Table of Contents

DOI: 10.1145/3774399.3774408

Links

SIGAI website: http://sigai.acm.org/
Newsletter: https://sigai.acm.org/main/aimatters/
Blog: https://sigai.acm.org/main/blog/
X: https://x.com/acm_sigai
Edition DOI: 10.1145/3774399

Join SIGAI

Students $11, others $25
For details, see http://sigai.acm.org/
Benefits: regular, student
Also consider joining ACM. Our mailing list is open to all.

Letter from Chairs

DOI: 10.1145/3774399.3774401

Dear Members,

After a short pause, we're excited to announce that AI Matters is returning with a new vision, a broader scope, and an updated format. We envision the newsletter as a useful resource for the three primary communities of ACM SIGAI – AI researchers, professional practitioners, and students – helping everyone stay up to date with developments across the field. AI Matters will continue to advance our mission of promoting the growth and application of AI principles and techniques through computing. We invite you, the members, to take part in this.

Alongside the usual conference reports, we plan to feature research highlights, opinion pieces, themed articles, tutorials, book reviews, advice for early-career researchers, competitions, and more. However, we can only make all of this possible with your contributions. If you have an idea for a column to write regularly, a topic to explore, or strong opinions on AI to share, we'd love to hear from you. We also welcome personal reflections and observations from your work in AI, as well as lighter pieces — from puzzles and jokes to memorable or amusing encounters with LLMs.

As well as reaching your ACM SIGAI colleagues, any relevant content can also be posted to AIhub.org and Robohub.org, so you can reach a wider audience. If you would like any advice or training in communicating your ideas, the team at AIhub.org can help.

You can expect these changes from the next issue onwards. If you would like to get involved, please contact us at aimatters@sigai.acm.org. Full submission guidelines are on our website at https://sigai.acm.org/main/ai-matters/. For now, enjoy this issue, where you'll find reports on the conferences we sponsored over the last year, and highlights from the Mentored Undergraduate Research Challenge.

Warm regards,

Nicholas Mattei
Chair, ACM SIGAI

Sanmay Das
Past-Chair, ACM SIGAI

Mentored Undergraduate Research Challenge

DOI: 10.1145/3774399.3774402

The Mentored Undergraduate Research Challenge (MURC) is an opportunity for undergraduate students to experience the complete research life-cycle for a semester or summer, working under the guidance of a mentor. For the 2025 MURC, the topic was "Playing Word Association Games". Participants asked questions about how AI-based systems can choose words that not only go together, but also satisfy other player's interests, personalities, and more, when they evaluate whether they like the word selections. Five teams completed the challenge, exploring a variety of problems and approaches that contribute to innovations in the topic's research area.

Calls for MURC 2026 will be published in AI Matters in a future issue. If you have any queries, please reach out to Rick Freedman at rfreedman@sift.net

Image Credits: "Digital Nomads Beyond the Cubicle" — Yutong Liu & Digit, https://betterimagesofai.org/ https://creativecommons.org/licenses/by/4.0/

Decoding the Chameleon Game

Authors

Tri Dang * (DePauw University; tridang 2025@depauw.edu)
Hieu Tran (Purdue University; tran335@purdue.edu)
Brian Howard (DePauw University; bhoward@depauw.edu)
Sutthirut Charoenphon (DePauw University; sutthirutcharoenphon@depauw.edu)
Dat Nguyen * (DePauw University; datnguyen 2025@depauw.edu

DOI: 10.1145/3774399.3774403

Abstract

The Chameleon game is a challenging word association activity where players are given a secret word and must respond with words relevant to that secret word. It requires strategic thinking and deduction. The Chameleon must cleverly guess the secret keyword in this game while avoiding suspicion. Our research aims to create an advanced artificial intelligence (AI) model that can play the Chameleon game from both perspectives: as the Chameleon and as a Human. This AI is designed to guess secret keywords based on the information provided by the players, choose the best strategies to avoid detection as the Chameleon, identify and vote for the most suspicious players when acting as a Human, and learn from previous games to improve its performance.

