Fantastic Strings and Where to Find Them:
The Quest for High-Quality Video Game Text Corpora

Judith van Stegeren and Mariët Theune
Human Media Interaction
University of Twente
Enschede, The Netherlands
j.e.vanstegeren@utwente.nl, m.theune@utwente.nl

Abstract

High-quality video game text corpora can be used as resources for many types of research, including but not limited to text generation for games. However, these corpora are scarce. We address this issue by proposing a number of quality criteria for video game text corpora, and describing from where such corpora can be obtained. We also present three datasets with game texts from popular video games *Torchlight II*, *Star Wars: Knights of the Old Republic* and *The Elder Scrolls*, together with examples of how these corpora can be used in research.

Introduction

Procedural content generation (PCG) for video games deals with the automatic creation of video game assets, such as textures, music and levels. We can also generate in-game text, such as dialogue and quests, by using techniques from natural language generation (NLG). NLG is a part of natural language processing, the research field that combines linguistics, computer science and AI. NLG has seen serious developments in the past years, especially due to machine learning and architectures built on neural networks, e.g. word2vec (Mikolov et al. 2013), BERT (Devlin et al. 2019), and GPT-2 (Radford et al. 2019).

Most text generators for video games still use techniques from more than fifty years ago, such as pattern-matching and string replacement. Newer methods often need large amounts of data for training, but text corpora sourced from video games are scarce. Additionally, rules and templates for text generators in academia are mostly written by amateur writers, which means that the output of these systems is not representative for the output of professional game writers.

High-quality video game text corpora can be used as resources for many types of research, including but not limited to text generation for games. Moreover, if multiple researchers use the same corpus this facilitates comparison of their results and can lead to research advancements, for example via the organisation of shared tasks.

In this paper, we address the scarcity of video game text corpora and make the following contributions:

- We propose a number of quality criteria for video game text corpora.
- We describe from where such corpora can be obtained.
- We present three datasets with game texts from popular video games, together with examples of how these corpora can be used in research.

With this, we hope to raise awareness of the importance of high-quality game text corpora as a resource for AI for digital entertainment, and to encourage researchers to collect and, if possible, share their own corpora, as a step towards shared progress in the field.

Related work

Corpora in games and AI

Inspired by the NLP field, where sharing corpora for shared tasks is the norm, games and AI researchers have started to share datasets to bring the research field forward as well. Although they are termed corpora, a term normally reserved for collections of text, most of these datasets do not contain game text, but other types of game assets. Summerville et al. (2016) collected a corpus of video game levels in three annotation formats, which can be used for procedural content generation and level design analysis. Various datasets with gameplay data have been released, for analysing player strategies and training AI-systems that can play games autonomously. For example, Lin et al. (2017) created a dataset of *StarCraft* replays that can be used for learning gameplay models. They also reviewed existing StarCraft datasets, and proposed a list of quality criteria for gameplay datasets for machine learning. Despite the differences in content, this research can be used as source of inspiration for creators of text corpora: what makes these datasets usable, how are they used by researchers after release, and what is ‘quality’ for datasets in games and AI?

Text generation for video games

Most research on generating textual game content uses either template-based techniques (Caropreso et al. 2012; Lukin, Ryan, and Walker 2014), rewriting techniques based on grammars (Ryan et al. 2016; Grinblat and Bucklew 2017; Hall, Williams, and Headleand 2017; Green et al. 2018) or graphs (Kybartas and Verbrugge 2014). The use of newer NLP techniques, which build on machine learning architectures, must largely still be explored in the context of video games. A notable exception is the work of Walker et al. (2011), who used used statistical machine learning to create language models of character linguistic style. The language models were used to generate dialogue with personality for *SpyFeet*, a prototype roleplaying game. The authors used film dialogues as the basis for the linguistic models. A game that does leverage state-of-the-art NLG techniques is text adventure game *AI Dungeon* (Walton 2019). This game uses OpenAI's GPT-2 language model (Radford et al. 2019) to generate personalized text adventures. The game's generator was trained on a corpus from the website ChooseYourAdventure.com, a community around choose-your-own-adventure style story games.

A limitation of current research is that the output of generator systems is often not based on material written by professional video game writers. For the purpose of academic research, most researchers create their own games, templates or grammars, or ask research participants to create game texts in crowdsource experiments (Orkin and Roy 2007). An exception is when game developers publish in academic venues about their own text generators, such as Grinblat and Bucklew (2017).

