Movie Guessing Name Dialogue System

Ziang He
Department of Computer Science
Georgetown University
zh188@georgetown.edu

Xinru Shi
Department of Computer Science
Georgetown University
xs176@georgetown.edu

Abstract

Guess can be intriguing because of uncertainty and excitement when figuring out the answer. Movie guessing game is a niche field. We designed a movie title-guessing game with a task-oriented dialogue system built via Rasa. This system aims to give movie lovers an entertainment platform. This paper mainly discusses the system structure and experiments. It succeeds in combining a word-guessing game with the dialogue system with a highly subjective and objective evaluation during the chat.

1 Introduction

The Wordle game is popular and encourages people to spend pieces of time playing with it for idleness. People would engage their brains and use their knowledge to try and figure out the answer to the question. This can be mentally stimulating and can provide a sense of satisfaction when the guess is correct. Therefore designing a movie guessing game can realize the same goals. It can be a fun and engaging way for people to test their knowledge of movies and challenge themselves and others to see who can guess the most movies correctly. It can also be a good way to introduce people to new movies and help them discover films they might not have heard of before. Additionally, a movie guessing game can be a good way to bring people together and facilitate social interaction, especially if it is played with a group of friends or family members.

We develop a task-oriented dialogue system that supports a complete word-based movie guessing game, while players try to guess the titles of movies based on hints provided and limited to rules. The system achieves functionality including generating questions, providing hints, providing feedback, keeping track of counters, and checking answers. We also customize the NLU corpus, intent prediction, and actions via RASA. Both subjective and Objective evaluation metrics are used for improvement.

More details will be discussed below.

2 Related Work

People always apply games for different objections. For dialogue systems, when a chatbot is merely a game, we can call it a game with a purpose (GWAP). These gamified systems are involved in both research and casual life improving efficiency.

Some scholars try to incorporate gaming into an annotation task to solve term associations.(Artignan et al., 2009) DEAL is a game interface designed for users to learn second languages.(Hjalmarsson et al., 2007) Also CALL-SLT is developed, an open-source translation game aligning with six languages, in order to improve fluency for language beginners. (Rayner et al., 2014) Study released the game Shiritori into Onsei-Assist which is a Siri-like assistant and found that adding a game could increase user engagement in daily life.(Kobayashi et al., 2015)

The movie-name guessing game is a small domain that we can hardly find related projects. The project from ankitjosh78 (2021) provides movie guessing depending on characters. The only interaction is guessing and judgment. Some public movie games on websites all rely on visual engagement, which means after reviewing some snaps in a specific movie, users should give their answers.

Our work aims to develop a system without any research purpose and bring fun for people. Inspired by the popular word game Wordle, the embedded movie hints are all word-based rather than pictures.

3 Framework

Our system is built upon the environment $Python 3.7+$. All data processing depends on the external library Pandas providing various data manipulations. The framework developed is $RASA 3.1$, which supports the NLU pipeline, deep learning models and built-in functions. To exhibit the whole conservation in a proper way, we choose $Widget 2.0$[1] connected with the RASA server. (Figure 1)

3.1 Data

Behind the game, all movies are scratched from the existing IMBD dataset[2] with about 5000 quantity. We select some columns of the original features, year, director, movie_title, genres, and actors.

We also applied TMDB(The Movie Database) API to scrape overviews for each movie based on its name and release year. To help users contract long sentences, the Stanford Stanze library is utilized to extract crucial noun words from these recaps. For the same movie having different genres, we expand them using one-hot encoding to represent whether it belongs to a specific genre. Eventually, there are 22 genres in the whole dataset. We can see the post-processed data example without genres as the Figure 2 shows.

We have two difficulty levels within the game that will be discussed in the next section. For the easy mode, we screen out the films overlapping with those with high ratings at IMDB[3] resulting in 586 movies at last.

3.2 Game Design

As shown in the designed workflow (Figure 3), the player needs to first tell the chatbot like, *Start.* or *Please start the game.* The system will then ask the player to choose a difficulty level from the easy class and the hard class, and tell the player to choose a favorite genre that he wants to guess. At this step, the chatbot will also tell the player all available genres can be chosen. After receiving a genre from the player, the system will randomly search for the name of a movie in this genre and reply to the player with an encoded name, and now, the player can start guessing the name.

The next part of the flow chart mainly reflects the rules of the game.