Introduction

Word association games, where players are given a word (the cue) and must provide the first word that comes to mind (De Deyne, Navarro, Perfors, Brysbaert, & Storms, 2019) have gained significant traction due to their engaging and strategic nature, offering both entertainment and cognitive benefits. These games, which involve players generating words in response to given cues, enhance various cognitive functions such as attentional control, working memory, and problem-solving (Ornaghi, Brock meier, & Gavazzi, 2011). Research shows that games (that is, chess and word association) can improve strategic thinking, memory, and creativity (Martinez, Gimenes, & Lambert, 2023)(Mercier & Lubart, 2021). While traditional word association games provide valuable cognitive and social benefits, they often rely on in-person interactions, which can be hindered by social isolation or geographical distance. Although networked online play can partially address these challenges, finding a sufficient number of players to participate simultaneously remains a significant obstacle.

To address this, we developed an AI model for the Chameleon game, a variation of word association where players must deduce a secret keyword while the Chameleon blends in by providing clues. By integrating techniques (e.g. Word2Vec, GloVe embedding, Naive Bayes, and Feed Forward Neural Network), the AI will learn from past gameplays and adapt to players' strategies over time. This approach not only replicates the interactive challenge of the original game but also creates a dynamic and evolving system that offers increasingly sophisticated gameplays, bridging social gaps and enhancing cognitive engagement.

Related Works

In board games, AI has demonstrated remarkable advancements through mastering complex games like Go, Chess, Poker, and Shogi. In “Mastering the game of Go without human knowledge”, Silver et al. demonstrated how deep reinforcement learning can achieve superhuman performance through self-play (a method where an AI trains by competing against itself, learning and improving through repeated games), and highlighted AI’s potential in strategic environments (Silver et al., 2016). Similarly, Silver and Hubert demon strate that general reinforcement learning algorithms using self-play and extensive training enable AI to challenge humans in Chess and Shogi (Silver et al., 2018), while Brown and Sandholm developed Pluribus, an AI model that achieved superhuman performance in six player no-limit Texas hold ’em poker (Brown & Sandholm, 2019). These studies highlight AI’s transformative impact on strategic board games, demonstrating how self-learning algorithms achieve unmatched proficiency through rigorous training.

In word association games, AI applications are increasingly proving their values. Shi et al. used a black-box model to predict the difficulty of word games like Wordle, showing how AI can enhance players’ experience and game design by analyzing challenge complexity (L. Shi, Chen, Lin, Chen, & Dai, 2023). Additionally, research by Richards and Amir focuses on opponent modeling in Scrabble (Richards & Amir, 2007). Their work investigates how AI can strategically predict and respond to opponents’ moves, providing deeper insights into game dynamics and improving competitive play. In another research, Hasan and Gan developed an unsupervised adaptive brain-computer interface that improves game play in Hangman by learning from user brain signals (Hasan & Gan, 2012). Their system demonstrated enhanced accuracy and efficiency compared to non-adaptive methods. Together, these studies exemplify the growing role of AI in understanding and mastering word association games, from predicting game difficulty to modeling opponents’ strategies.

While AI models have been applied to various word-association board games, little to no prior work has focused on modeling **The Chameleon**. Designed by Rikki Tahta and published in 2017 by Big Potato Games, **The Chameleon** is a social deduction word association game where one player (the Chameleon) must bluff to conceal their lack of knowledge about a target word, while others attempt to identify them. The game’s intricate player interactions and strategic deception introduce significant challenges for AI simulation and analysis. In this work, we address these challenges by employing neural networks to model the complex interactions inherent in The Chameleon. Specifically, we focus on developing a model that can effectively act as the Chameleon, blending in and successfully deceiving the opponents. Thus, our work not only applied to this game specifically, but has the potential to model highly-complex interaction between humans in a social setting.