Text analysis for video games

We cannot separate *text generation* from *text analysis*, as text analysis can inform generative systems before and after generation. Before generation, a generator can use text analysis techniques to model the structure of its output by codifying patterns observed in examples. After generation, text analysis can be used to evaluate properties of the generated artifacts.

There is also an increasing interest in generators that can respond to their input and create context-sensitive outputs. Kreminski, Karth, and Wardrip-Fruin (2019) argue that describing how a procedural generator reads and interprets its input is as important as describing how the generator generates its output. This is especially true for procedural generation that is based on machine learning (Summerville et al. 2018), where the input (i.e. training data) is a determining factor in the generation process.

Landwehr, Diesner, and Carley (2009) scraped a corpus of World of Warcraft quests from quest repository Allakhazam,[1] and used this to analyse the cultural and narrative elements embedded in quest text. Kybartas and Verbrugge (2014) presented an approach for quest generation by using graph rewriting techniques. Their quest generator, called ReGEN, can generate new quests based on changes in game state, making player choices more meaningful. To validate their approach in a quantitative way, the authors also proposed a metric for the quality of a (game) narrative. They used this metric to measure the performance of their quest generator compared to the quests of *The Witcher* and *Elder Scrolls V: Skyrim*. The quest data for the two games was collected from game wikis. Schlünder and Klabunde (2013) analysed greetings in NPC dialogue transcriptions of *Skyrim*, and proposed an algorithm for more context-sensitive greeting generation.

Related text corpora

Video game text constitutes many different types of text. Depending on properties like genre and gameplay, a game might consist of dialogue, narratives, quests, and flavor text. By flavor text, we mean game text that has a cosmetic purpose as opposed to a functional one. Text processing for the video games domain can profit from NLP research that studies the types of text that we also encounter in games, such as dialogue and stories. Text corpora for these types of text are much more common, e.g. the CMU movie summary corpus (Bamman, O'Connor, and Smith 2013) and the ROCStories corpus (Mostafazadeh et al. 2016) for stories, and the switchboard corpus (Godfrey, Holliman, and McDaniel 1992) for dialogues. However, the usefulness of these corpora for the video games domain is limited, as results on these corpora might not be transferable to video games. For example, story corpora might contain stories that consist of a few sentences, which is not comparable to the interactive and complex narratives found in video games.

Quality of video game corpora

In order to benefit from the recent developments in NLP and AI, we need high-quality datasets of video game texts, both for training and evaluation. Recent neural architectures, such as GPT-2 and BERT, can be fine-tuned on small, domain-specific datasets to increase their performance for specific domains or tasks. Video game text corpora can be used for fine-tuning these systems specifically to video game texts, which is likely to increase the effectivity of NLP techniques for the games domain. Additionally, text corpora with ground truth data can be used for evaluating new techniques and systems.

Below, we propose quality criteria for video game text corpora, based on our experience in researching NLP for video games. This list is a first attempt to create an overview of desirable properties for new datasets, similarly to the list provided by Lin et al. (2017).

Richness Datasets should contain both game text and information about their in-game context.

Representativeness Strings in the dataset should be written by professional video game writers. Strings should preferably be sourced from popular or well-known (commercial) games that have a substantial user base.

Diversity Datasets should reflect the diversity of the video games domain.

Portability Datasets should be shared in a portable plain-text format that does not require special tools to read or modify.

Researchers might be able to find 'text dumps' of popu-

[1] https://wow.allakhazam.com/
Allakhazam's World of Warcraft quest database was discontinued in 2013.

lar games[2] online, which consist of strings from the game without any context. However, because game texts are governed by the underlying game logic, game texts are inherently context-sensitive. If we try to analyse a game text in isolation, we cannot interpret it correctly. Consequently, game text corpora should provide *rich* information about the context of each text. For example, for dialogue lines, we need information about conversation participants. Which NPC is saying what, to whom, and why? What is their relation to the player character? Is a particular dialogue line part of a larger narrative (such as the main storyline) or a story of minor importance (a side quest, an NPC backstory, flavor text)? Are there specific conditions in which the text is shown, or explicitly hidden from the player? Is there a specific order in which text is presented, or is the player free to choose?