1) Each player has 4 chances to ask for hints from the chatbot. The chatbot will reject the requirement if it is the fifth time it gives a hint.

2) If the movie name has only one word, the system will hide part of the letters with x. If the movie name contains multiple words, each hidden word is hidden by x. Special symbols such as quotes are supported. When they are hidden, they will not default to a word being replaced by an x symbol.

3) If the player can guess the name of the movie correctly within 3 chances, the player wins. Otherwise, the chatbot will announce the player's game failure when the answer is wrong for the third time.

4) The player can restart the game after guessing the correct name or using up all guessing chances.

The hint can be a general statement like *I want a hint.* Or the player can ask for a specific question related to actors, years, directors, or keywords, like *Could you tell me the release year?* And the guessing pattern should be *The movie is: { name of movie}*

We will illustrate this flow chart in more detail in §4 Experiments, like how to implement difficulty selection, counters, the NLU design, and custom actions.

3.3 Web

The frontend design involves three pages that can be chosen on the menu: Game Rules, The Beginner's Guide, and Help us to do better, as Figure 4 shows. Players can check detailed game rules on the first page and read a game example on the guide page. Also, we collect players' comments and suggestions on this game from the survey on the last page.

Players can click the round button at the lower right corner of the webpage to start chitchatting with our chatbot, and a chat interface will be

[1] https://github.com/JiteshGaikwad/Chatbot-Widget/

[2] https://www.kaggle.com/datasets/rakkesharv/imdb-5000-movies-multiple-genres-dataset

[3] https://www.kaggle.com/datasets/mustafacicek/imdb-top-250-lists-1996-2020/

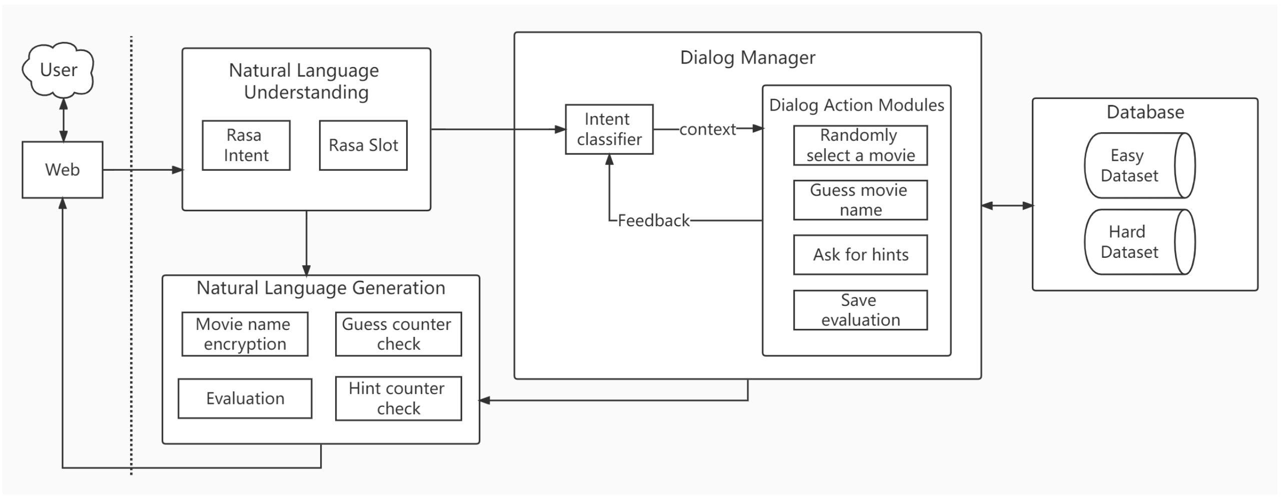
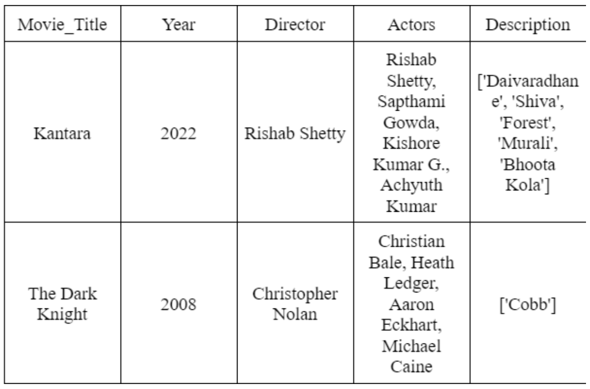


Figure 1: The pipeline



Movie_Title	Year	Director	Actors	Description
Kantara	2022	Rishab Shetty	Rishab Shetty, Sapthami Gowda, Kishore Kumar G., Achyuth Kumar	['Daivaradhane', 'Shiva', 'Forest', 'Murali', 'Bhoota Kola']
The Dark Knight	2008	Christopher Nolan	Christian Bale, Heath Ledger, Aaron Eckhart, Michael Caine	['Cobb']

Figure 2: Data Example

popped up as shown in image 5.