About the Chameleon game

Game Rules

The Chameleon game is designed for 4-8 players. The Game Components are shown in Figure 1. Starting from the Setup phase, players are randomly assigned as either Humans or Chameleons. All players receive a Code Card, except one who gets the Chameleon Card. A Topic Card with sixteen words relating to a theme is placed face-up. Humans use a grid reference from the Code Card to get a secret word. Next, in the Give attributes phase, players take turns giving one-word clues (attributes) related to the secret word. Human players will give attributes relevant to the keyword, while the Chameleon, who doesn’t know it, needs to guess attributes based on Human clues.

After all clues are given, the game comes to the Voting phase, where Humans vote to identify the Chameleon. If the Chameleon is identified (Voting True), the game comes to the Guessing Secret Keyword phase, where the Chameleon gets one chance to guess the secret word. A Correct guess (Guess True) means a win for the Chameleon, while an Incorrect guess (Guess False) means a win for the Humans. If the Chameleon is not identified (Voting False), the Chameleon

wins the round. The game is played over multiple rounds, with total points determining the overall winner.

In this study, we studied a six-player version under the food theme. The six-player setup balances the game's complexity, providing enough interaction to challenge the Chameleon while keeping the deduction process engaging for Humans. The food theme works well because it draws from everyday experiences, making it easy for players to offer specific, recognizable attributes like flavors, textures, or common dishes. Other themes, such as animals, movies, colors, geography, or sports, can also be engaging options; however, their accessibility may vary based on players' knowledge and interests, particularly for younger players. The example of a play is shown in Appendix A. Versions of Chameleon Games

The Chameleon game, a social deduction board game, has multiple versions, each offering unique gameplay elements. In the simple version, players don't have time to think before or during the Giving Attributes phase. As a result, subsequent players don't have time to consider previous players' clues and need to prepare a word before this phase. Voting in this version is based on identifying which attributes seem less relevant to the keyword.

In the complex version, during the Giving Attributes phase, players have time to think carefully before declaring unique attributes. Consequently, they can change their attributes or strategy based on previous players' clues. Voting considers various aspects, such as player behavior and the relevance of attributes across multiple keywords. In this research, we chose to focus on the harder version of the game, as mastering the more challenging aspects implies proficiency in simpler versions.

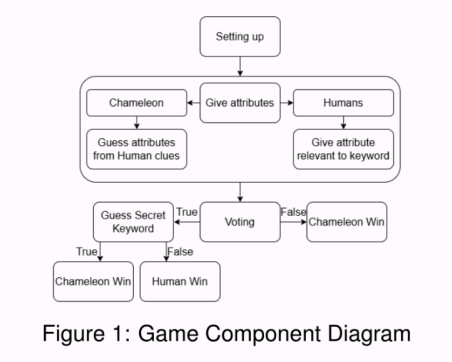


Figure 1: Game Component Diagram

Giving attribute as Chameleon

The Giving attributes phase is a key moment for the Chameleon, demanding both linguistic finesse and strategic forethought. Following the setup phase, where player roles and the secret word are determined, each participant must offer an attribute that upholds their credibility and furthers their strategic aims.

Attribute Representation in the Database

The database for the Chameleon game is structured to store attributes. Their relevance rate to each keyword is built on the idea of the Word2Vec model, which represents words in vector form (Jatnika, Bijaksana, & Suryani, 2019). A vector represents attributes in a 16-dimensional space, where each element indicates the relevance of that attribute to the corresponding keyword. When adding an attribute to the database, the default vector will be initialized with zeros, indicating no relevance to any keyword recorded. After matches, as the model learns from gameplay data, these vectors can be updated to reflect the true relevance of each attribute to the corresponding keywords.

For example, "Attribute 1" is initially added with a zero vector, indicating no relevance to any of the keywords. However, if "Attribute 1" is found to be relevant to "Pizza" and "Cake," the relevance value for "Pizza" and "Cake" in the vector will be adjusted accordingly. As the game progresses and more data is gathered, these relevance values will continuously refine, allowing the model to better understand the relationships

between different attributes and key words. This helps the technique make more informed decisions during gameplay, improving its ability to identify the keyword accurately or blend in as the Chameleon.