Another challenge is that corpora need labels or some other kind of ground truth before we can use them for supervised machine learning and evaluation. Although in most cases game texts do not have labels in the strict sense of the word, we can use properties from the in-game context as ground truth. We will discuss below how this applies to the datasets presented in this paper.

Some research uses corpora of video game text that are not *representative* of the video games domain, such as text sourced from research games, text written by academics, or text crowdsourced from research participants. Ideally, video game corpora consist of text written by (professional) game writers, sourced from real-world video games. Here, we mean real-world games as opposed to prototype games or research games, which are also prevalent in research but are generally shared with and played by a limited audience.

Diverse corpora are needed to reflect the diversity in games. There are many different types of in-game texts: NPC dialogue, item descriptions, in-game lore, puzzles and riddles, narration, flavor text, names, quests, tutorials and text from graphical user interfaces. If research is limited to only one type of game text, it does not do justice to the diversity of video games. Similarly, we need corpora that span the diversity in game genres, narrative genres and game developer backgrounds. Diversity is in the interest of the research field, as text processing methods might not transfer across game genres, narrative genres, storytelling methods, settings, writing styles and other aspects of game writing.

Finally, to ensure *portability*, corpora should be shared in a plain-text data format that is supported on a variety of platforms, such as CSV or JSON.

Obtaining new video game corpora

In this section we discuss methods for obtaining data that can be used as a source for new video game corpora: extracting text from game files, and scraping text from fan-websites.

Extracting text from game files

The highest quality data can be obtained directly from game files, as these contain the actual text that players will see during the game. We discuss three different approaches for this: extracting data from files of open-source games, using modding software provided by the publisher or developer, and using tools provided by online modding communities. Since we want to collect datasets that fulfill the representativeness property discussed above, we focus on real-world games.

Extracting text from *open-source games* can be an accessible approach to obtaining game texts from real-world games. It is in the interest of the open source community to make the inner working of the game, such as the working of the game engine and the structure of game assets, as understandable and usable as possible. Consequently, files are often stored in open and human-readable formats, the structure of game files and the working of the game engine is often documented and published, and game files require no proprietary or unpublished tools for inspection or modification. This is an advantage if we want to extract data from them for analysis.

Open-source games exist in a variety of types. They can be original games that were made available as open source from the start, such as *Endless Sky*, or open-source clones of closed-source games, such as *openRA*, an open-source clone of *Command & Conquer: Red Alert*. Some open-source clones are shipped with assets from the original game; others require the original game disks. Besides open-source games, there are also efforts to create open-source *game engines*, such as xoreos,[3] a project to opensource Bioware's Aurora game engine. An open-source game engine, and the accompanying tools, can help us extract game assets from commercial games.

However, most games are not open source. Games files of commercial games might be compressed, to save space and provide fast access for the game engine, or even encrypted, to prevent tampering and theft. In that case, we can use modding (modification) software to access the files. As modding tools are created with modification in mind, it depends on the tool whether it is possible to export (textual) game assets in bulk.

It is becoming more common for game publishers to release official modding software after the release of the game. Examples of games that come with their own modding toolkit are *Torchlight II* (GUTS), *Morrowind* (TES Construction Kit) and *Skyrim* (Creation Kit). The game's publisher or game development studio has an interest in the success of the official tools, as an active modding community can improve the life expectancy of a newly released game (Lee et al. 2020).

Official modding tools are often based on the developer's in-house tools. Consequently, they tend to be more robust than their community-provided counterparts discussed below. Their biggest advantage is that they often integrate well with the game engine and game files. Sometimes the publisher also provides extras that increase the usability of the tools, such as documentation and tutorials.

If the publisher has not released any tooling for modifying the game, or the official tooling is found to be too restrictive, the player community often starts making their own tooling. Community-provided tools are shared online via modding community websites (such as NexusMods), gaming forums, and gaming platforms (such as Steam Workshop). Tools vary

[2] Such as this text dump with dialogue from role-playing game Disco Elysium (ZA/UM 2019): https://gist.github.com/jd7h/e724eb2b23faa42b51424ac110c7b976

[3] https://xoreos.org/

from simple scripts to professionally developed software with a GUI and documentation.

However, not every game has an active modding community. Secondly, there is no guarantee that a community-provided tool will actually function correctly. Code may be untested, undocumented, or incompatible with newer computer systems. Finally, community-provided modding tools might require a high level of technical proficiency of the user.