Three functions are hidden in the menu at the upper right corner of this interface: clear, restart and close. Players can click 'Clear' to clear all messages in the chat window but chat history is still saved in the backend system, and they can continue chatting with the bot. 'Restart' is a function same as the restart function in Rasa. 'Close' means to minimize the chat window.

4 Experiments

For the customized RASA system, we should define the configuration for conversation sessions. The pipeline includes intents, entities, slots, responses, forms, and actions.

4.1 Slots

Slots are the long-term memory in a conversation. We could save information in the slots and use them later. As mentioned in the game flow chart, we have several slots throughout the conversation.

- Difficulty: We choose the built-in button function to make it more interactive. This slot will be filled when the user clicks the corresponding element. The easy and hard class points to different data files.
- Genre: When the users provide their preferred genre, the system would extract the word and use it for name generation.
- Name: This slot only works by the system. After making up the genre slot, the system would extract a movie and fill it into the name slot. The whole game depends on this slot to retrieve hints.
- hint: This slot is used for offering hints after requesting. If the system detects the hint slot can be filled, the slot would be canceled after the given response.
- Counter: We design two counters: a hint counter and a guess counter. They are all used to limit user chance in accordance with rules

4.2 Entities

Entities are structured pieces of information inside a user message. In our system, we mainly apply entities for filling the slots, especially genre and hint. Difficulty and bot_eval both link to the direct button and payloads so that the system can capture these entities with briefness. The following formats illustrate what are the entities in user sentences.
I want a hint about [year]: (hint) and *I'd like [action]: (genre) movie*

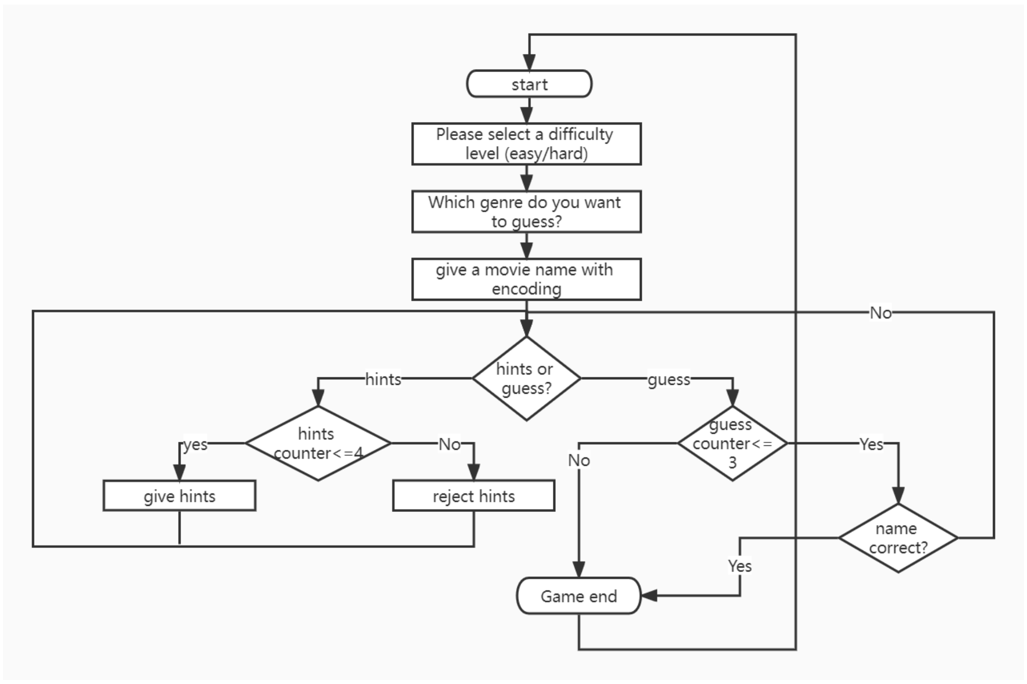
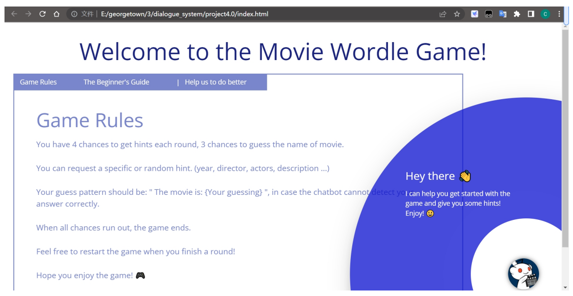