Getting and Handling Input Attributes

When it is the Chameleon's turn, the model will take the record of attributes provided by previous players as input to the model and, at the same time, use the database as a reference system to make predictions. When the AI model encounters a new attribute not present in the database, it will create a new entry for this attribute. This approach ensures that the model can handle unknown attributes and start with a neutral baseline. On the other hand, if the attributes already exist in the database, the AI model will retrieve the pre-existing vectors for these attributes.

Finding Attributes Intersection

After compiling the list of attributes given by the players, the AI model uses the data in the database to determine relevance. In this database, a value of 0 indicates no relevance between the meanings of attributes and key words, while a value greater than 0 indicates at least some relevance between them. The model then searches for keywords relevant to all the attributes other players provide. If the AI model cannot find any keywords that are relevant to all attributes (possibly due to missing data or because the combination of relevant keywords and attributes has never appeared in previous games), then it will progressively reduce the number of attributes it considers, checking keywords relevant to all but one of the previous players' attributes, then all but two attributes, and so on. This process continues until the AI model identifies at least one keyword that satisfies the requirement. By using this adaptive strategy, the AI ensures that it can always find at least a suitable key word, even when the data is incomplete or when encountering novel combinations of attributes and keywords. This approach helps the AI maintain robust performance and effective gameplay.

To illustrate how the Chameleon identifies the intersection of attributes, consider the scenario where the Chameleon is in position 4, requiring it to analyze the previous three attributes. These attributes are represented as vectors, with each position corresponding to a specific keyword and the value indicating the relevance of the attribute to the keyword.

The model compares the vectors at each position to find the intersection as shown in Table 1. The intersection occurs where all three attributes share a common keyword, indicated by non-zero values across the vectors at that specific position. In this example, the key words with index 2, 4, 6, 7, and 12 are identified as the intersection, where the Chameleon recognizes shared relevance among the previous attributes. This helps the Chameleon identify potential keywords that align with the input from previous players.

Keyword ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Attribute 1	2	1	0	3	0	2	3	0	1	2	0	3	0	1	0	3
Attribute 2	1	2	0	3	0	1	2	0	0	0	3	2	3	0	1	2
Attribute 3	0	3	2	1	1	3	2	3	0	1	2	3	2	3	1	0
Intersection		x		x		x	x					x				

Table 1: Attribute vectors and Intersection

Calculating Attributes Probability

After finding the attributes' intersection, we obtain a list of keywords that relate to all attributes from the previous players (the list of intersection keywords). Before calculating the probability, we set to zero any data that does not represent the relevance of an attribute to a keyword in the list of intersection keywords. This process is applied to all attributes in the database, ensuring that only the relevant values corresponding to the identified intersections are kept.

In Table 1, Attribute 1, 2, and 3 is represented by the vectors. The intersection was identified at index 2, 4, 6, 7, and 12. In the new table, only the values at these positions are retained, while all other values are set to zero as shown in Table 2.

Keyword ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Attribute 1	0	1	0	3	0	2	3	0	0	0	0	3	0	0	0	0
Attribute 2	0	2	0	3	0	1	2	0	0	0	0	2	0	0	0	0
Attribute 3	0	3	0	1	0	3	2	0	0	0	0	3	0	0	0	0
Intersection		x		x		x	x					x				

Table 2: Modified attribute vectors

This step helps avoid data distractions that might come from attributes not in the intersection column but with high weight. Such data can significantly reduce the accuracy of the intersection column, which affects both the probability calculation and the subsequent spreading step. This process involves creating a new table where values irrelevant to the maximum number of available attributes are set to zero. By doing this, the model focuses solely on the relevant data, reducing potential distractions from unrelated attributes. This refined dataset enhances the technique's ability to make precise predictions and blend in effectively as the Chameleon.