Extracting game text from fan websites

Fan culture can give rise to extensive fan-made websites and wikis, where players collect information about the game, discuss strategies and share fanart. Often these fan-made websites are a great resource for texts (and other media) from the game. Kybartas and Verbrugge (2014) used the fan wikis of *The Witcher* and *Skyrim* to obtain information about game quests. Bergsma, van Stegeren, and Theune (2020) used in-game lore books and NPC dialogue sourced from *The Elder Scrolls* fan websites for their sentiment analysis research. The main advantage of collecting data from fan websites is that the text is already available in plain text, as opposed to text in game files, which is often compressed, encrypted or stored in a proprietary format. A possible drawback is that data from fan websites generally needs considerable data cleaning before it is of comparable quality to data extracted from the games themselves. Since fan wikis are often crowd-sourced, we cannot be sure of the accuracy of the text we find there. Information might be spread over various pages, structured in a heterogeneous format or missing. Similarly to other crowd-sourced internet resources such as Wikipedia, we might find text with errors ranging from spelling mistakes to untrue information. Another drawback of extracting game text from fan websites is that the texts might be presented without information about in-game context, which is contrary to our richness requirement.

Datasets

We used the techniques mentioned in the previous section to create three datasets with game text. The texts were sourced from popular commercial games: *Torchlight II* (Runic Games 2012), *Star Wars: Knights of the Old Republic* (BioWare 2003) and games from *The Elder Scrolls* video game series. The three datasets contain a broad range of game texts: linear NPC dialogue, branching NPC dialogue, quest objectives, GUI text, and flavor text. We briefly discuss the method for collecting each dataset, the contents and possible applications. Our methods for extracting text data from game assets do not generalise to other games, which is why we have not included a detailed technical description of our data collection methods in this paper. However, we will provide detailed descriptions of our extraction methods with the released datasets.

Dataset: *Torchlight II* quests

An example of a game that comes with modding software provided by the publisher is *Torchlight II*. *Torchlight II* is an action role-playing game that takes place in a fantasy world. The game consists of a main story that revolves around the destructive and corrupted Alchemist, and a collection of randomly generated dungeons that the player can explore as side-quests. The publisher, Runic Games, has published their in-house development kit "GUTS", together with a set of tutorials to teach players how they can change parts of the game and write their own extensions.

Torchlight's game assets are stored as XML-like UTF-16-encoded plaintext files, which are compressed and stored in PAK archives. We used GUTS to unpack Torchlight's game files from its main PAK archive. We then created a Python script to parse the XML files, extract the game text, and turn this into a ready-to-use dataset with quest texts and associated NPC dialogue. For accessibility reasons, we have created two datasets: a 'flattened' two-dimensional CSV, and a JSON-dataset that resembles the structure of the original game files. Both formats are highly portable, as they consist of plaintext data that is compatible with all kinds of tools and libraries.

In order to create the dataset, we combined data from two types of game assets: quest files and unit files. *Quest files* describe events, story components and dialogue. Quests are used to control the flow of the game narrative. They make up the main storyline and a set of side quests that revolve around procedurally-generated dungeons. *Unit files* describe interactive in-game objects, such as NPCs, items and doors. We used the unit files to translate the NPC identifiers found in quest dialogue data to human-readable NPC names.

Dataset contents The dataset of *Torchlight II* quests consists of 184 quests, out of which 131 quests contain text. The quests that have no text are used for controlling in-game objects, such as doors and checkpoints.

A quest can contain many different types of texts, such as NPC dialogue, flavor text, back story and GUI text. Most of the texts are dialogue lines. Whether a particular dialogue line is shown in-game depends on the player's progress for that respective quest. Quests might also contain flavor text. For an overview of the different text types, see Figure 1. Figure 2 shows three lines of dialogue from one of the side-quests in the game. The amount of dialogue contained in each quest varies. Simple side-quests contain only a few lines of dialogue for one NPC, i.e. client or the quest-giver that acts as the start and completion point of a quest. Larger quests may contain dialogue lines for multiple NPCs. The *Torchlight II* dataset consists of about 1000 datapoints, of which approximately 70% is NPC dialogue. 27 datapoints contain a long-form story synopsis that summarizes part of the main quest. The remaining datapoints are GUI text, which describe quest objectives in one of two lines.