Figure 3: The designed workflow of this system


E:/georgetown/3/dialogue_system/project4.0/index.html

Welcome to the Movie Wordle Game!

Game Rules | The Beginner's Guide | Help us to do better

Game Rules

You have 4 chances to get hints each round, 3 chances to guess the name of movie.

You can request a specific or random hint. (year, director, actors, description ...)

Your guess pattern should be: " The movie is: {Your guessing} ", in case the chatbot cannot detect your answer correctly.

When all chances run out, the game ends.

Feel free to restart the game when you finish a round!

Hope you enjoy the game! 🎮

Hey there 👋

I can help you get started with the game and give you some hints! Enjoy! 😃

Figure 4: Web page

Here system points year to entity hint and action to entity genre.

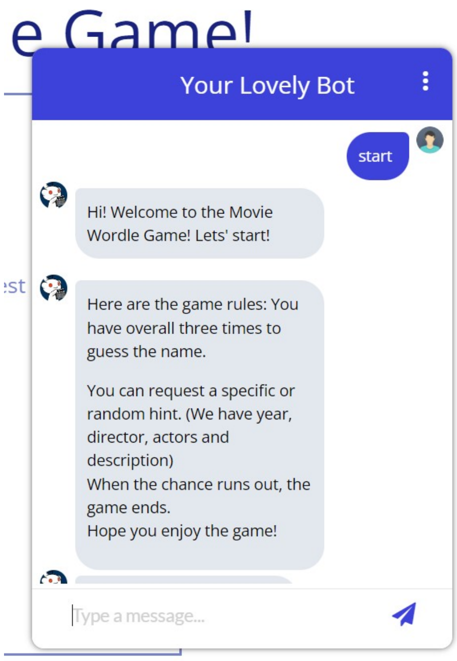
4.3 Intents

The intents are the inputs from users that would be predicted for each user's turn. Thus the system could make the corresponding action. For our movie guessing game, we specify some intents.

- Start: Users initialize the game which is the debut of the conversation.

- Provide_genre: Users can state their preferred movie genres. We specified all 22 genres in this part making it easier to extract the genre entity.

- Request_hint: It's the intention for requesting hints in order to figure out the answer. Still, we cover all four categories. As we find it's easy to have a typo for the word 'description', we add 'keyword' as a hint entity in this intent.


e Game!

Your Lovely Bot

start

Hi! Welcome to the Movie Wordle Game! Lets' start!

Here are the game rules: You have overall three times to guess the name.

You can request a specific or random hint. (We have year, director, actors and description)
When the chance runs out, the game ends.
Hope you enjoy the game!

Type a message...

Figure 5: The chat window

- guess: In this intent, we fix the answer format *The movie is: answer* instead of writing distinctive answer patterns.

4.4 Actions

In order to realize all functions, we define extra six custom actions.

- validate_genre_name_form: This form involves two slots: difficulty and genre. The system randomly selects a movie and encodes its name based on the difficulty slot (to find the data path) and genre slot (to narrow down the scope). Initially, we intend to match the name slot with a custom action. However, the extraction method iterates for each user turn and refresh it as None. Thus we put the name-filling manipulation in the form validation function.
- action_hint: The system will search for the information from the dataset when the player asks a specific question to it, or randomly give a hint if the message is a general statement or question. Other functions are also included here, such as only giving one actor's name each time, ensuring that the hint given is different each time, and checking that the information the player wants is in the database, etc.
- action_guess: Judge whether the answer given by the player is correct or not. If the answer is correct, this function would reset all slots and tell the users to restart a new game.
- action_guess_counter: After each turn, count the number of times the intent sent by the player is a 'guess' based on the events in the dialogue history. When users reach the guess limitation, it would tell them the answer and stop the game. Besides, it will send a message to announce the failure if the name is incorrect for the third time.
- action_hint_counter: Similar to the previous one, the system counts the number of times the intent sent by the player is a 'hint'. It would also warn the users not to ask for more hints when it comes to the limitation.
- action_record_eval: Save the evaluation score to a local file for each game.