We calculate the probability of relevant attributes for each keyword to account for the relevance of attributes to keywords and avoid interference from irrelevant data. This helps ensure that only significant data is considered in the spreading calculation. The probability for each keyword is determined by dividing the relevance value for that attribute by the sum of all the relevance values in the row (after masking out those not in the intersection).

After calculating the probability for each cell, we obtain a table showing the relevance of each attribute to each keyword as a percentage. This table helps to understand which key words are more likely to be associated with the given attributes based on these probabilities. The percentage reflects the proportion of attributes relevant to each keyword compared to the total number of attributes available. Keywords with a relevance percentage greater than 0.01 are included in the spreading calculation. This threshold ensures that only keywords with a meaningful number of relevant attributes are considered, thus reducing noise and improving the accuracy of key word selection. For example, after applying the probability formula to Attribute 1 in Table 2, the resulting vector is shown in Table 3:

Keyword ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Attribute 1 Probabilities	0	0.08	0	0.25	0	0.17	0.25	0	0	0	0	0.25	0	0	0	0
Sufficient relevance		x		x		x	x					x				

Table 3: Attribute 1 probabilities

Following the rule above that considers an attribute to be relevant to a keyword if its percentage exceeds 0.01, we observe that there are five values that meet this criteria. Therefore, the spreading of attribute 1 corresponds to five keywords. After calculating the keyword spreading for all attributes, the table is then sorted by the number of keywords each attribute spreads, highlighting the significance of each attribute in the selection process.

To avoid predictable patterns for other players, the AI randomly selects between attributes with equal relevance within the range of in intersection (spreading). This randomness ensures that the AI's choices are not easily anticipated by other players, maintaining the element of surprise and making it more challenging for Humans to guess who is the Chameleon.

Updating the Database

After the match, the model records the attributes given by Humans and updates the database accordingly. For each attribute provided by the players that is relevant to the key word used in the match, the corresponding value in the database increases by 1. This increment represents an increase in the relevance of that attribute to the keyword, indicating that real players are more likely to choose that attribute when considering the keyword. This continual learning process allows the model to refine its understanding of the relationships between attributes and keywords. By incrementally updating the database after each match, the technique becomes better at making accurate guesses and effectively blending in as the Chameleon. This adaptive learning enhances the model's performance and its ability to participate in the game in a more human-like manner.

Voting

After all players finish giving the attributes, the game comes to the Voting phase as shown in Figure 1. In this phase, each player is presented with key information that includes the six attributes provided by all six players, one of whom is the Chameleon. Five Humans will know what the secret keyword is. These attributes are revealed in a specific order, corresponding to the order of the players. Each Human must assess these attributes' relation to the keyword to determine who might be the Chameleon. The Chameleon, however, operates under different conditions. In the "Guessing Keyword as Chameleon" phase, the Chameleon can deduce the secret key word based on the attributes provided by the other players. This deduction occurs before the Voting phase and does not rely on information from the Voting process itself. During this phase, the Chameleon uses the attributes given by other players to infer the keyword and strategically craft responses to blend in with the other players.

To enable Humans to identify the Chameleon and allow the Chameleon to use the network during voting to decrease suspicion and shift focus away from themselves, we employ a Neural Network (Guresen & Kayakutlu, 2011). The network layers are described in the following subsections, and the structure is shown in Figure 2.

Input Layer

The Neural Network's input layer is designed to process the information available to each player. Specifically, the player is given six attributes—one from each of the six players, including the Chameleon—and a secret key word. The first step in this process is to calculate the distance between each of the six attributes and the keyword using cosine similarity (Chowdhury, 2010).

These distances are essential as they measure how closely related each attribute is to the keyword, reflecting the likelihood that the attribute was chosen by someone who knows the keyword or by the Chameleon attempting to blend in. The resulting six distances are fed into the Neural Network as input features.

