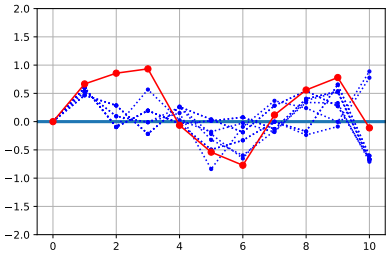
Cluster 5



Cluster 6



Cluster 7



Cluster 8

Figure 6.2: Reference shape and nearest stories tension curves for each cluster.

- cluster 2 and cluster 8 both have a high quality score, respectively 58.89 and 57.57, and hence the stories they contain are arguably good ones;
- cluster 1 and cluster 5, on the other hand, have very low quality scores, respectively 44.25 and 39.50, and hence the stories they contain shall be reviewed to see whether or not they are amenable of improvement.

In order to check these things out, only a few stories per cluster need be inspected directly, since the clustering makes sure that all of them have a similar tension curves.

In Figure 6.2 we have for each cluster the reference shape and the 10 nearest stories in that cluster. The reference shape is displayed in red, while the nearest stories are shown as dotted blue lines. We can see that there is actually quite some difference even between the nearest tension curves and the reference shape, that is probably a reason for the low silhouette values obtained earlier.

Despite the disputable quality of clusters, still some correlation is apparent between the clusters and the quality of stories. In fact, every shape might be analyzed from a dramatic perspective. For example, the shape of cluster 8 is really interesting, since it follows the "Cinderella" (rise-fall-rise) emotional arc [38], and has a high quality score associated.

On the other side, cluster 1 seems unbalanced, in the sense that most of the story has a negative trend that may hinder the overall enjoyability of the story. Cluster 5 instead has an almost flat shape, so the absence of turning points might make the story boring. Not surprisingly, the quality scores of these two clusters are the lowest.

In an applicative context this would be helpul information. The story designer would need to check if these results relate to the actual state of the interactive story: for each cluster a few stories would be inspected directly. Finally, the story designer should understand how stories in the same cluster relate to its shape and quality score. Thanks to this new information it is easier to identify issues or notice good features in stories. Most of all, the quality score should reflect the appreciation of the final users, i.e. the actual readers at which the IS system is targeted.

Chapter 7

Conclusions

Starting from the work of Bída et al. [6] we obtained a new methodology tailored at story-driven IS systems. By means of a tension evaluation function, that associates every unit of the IS system with a value corresponding to the tension in that point of the story, we can get a big data set (for each story extracted from the IS system) on which we can perform a clustering. The choice of a suitable tension evaluation function is a crucial part of this methodology and is specific for each IS system. By doing so, we have a reference shape for each cluster, that is a specific tension curve that expresses the overall dramatic trend of all stories in that specific cluster.

Furthermore, by means of a user survey, in which users are presented with a random story extracted from the IS system, we can gather information about how much a story is considered engaging or enjoyable. Combining this with the clustering results we can obtain a quality score for each cluster, that assesses whether it contains good or bad stories.

Finally, the story designer is provided with a list of clusters, each with a specific reference shape and a quality score. With this information a few stories may be inspected directly, to understand what makes them good or not. It is also possible to try to find a relation between tension curve shapes and quality scores directly, for example noting that a particular shape had an unexpected success. This shape can be used as an outline for composition of further stories.

In our case, while the results were not impressive, we were able to get useful insights into the inner workings of our IS system and we also identified a successful reference shape. It would be interesting to apply this methodology to other IS systems, maybe more complex ones, and try different tension evaluation functions and clustering algorithms. We were not

able to collect a large sample from the user survey, that is an important step to obtain good data to compute the quality score.

If this methodology proves to be useful for story designers, a software may be created to let them upload their own IS system and let it be automatically processed, choosing beforehand all needed parameters. Ideally, there would be an online procedure to obtain the tension curve of a new story composed by the story designer, that could help to visualize immediately its dramatic trend and eventually compare it to ones that were proven successful in previous analyses.

Bibliography

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. "On the Surprising Behavior of Distance Metrics in High Dimensional Space". In: *Database Theory — ICDT 2001*. Ed. by Jan Van den Bussche and Victor Vianu. Springer Berlin Heidelberg, 2001, pp. 420–434.