Applications The *Torchlight II* quest dataset contains text type annotations, which can be used to filter specific types of text by in-game purpose. For example, since we can distinguish between quests from the main quest line and side quests, we can use this dataset to study the differences between these quest types. If we are researching flavor text, we can look at quest objects that contain 'passive dialogue'. Another example is quest objectives. We can use the list of quest objectives as ground truth for summaries of quest introduction dialogue. This data combination can be used to evaluate summarization techniques from the NLP field, to see how well they perform in a video games context.

Dialogue type	Description
intro	Dialogue text of the NPC that introduces the quest to the player.
return	Dialogue that the NPC speaks when the player returns to the quest-giver before completion of the quest.
details	The goal or objectives of the quest, as shown upon quest acceptance.
huddetails	A list of quest objectives. This list is shown in the game UI when the quest is active.
more details	Extra backstory for quests from the main questline.
complete	Dialogue for when the player returns to the quest-giver NPC after successfully completing the quest objective. After this text, the player receives a reward for completion of the quest, or is shown a new intro text to start a follow-up quest.
passive	Stand-alone dialogue lines that act as flavor text.

Figure 1: Text types in *Torchlight II* quest data, and their purpose in the game.

Dialogue type	Example text
intro	Hello! I thought I heard a human moving around out there. Listen, my name's Medrus. I got ambushed by some Sturmbeorn, and managed to get clear ... but I got pretty badly injured in the process. I can treat it, but I need some Merryweather Leaves. They grow around here, but I'm too weak to look for them. Think you can find some for me, bring 'em back here? You'll be rewarded, I promise.
return	Any luck finding the Merryweather Leaves? I'm not sure how much longer I can hold on ...
complete	You found some! Oh, thank the gods. A few moments' work, and . . . yes, there it is: a healing poultice. Now it will just take a little rest, and I'll be good as new. As it turns out, you brought back more leaves than I needed. So, here: a Healing Poultice for you, as a reward. Should you be badly injured, it'll set you right in no time!"

Figure 2: Dialogue lines from *Torchlight II* quest data for the quest "The Merryweather Poultice"

Dataset: Knights of the Old Republic dialogue

Star Wars: Knights of the Old Republic (KOTOR) is a turn-based action RPG by BioWare (2003). The game, which is set in the Star Wars universe, is famous for its high-quality writing, complex narrative and branching dialogues. During conversations with NPCs, players can choose from a set of pre-written dialogue options. Depending on their choices, different things happen in the game. Player's choices affect player character's abilities, NPC relations and story endings.

KOTOR's dialogues are also highly subjective and affective. The game story deals with the battle of good against evil, and conversations in the game reflect this theme: dialogues do not only revolve around collecting information, but also around feelings, relationships, and complex moral choices. Because of this, the dataset contains many different dialogue acts: characters joke, fight, grieve, lie, bargain, persuade and fall in love with each other. For an in-depth discussion of KOTOR's narrative, we refer the reader to (Wardrip-Fruin 2009, p. 59–69).

Text from KOTOR is not easily accessible outside the game, since the game assets are stored in compressed archive files in a proprietary format. We extracted all game assets with text using `xoreos-tools`, a collection of open-source modding tools[4] provided by the xoreos project. We then used a customized Python parser to parse the game files. The parsed data could be used to reconstruct all dialogue trees from the game to create a dialogue corpus in CSV-format.

Dataset contents Our final dataset contains over 25,000 lines of dialogue of 556 uniquely-named dialogue participants (listeners and speakers). The dataset contains 3305 dialogue tree root nodes, i.e. dialogue lines where the player or an NPC starts a conversation. Besides conversations between multiple humanoid characters, the dataset also includes interactions between the player and droids (robots), security systems, doors, and other interactive game objects, as the game models these interactions as dialogue lines. For example, if the player interacts with a droid, the droid might "reply" with "This droid is damaged and inactive". Since it is text data from the game's dialog files, we decided to keep these object interactions in the dataset.

Each datapoint in the dataset describes one turn in a conversation between two or more characters. Besides the dialogue text, the dataset contains the name of the speaker, the name of the listener (optional), the dialogue tree (references to other dialogue lines), which character animations should be played during the dialogue line, and game developer comments. For an overview of the information included in each datapoint, see Figure 3.