Hidden Layer

The Neural Network is designed with multiple layers, each playing a critical role in processing the input data. The network starts by receiving six inputs from the input layer, which represent the calculated distances between the attributes and the

Neural Network	Training Min. Loss	Training Max. Accuracy	Testing Accuracy
6-1-6+DO(0.2)+BN	1.70	29.58	13.30
6-2-6+DO(0.2)+BN	1.64	36.77	15.14
6-1024-6+DO(0.2)+BN	1.46	57.83	16.22
6-1024-512-6+DO(0.2)+BN	1.39	64.83	18.34
6-1024-512-256-6+DO(0.2)+BN	1.30	75.29	25.32
6-1024-512-256-128-6+DO(0.2)+BN	**1.24**	**80.84**	32.75
6-1024-1024-1024-1024-6+DO(0.2)+BN	1.45	58.61	53.30
6-1024-512-256-512-1024-6+DO(0.2)+BN	1.39	64.43	**64.25**

Table 4: Comparison between different Neural Network

Figure 2: Illustration of Neural Network

keyword. The hidden layers consist of 1024 neurons each. As the data passes through these layers, it is progressively refined, allowing the network to capture increasingly nuanced patterns and relationships within the data (Uzair & Jamil, 2020). The ReLU (Rectified Linear Unit), max(0*, x*), is used as the activation function between these layers (Arora, Basu, Mianjy, & Mukherjee, 2018; Nair & Hinton, 2010).

ReLU effectively introduces non-linearity into the model, which is essential for solving complex classification problems. It allows the network to pass only positive values to the next layer while setting negative values to zero, enabling the network to learn diverse patterns without the issue of vanishing gradients that can occur with other activation functions, such as the sigmoid function. Unlike the sigmoid function, which compresses outputs between 0 and 1 and is typically used in binary classification, ReLU allows the network to learn a broader range of patterns by not restricting the output values (Pratiwi et al., 2020). To further enhance the network's stability and performance, we incorporated Batch Normalization (BN) between the hidden layers (Bjorck, Gomes, Selman, & Weinberger, 2018). BN normalizes the input to each layer within a mini-batch, reducing internal covariate shift and speeding up the training process. This normalization enables the use of higher learning rates, leading to faster convergence and improved overall accuracy. Additionally, BN acts as a form of regularization, helping to mitigate overfitting by ensuring that the network's learning process is more robust.

Despite these improvements, overfitting remained a concern, particularly as indicated by the significant difference between training and test accuracy (as shown in Table 4). To address this issue, we employ Dropout with a rate of 0.2. The dropout randomly deactivates 20% of neurons during each training iteration, reducing the bias, and enhancing the generalizability of the model (Baldi & Sadowski, 2013). This technique effectively reduces overfitting, leading to improved performance on unseen data. In parallel with Dropout, we also applied L2 Regularization across the entire model. L2 regularization adds a penalty proportional to the square of the magnitude of the model parameters, encouraging the network to maintain smaller weights and avoid overfitting (G. Shi, Zhang, Li, & Wang, 2019). While L2 regularization provides a baseline level of regularization for all models, the introduction of Dropout further enhances the model's ability to avoid overfitting by promoting diversity in the learned representations.

After extensive testing and tuning, this specific network configuration, consisting of 6 input neurons, 4 hidden layers with 1024 neurons each, and 6 output neurons, using ReLU as the activation function, L2 regularization, dropout (rate 0.2), and batch normalization, was found to provide the best performance for the task, as shown in Table 4. This setup enables the network to effectively analyze the input data and make accurate classifications, whether for identifying the Chameleon or helping the Chameleon reduce suspicion during gameplay.

Output Layer

The output layer of the Neural Network is a multi-class classifier. It generates a prediction that identifies which player is most likely to be the Chameleon. The prediction is represented as one of six possible integers corresponding to the players in the game, with the model's objective being to accurately predict the player most likely to be the Chameleon.