[2] Tim Anderson et al. *Zork*. 1980.

[3] R. S. Aylett et al. "FearNot! – An Experiment in Emergent Narrative". In: *Intelligent Virtual Agents: 5th International Working Conference*. Ed. by Themis Panayiotopoulos et al. Springer Berlin Heidelberg, 2005, pp. 305–316.

[4] BioWare. *Neverwinter Nights*. 2002.

[5] Tanja Bänziger, Marcello Mortillaro, and Klaus Scherer. "Introducing the Geneva Multimodal Expression Corpus for Experimental Research on Emotion Perception". In: *Emotion* 12.5 (Nov. 2011), 1161—1179.

[6] Michal Bída, Martin Černý, and Cyril Brom. "Towards Automatic Story Clustering for Interactive Narrative Authoring". In: *Interactive Storytelling: 6th International Conference*. Springer International Publishing, 2013, pp. 95–106.

[7] Italo Calvino. *If On A Winter's Night A Traveller*. 1979.

[8] Marc Cavazza et al. "Madame Bovary on the Holodeck: Immersive Interactive Storytelling". In: *Proceedings of the 15th ACM International Conference on Multimedia*. MM '07. Augsburg, Germany: ACM, 2007, pp. 651–660. ISBN: 978-1-59593-702-5.

[9] Stephen Cave. *The 4 stories we tell ourselves about death*. July 2013. URL: https://www.ted.com/talks/stephen_cave_the_4_stories_we_tell_ourselves_about_death.

[10] Julia Chaitin. *Narratives and Story-Telling*. July 2003. URL: http://www.beyondintractability.org/essay/narratives.

[11] Chunsoft. *Nine Hours, Nine Persons, Nine Doors*. 2009.

[12] Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo.
"DoppioGioco. Playing with the Audience in an Interactive Storytelling Platform".
In: *Complex, Intelligent, and Software Intensive Systems: Proceedings of the 11th International Conference on Complex, Intelligent, and Software Intensive Systems*.
Ed. by Leonard Barolli and Olivier Terzo. Springer International Publishing, 2018, pp. 287–298.

[13] Peter Sheridan Dodds. *Homo Narrativus and the Trouble with Fame*. Sept. 5, 2013.
URL: http://nautil.us/issue/5/fame/homo-narrativus-and-the-trouble-with-fame.

[14] Steven Dow et al.
"Initial Lessons from AR Façade, an Interactive Augmented Reality Drama".
In: *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. ACE '06.
Hollywood, California, USA: ACM, 2006. ISBN: 1-59593-380-8.

[15] Paul Ekman. "Basic Emotions". In: *Handbook of Cognition and Emotion*.
Ed. by Tim Dalgleish and Michael J. Power.
Hollywood, California, USA: Wiley, Feb. 25, 1999, pp. 301–320.

[16] Gustave Flaubert. *Madame Bovary*. 1857.

[17] E. Forgy. "Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification". In: *Biometrics* 21.3 (1965), pp. 768–769.

[18] Gustav Freytag.
Technique of the Drama: An Exposition of Dramatic Composition and Art. 1863.

[19] Telltale Games. *The Walking Dead*. 2012.

[20] Telltale Games. *The Walking Dead: Season Two*. 2013–2014.

[21] Greg Hamerly and Charles Elkan.
"Alternatives to the K-means Algorithm That Find Better Clusterings".
In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*. CIKM '02. ACM, 2002, pp. 600–607. ISBN: 1-58113-492-4.

[22] Grady Hendrix. *Choose Your Own Adventure: How The Cave of Time taught us to love interactive entertainment*. Feb. 18, 2011.
URL: http://www.slate.com/articles/arts/culturebox/2011/02/choose_your_own_adventure.single.html.

[23] Matthew A. Jaro. "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida".
In: *Journal of the American Statistical Association* 84.406 (1989), pp. 414–420.

[24] Demian Katz. "The Early History of Gamebooks: Some New Discoveries".
In: *Fighting Fantazine* 15 (May 31, 2016), p. 64.
URL: http://www.fightingfantazine.co.uk/page/.