Because KOTOR's dialogue is branching, dialogues are graphs. As some of these graphs are cyclical (by making certain choices, the player can have a conversation that never ends), these are strictly speaking not dialogue trees. For ease of access, we have stored these graphs as double linked lists: each data point contains a list of predecessors and successors.

Applications In contrast to the *Torchlight II* dataset, the KOTOR dataset consists of only one type of text: dialogue. We can use it to analyse and generate both linear and branch-

[4] https://github.com/xoreos/xoreos-tools

ing dialogue.

The main strength of this dataset is its size, in terms of total lines of dialogue, the different speakers, and the breadth of the covered topics and sentiments. Because the lines were directly extracted from a game that is known for its high-quality writing, the dataset can be considered representative of commercial video game writing. As a result, this dataset can be used for style analysis, and training dialogue generation systems where the envisioned application domain is video games. The dataset is annotated with speaker and listener information, and some conversations involve more than two characters, so the dataset can be used for multi-party dialogue generation. Additionally, the dialogues can be used for analysing character relationships and sentiment. It can also be used to study the writing of a particular genre or setting, in this case science fiction and the Star Wars universe. Because of the high number of domain-specific fantasy words, the dataset can also be used to evaluate NLP techniques for domain-specific language.

2207 dialogue lines are annotated with animation data that indicate which character animations should be played during the delivery of the dialogue line. Although less than 10% of the dataset is annotated this way, the animation annotations are particularly rich because they convey emotions of game characters. In other words, we can interpret these annotations as affective labels. Figure 4 contains an example of a dialogue where the lines have animation annotations. The dialogue lines with affective labels can be used for sentiment analysis and affective text generation.

We can use this dataset as a basis for smaller, task-specific datasets. For example, we could filter the dataset for questions and answers by searching for dialogue trees with question marks. We could also filter dialogues with personal histories (search for sentences with high sentiment and subjectivity scores), jokes (lines with 'laughing' animations), or requests for help (lines with a 'talk pleading' animation).

Dataset: *The Elder Scrolls* documents

The Elder Scrolls (TES) is a series of video role-playing games by Bethesda Softworks (1994–2014), consisting of single-player role-playing games *Arena* (1994), *Daggerfall* (1996), *Morrowind* (2002), *Oblivion* (2006), *Skyrim* (2011), and an MMORPG, *The Elder Scrolls Online* (2014). Games in the series are open-world games, which means the player can explore the game world at their own pace and choose which objectives they want to focus on. The games take place in a fantasy world called Tamriel, which has a rich history that is communicated in various ways throughout the game: through NPC dialogues, quests objectives, cut scenes, and in-game documents, such as books and notes. These documents are collectible objects that the player can find as they travel through the world. Books can be opened and read by the player. Their length varies from a few words to a few hundred words, and some books are part of a series of multiple volumes. The books contain flavor text, i.e. text that is not a critical part of the game's main narrative, but gives the player background information about the world they are exploring.

The Imperial Library[5] is a fan-website for *The Elder Scrolls*, which collects in-game documents from the series. We scraped the text of over 4800 in-game books, letters and notes from the website. The dataset includes documents from all six role-playing games in the series.

Dataset contents The final dataset consists of 4890 documents (at least 4470 unique titles) from six games. Together, they form a corpus of over 160,000 sentences and 2,000,000 tokens. The Imperial Library website lists metadata for the in-game documents, such as title, fictional author information, and a short summary of each document. We annotated the texts of the documents with this metadata. For an overview of the structure of the data and an example, see Figure 5.

Applications The TES dataset consists of flavor text (decorative text) that describes the game world that the player's character inhabits. It can be used to study the structure and contents of game lore and game settings. This can be used for analysis, like in the research of Landwehr, Diesner, and Carley (2009), or for generation of new game lore (Grinblat and Bucklew 2017; Hall, Williams, and Headleand 2017).

The in-game books are interesting to analyse because of the way they explicitly inform the player about the game world, which differs from dialogue. The dialogues from *Torchlight II* and KOTOR mostly implicitly describe their setting. In KOTOR, players should derive the meaning of words like *rancor* (a monster), *droid* (robot) and *vibrosword* (a melee weapon) from their context, as these terms are not explained in the game. This differs from the books in *The Elder Scrolls*, which explicitly describe the game's high-fantasy setting through fictional reference works such as dictionaries, maps, manuals, cookbooks and histories.