In this model, we use Cross Entropy Loss as the loss function because it is the most effective and commonly used method for multi class classification problems (Hinton, Osin dero, & Teh, 2006). Cross Entropy Loss measures the performance of the classification model whose output is a probability value between 0 and 1. It calculates the difference between the actual label and the predicted probability distribution, penalizing predictions that are further from the true label. This loss function is particularly suitable for multi-class classification because it considers the probability distribution over all classes, ensuring that the model not only predicts the correct class but also assigns low

probabilities to incorrect classes. By minimizing Cross Entropy Loss, the model is trained to provide more accurate and confident predictions across multiple classes, making it an essential component in developing a reliable AI model for tasks like identifying the Chameleon in the game. The process of finding the intersection of these 5 attributes is similar to how the attribute intersection is determined in the "Giving Attributes as the Chameleon" part. This process results in a table showing how many attributes intersect at each keyword.

After that, we calculate attribute probabilities, particularly in the context of the "Giving Attributes as Chameleon" step. The database is refined by setting any data that is below the highest number of intersecting attributes to zero. Subsequently, the refined dataset is utilized to compute the probability table for the 5 attributes. After determining the probability of each attribute given a relevant keyword, the Naive Bayes classifier—a simple learning algorithm that utilizes Bayes' rule along with the strong assumption that the attributes are conditionally independent given the class—is employed to estimate the probability of each keyword being the secret one (Webb, Keogh, & Miikkulainen, 2010). Initially, each keyword $P(k)$ is set to $\frac{1}{16}$. For each keyword k, the model calculates the conditional probability of each attribute A_i given the keyword: $P(A_i|k)$. Using the Naive Bayes formula, the combined probability of all attributes given the keyword is determined. Specifically, the model computes (Alpaydin, 2020):

$$P(k|A_1, \ldots, A_n) = P(k) \cdot \prod_{i=1}^{n} P(A_i|k) \quad (1)$$

The initial probability will be $P(k) = \frac{1}{16}$, since we have 16 keywords, and $A_n = A_5$ since we only have 5 attributes. To avoid underflow, which occurs when a calculation gets so close to 0 that the computer treats it as 0, we take the log of both sides of the formula (Haarhoff, Kok, & Wilke, 2013):

$$\ln(P(k|A_1, \ldots, A_5)) = \ln\left(\tfrac{1}{16}\right) + \sum_{i=1}^{5} \ln(P(A_i|k)) \quad (2)$$

Guessing Keyword as Chameleon

Suppose the Humans identified exactly who the Chameleon is in the Voting phase (Voting True in Figure 1), the game comes to the Guessing keyword as Chameleon phase. In this phase, the Chameleon must guess the keyword based on all the attributes provided

After calculating the naive probabilities of the 16 keywords, the Chameleon will guess the keyword with the highest probability as the secret word for the game. If more than one key word has the same Naive Bayes probability value, the Chameleon will randomly select one among them. This approach ensures that the Chameleon makes an informed guess based on the calculated probabilities while also introducing an element of unpredictability.

Training Model

In the training phase, our best performance model will interact with five bots playing as Humans. Each bot is designed to give attributes to keywords using methods described in the next section, and the voting strategy will have strategies similar to our best performance model in the Voting section above.

Training Database

We first used the Global Vectors for Word Representation (GloVe) model to generate an at tribute list (GloVe list) (Pennington, Socher, & Manning, 2014). GloVe model captures semantic relationships between words by embedding them in a continuous vector space, where the distance between vectors reflects their semantic similarity. Words with similar meanings or contextual usage are positioned closely, enabling the identification and quantification of relationships between terms.

Using the GloVe model, we calculate the cosine similarity between each attribute in the GloVe list and each of the 16 key words

(Mikolov, Chen, Corrado, & Dean, 2013). Cosine similarity values close to 1 indicate high similarity, while values near 0 represent minimal relevance. Starting with attributes that have a cosine similarity of at least 0.50 with the keywords ensures that the attributes have at least moderate semantic similarity to the keywords, avoiding those that are too dissimilar. This threshold balances diversity and relevance, ensuring attributes remain distinct without drifting too far from the key words' meaning. We then divided these attributes into six pools based on the cosine similarity value. Table 5 represents keywords with associated cosine similarity values.