[25] L. Kaufman and Peter Rousseeuw. "Least squares quantization in PCM".
In: *Statistical Data Analysis Based on the L1 Norm and Related Methods*.
North-Holland, 1987, pp. 405–416.

[26] Key. *Clannad*. Apr. 28, 2004.

[27] Chris Klimas. *Twine*. 2009. URL: http://twinery.org/.

[28] Robin D. Laws. *Robin's Laws of Good Gamemastering*. 2002.

[29] Dave Lebling and Marc Blank. *Zork I: The Great Underground Empire*. Infocom, 1984. URL: http://infodoc.plover.net/manuals/zork1.pdf.

[30] Dan Lekic. *Character-Driven Vs. Plot Driven: Which is Best*. Feb. 2017. URL: https://nybookeditors.com/2017/02/character-driven-vs-plot-driven-best/.

[31] Vladimir Iosifovich Levenshtein. "Binary codes capable of correcting deletions, insertions and reversals." In: *Soviet Physics Doklady* 10.8 (1966), pp. 707–710.

[32] Stuart P. Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137.

[33] James B. MacQueen. "Some Methods for Classification and Analysis of MultiVariate Observations". In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by L. M. Le Cam and J. Neyman. Vol. 1. University of California Press, 1967, pp. 281–297.

[34] Michael Mateas and Andrew Stern. "Façade: An Experiment in Building a Fully-Realized Interactive Drama". In: *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment* (Apr. 2003).

[35] Andrew Ortony, Gerald Clore, and Allan Collins. "The Cognitive Structure of Emotion". In: *Contemporary Sociology*. Vol. 18. Jan. 1988.

[36] Interplay Productions. *Fallout: A Post Nuclear Role Playing Game*. 1997.

[37] Keith Quesenberry and Michael Coolsen. "What Makes a Super Bowl Ad Super? Five-Act Dramatic Form Affects Consumer Super Bowl Advertising Ratings". In: *The Journal of Marketing Theory and Practice* 22 (Oct. 2014), pp. 437–454.

[38] Andrew J. Reagan et al. "The emotional arcs of stories are dominated by six basic shapes". In: *EPJ Data Science* 5.1 (Nov. 4, 2016), p. 31.

[39] D. L. Robinson. "Brain function, mental experience and personality". In: *The Netherlands Journal of Psychology* (2009), pp. 152–167.

[40] J. K. Rowling. *Harry Potter and the Deathly Hallows*. July 21, 2007.

[41] T.L. Saaty and M.S. Ozdemir. "Why the magic number seven plus or minus two". In: *Mathematical and Computer Modelling* 38.3 (2003), pp. 233–244.

[42] Henrik Schoenau-Fog. "Hooked! – Evaluating Engagement as Continuation Desire in Interactive Narratives". In: *Interactive Storytelling: Fourth International Conference on Interactive Digital Storytelling*. Springer Berlin Heidelberg, 2011, pp. 219–230.

[43] Henrik Schoenau-Fog. "Hooked! – Evaluating Engagement as Continuation Desire in Interactive Narratives". In: 7069 (Nov. 2011), pp. 219–230.

[44] Forge Reply srl. *Joe Dever's Lone Wolf HD Remastered*. Ed. by 505 Games. Nov. 18, 2014.

[45] David Thue et al. "Interactive Storytelling: A Player Modelling Approach." In: *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference* (Jan. 2007), pp. 43–48.

[46] Kurt Vonnegut. *Shapes of Stories*. Oct. 2010. URL: https://www.youtube.com/watch?v=oP3c1h8v2ZQ.

[47] William E. Winkler. "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage". In: *Proceedings of the Section on Survey Research*. Wachington, DC, 1990, pp. 354–359.

[48] Georgios N. Yannakakis and John Hallam. "Ranking vs. Preference: A Comparative Study of Self-reporting". In: *Affective Computing and Intelligent Interaction*. Springer Berlin Heidelberg, 2011, pp. 437–446.

[49] Jim Yu. *Go Beyond Advertising and into Storytelling*. Jan. 29, 2014. URL: https://www.huffingtonpost.com/jim-yu/go-beyond-advertising-and_b_4683818.htm.