Because of its size, the dataset can also be useful in cases where a relatively large corpus is needed for machine learning. For example, Bergsma, van Stegeren, and Theune (2020) used a preliminary version of this dataset in their research on sentiment analysis for game texts.They created a language model from the lore text to learn the implicit relations between English words and non-English words from the games' setting, which was then used to adapt a sentiment analysis lexicon for English to the domain of *The Elder Scrolls*.

Conclusion

We have proposed a list of requirements for video game text corpora: richness, representativeness, diversity and portability. We have discussed the places where source data for building new corpora can be found, namely in game files and on fan websites. Finally, we have presented three ready-to-use datasets with text from a number of popular role-playing games. These datasets can be used for various applications, such as NPC personality modeling, sentiment analysis, dialogue generation, lore generation, and quest generation.

The datasets and the code for this research are available online: https://github.com/hmi-utwente/video-game-text-corpora.

[5] https://www.imperial-library.info/books/all/by-category

Key	Description	Example value
Id	28209	Identifier of this dialogue act in the dataset
Speaker	Judge Shelkar	The character or object that communicates the line
Listener	PLAYER	The character that listens to the line
Text	For your crimes against Manaan and the Selkath you are banned forever from this world, on pain of death!	String literal
Animation	‘Judge Shelkar’: ‘Talk_Forceful’	3D animation that should be played during the delivery of the line
Comment	if the player is exiled	Game development notes
Previous	[28208, 28252, 28314, 28332]	Identifiers of previous dialogue lines
Next	[28210, 28213, 28215, 28218]	Identifiers of next possible dialogue lines, i.e. possible replies
Source DLG	man26_pcexile	The game file in which this dialogue act can be found.

Figure 3: Datapoint (conversation turn) from the KOTOR dataset. Dialogues consist of multiple turns. Dialogues are stored as double linked list and can be reconstructed by walking the linked list, i.e. following the ‘previous’ and ‘next’ references.

Id	Speaker	Text	Animation
28181	Mandalorian	You’ve been holding out on us again. Since you haven’t given us enough money, I guess we’re going to have to take it out of you piece by piece!	‘Mandalorian’: ‘Taunt’, ‘Farmer’: ‘Horror’, ‘Duros Warrior’: ‘Talk_Laughing’
28182	Farmer	No! Please! Take my wife and children instead! Anything!	‘Farmer’: ‘Talk_Pleading’, ‘Mandalorian’: ‘Ready weapon’
28183	Mandalorian	Ha-ha! Mmm... Wife and children. Sounds like a good idea...	‘Mandalorian’: ‘Victory’, ‘Duros Warrior’: ‘Talk_Laughing’, ‘Duros Warrior’: ‘Talk_Laughing’, ‘Duros Warrior’: ‘Talk_Laughing’

Figure 4: Reconstructed dialogue from the KOTOR dataset. This dialogue is taken from a cut scene in which raiders, a Mandalorian and multiple Duros aliens, harrass a farmer on the planet Dantooine. All three lines contain animation data. Every turn has only one possible successor, so this dialogue is linear. After line 28182, the Mandalorian shoots the farmer.

Key	Description	Example value
game	The game from which this document is sourced.	Morrowind
url	The url of the original webpage	https://www.imperial-library.info/content/dying-mans-last-words
author	The (fictional) author of the document	Indie
title	The title of the document	A Dying Man’s Last Words
summary	Summary of the document	The last words of a world-renowned archaeologist.
text	Text of the document	It’s been many days since the collapse. I have had many good and exciting adventures. I fear this is the last. I am still unsure what happened. (...)

Figure 5: Datapoint (in-game document) from the *The Elder Scrolls* dataset for the document titled *A Dying Man’s Last Words*.

Acknowledgments

This research is supported by the Netherlands Organisation for Scientific Research (NWO) via the DATA2GAME project (project number 055.16.114). We would like to thank the reviewers for their useful remarks. Judith van Stegeren would like to thank Michiel Werring and the members of the xoreos project for their helpful suggestions during data collection.

References

Bamman, D.; O'Connor, B.; and Smith, N. A. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the ACL (Volume 1: Long Papers)*, 352–361. Sofia, Bulgaria: Association for Computational Linguistics.