4.5 Improvement

Compared with the initial version of the system on Dec.1st, we move forward and make some improvements.

Hints adjustment As our game only provides four categories of hints, the users always can run out of all chances for getting intact information about movies. Therefore we decide to give out a random name or keyword from hint actors and description, which adds more fun and enhances the difficulty.

A build-in evaluation Online agents always request the users to comment on their performance. To simulate the real dialogue systems, we add an evaluation utterance after each story. This can help us receive quantitative scores more smoothly. It also enables users not to spend time filling up the long questionnaire on the 'help us better' page if they are reluctant.

Error handling We cover error handling in hint and genre extraction. If users have a typo for specific hints, the system would warn them to enter the correct name. However, they could lose a precious chance for requesting more hints. In terms of genre validation, we would iterate this process until we successfully extract the genre in provided range.

5 Evaluations

We mainly use Subjective metrics and complementary two objective metrics to evaluate our system. We'll talk about them in the next two sections.

5.1 Subjective Metrics

We form a survey on the website covering questions from rule clarity to bug detection. We invite two friends to assess our initial system. Credit to their qualitative suggestions, we make some progress mentioned in §4.5. We collect 12 overall scores, 3 scores about website spoken style satisfaction, and 3 ratings for difficulty evaluation after Dec.6th alongside two pieces of advice. The first feedback is that the response speed from the system is not reasonable. Also, some people are not willing to enter a long fixed guess format. They want to add more possible patterns. The table 1 represents the average scores.

We can see that users are satisfied with the whole performance of the system. The difficulty is relatively high as those we invited are all non-English native speakers, thus the scores have bias. Mean-

Table 1: Subjective Metrics

Avg Overall score	Avg spoken style satisfaction	Avg difficulty
4.08	4.33	4.00

while, our interesting format gives a great impression to users.

5.2 Objective Metrics

We use two metrics for objective evaluation, average hints requesting numbers and guess intent prediction accuracy. The table 2 provides the results:

Table 2: Objective Metrics

Avg Hints Numbers	Guess Pred Acc
3.58	0.74

The users almost run out of chances to request hints, which aligns with the difficulty evaluation. It's due to some typos as well. Some users would directly enter their guessing instead of the default format, the system would thus predict it as zero_day intent and track back to the action_listen without any response. Similarly, when a user enters the message *movie: answer*, the system would also neglect this intent.

6 Conclusions

Based on the framework Rasa, we finished designing a task-oriented system capable of executing a movie title-guessing game and interacting with users. The system helps people who want to have fun and enables movie buffs to show off their expertise. The overall performance is reasonable though it's relatively tough for those, not native speakers.

The system can be published online for entertainment in the future but it needs refinement yet. First, the movies in the database are not enough for thousands of trials. Maybe one person can encounter the same movie, especially for small fields. Also, it only predicts intents with a few scenarios. The system can't identify new expressions and messages not occurring in the training corpus. Additionally, the number of turns for the conversation can't meet the user's requirement. All the turns are system-initialized with proper prompts. We can add more rules so that users can raise questions within the game and the chatbot responds. Theoretically, we can't cover all distinctive situations, thus we still have a long way to progress.

References

ankitjosh78. 2021. hang-movie-man.

Guillaume Artignan, Mountaz Hascoët, and Mathieu Lafourcade. 2009. Multiscale visual analysis of lexical networks. In *2009 13th International Conference Information Visualisation*, pages 685–690.

Anna Hjalmarsson, Preben Wik, and Jenny Brusk. 2007. Dealing with DEAL: A dialogue system for conversation training. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 132–135, Antwerp, Belgium. Association for Computational Linguistics.

Hayato Kobayashi, Kaori Tanio, and Manabu Sassano. 2015. Effects of game on user engagement with spoken dialogue system. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 422–426, Prague, Czech Republic. Association for Computational Linguistics.

Manny Rayner, Nikos Isourakis, Claudia Baur, Pierrette Bouillon, and Johannna Gerlach. 2014. CALL-SLT: A spoken CALL system based on grammar and speech recognition. In *Linguistic Issues in Language Technology, Volume 10, 2014*. CSLI Publications.