Bergsma, T.; van Stegeren, J.; and Theune, M. 2020. Creating a sentiment lexicon with game-specific words for analyzing NPC dialogue in the elder scrolls V: Skyrim. In *Workshop on Games and Natural Language Processing*, 1–9. Marseille, France: European Language Resources Association.

Bethesda Softworks. 1994–2014. *The Elder Scrolls I-V* and *The Elder Scrolls Online*. Game series [PC]. Bethesda Softworks, Rockville, Maryland, US.

BioWare. 2003. *Star Wars: Knights of the Old Republic*. Game [PC]. LucasArts, San Francisco, US.

Caropreso, M. F.; Inkpen, D.; Keshtkar, F.; and Khan, S. 2012. Template authoring environment for the automatic generation of narrative content. *Journal of Interactive Learning Research* 23(3):227–249.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL 2019: Human Language Technologies*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Godfrey, J. J.; Holliman, E. C.; and McDaniel, J. 1992. Switchboard: Telephone speech corpus for research and development. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, 517–520. IEEE.

Green, M. C.; Barros, G. A.; Liapis, A.; and Togelius, J. 2018. Data agent. In *Foundations of Digital Games 2018*, 19. ACM.

Grinblat, J., and Bucklew, C. B. 2017. Subverting historical cause & effect: generation of mythic biographies in Caves of Qud. In *Foundations of Digital Games 2017*, 1–7. New York, NY, USA: ACM.

Hall, J. A.; Williams, B.; and Headleand, C. J. 2017. Artificial folklore for simulated religions. In *2017 International Conference on Cyberworlds (CW)*, 229–232. IEEE.

Kreminski, M.; Karth, I.; and Wardrip-Fruin, N. 2019. Generators that read. In *Foundations of Digital Games 2019*. New York, NY, USA: Association for Computing Machinery.

Kybartas, B., and Verbrugge, C. 2014. Analysis of ReGEN as a graph-rewriting system for quest generation. *IEEE Transactions on Computational Intelligence and AI in Games* 6(2):228–242.

Landwehr, P.; Diesner, J.; and Carley, K. M. 2009. The Words of Warcraft: relational text analysis of quests in an MMORPG. In *Proceedings of DiGRA 2009*. Brunel University.

Lee, D.; Lin, D.; Bezemer, C.-P.; and Hassan, A. E. 2020. Building the perfect game–an empirical study of game modifications. *Empirical Software Engineering* 1–34.

Lin, Z.; Gehring, J.; Khalidov, V.; and Synnaeve, G. 2017. Stardata: A StarCraft AI research dataset. In *Thirteenth AIIDE Conference*.

Lukin, S. M.; Ryan, J. O.; and Walker, M. A. 2014. Automating direct speech variations in stories and games. In *Tenth AIIDE Conference*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc. 3111–3119.

Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL 2016: Human Language Technologies*, 839–849. San Diego, California: Association for Computational Linguistics.

Orkin, J., and Roy, D. 2007. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development* 3(1):39–60.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. https://github.com/openai/gpt-2. Retrieved August 31, 2020.

Runic Games. 2012. *Torchlight 2*. Game [PC]. Runic Games, Seattle, Washington, US.

Ryan, J.; Seither, E.; Mateas, M.; and Wardrip-Fruin, N. 2016. Expressionist: An authoring tool for in-game text generation. In *International Conference on Interactive Digital Storytelling*, 221–233. Springer.

Schlünder, B., and Klabunde, R. 2013. Greetings generation in video role playing games. In *Proceedings of the 14th European Workshop on NLG*, 167–171.

Summerville, A. J.; Snodgrass, S.; Mateas, M.; and Ontanón, S. 2016. The VGLC: The video game level corpus. In *Workshop on Procedural Content Generation*.

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgård, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2018. Procedural content generation via machine learning (PCGML). *IEEE Transactions on Games* 10(3):257–270.

Walker, M. A.; Grant, R.; Sawyer, J.; Lin, G. I.; Wardrip-Fruin, N.; and Buell, M. 2011. Perceived or not perceived: Film character models for expressive NLG. In *ICIDS*, 109–121. Springer.

Walton, N. 2019. *AI Dungeon*. Game [PC, Android, IOS]. https://www.aidungeon.io.

Wardrip-Fruin, N. 2009. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. The MIT Press.