In addition, we eliminate all attributes that are either too similar to a keyword or have a singular form already present. This ensures that the remaining attributes have a distinct semantic distance from the keywords and other attributes, allowing them to provide meaningful and diverse information. The higher the pool number, the more relevant the attribute is to the keyword, indicating a stronger semantic similarity. However, classification stops at Pool 6 because cosine similarity scores above 0.80 are rare for some attribute-keyword pairs, making it impractical to create additional pools beyond this threshold.

To determine which attributes the bots provide in the database, we apply a random sampling method that uses weights based on co sine similarity scores. Words with higher co sine similarity scores will be assigned greater weights, increasing their probability of being selected from the pool. This ensures that data points are appropriately chosen according to the similarity metrics.

The weights for each pool are determined using the inverse cosine function ($\cos^{-1}$) of the cosine similarity scores, given in degrees as can be seen in Table 5:

If two attributes are identified as similar, the process is repeated to generate a new list of six random keywords. This revised list is then distributed to five bots, ensuring that the final selection benefits from a combination of algorithmic randomness and human judgment.

By using the algorithm described above, we enhance the performance and reliability of the model by conducting a random sampling process 30,000 times to generate a diverse set of samples. This extensive sampling is designed to capture various data dimensions, ensuring that the model is exposed to a wide range of scenarios and variations. Figure 3 shows how the accuracy of guessing the secret word as the

Pool	Cosine Similarity	$\cos^{-1}(x)$	Weight	Pool Percentage
1	0.50	60.0°	18.90	13.90%
2	0.55	56.3°	20.14	14.86%
3	0.60	53.1°	21.36	15.76%
4	0.65	49.5°	22.91	16.90%
5	0.70	45.6°	24.87	18.34%
6	0.75	41.4°	27.39	20.20%

Table 5: Weight of Pool

This weighting ensures that the most semantically relevant words (those in higher pools) are more likely to be chosen, reflecting their closer relationship to the keywords. By employing this method, the analysis effectively leverages the GloVe model's ability to capture nuanced semantic relationships, enabling a more precise and contextually aware interpretation of textual data.

To determine which attribute to use, we employ a two-layer random selection process.

Layer 1: Distribute Words into Pools Attributes are first distributed into different pools. Within each pool, attributes are chosen randomly, with each having an equal probability of appearing.

Layer 2: Randomly Select Pool A pool is then randomly selected based on their weights. The numerator is the weight of a pool, and the denominator is the total weight of all pools. The corresponding percentage of being selected for each pool is shown in Table 5.

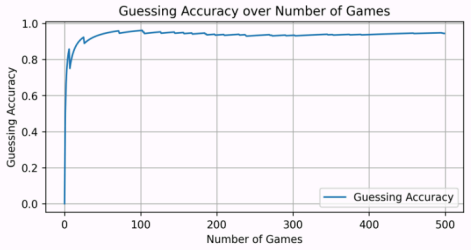


Figure 3: Chameleon Guessing Accuracy over Time

Chameleon changes over time.

The Chameleon model achieves a high guessing accuracy rate of over 92% with the training dataset, demonstrating its effectiveness in predicting keywords. Initially, the model's win rate starts low but rapidly climbs to over 90% as it gains more experience through the training process. Even after 1,000 games, the win rate keeps fluctuating slightly around this high value, indicating consistent performance. The large-scale sampling process further validates this performance, ensuring consistent and reliable accuracy across diverse data conditions.

Training Neural Network

Although we generated data from 30,000 games in the training database process, we used only the last 10,000 as training data for the neural network. The first 20,000 games were primarily used for populating the database, providing a robust foundation for the model's training. The goal was to ensure that the database captured a wide range of scenarios, player interactions, and possible outcomes, thereby enhancing the model's ability to generalize effectively.

Following this initialization, the model was trained across ten sessions, with each session consisting of 400 epochs. Figure 4 corresponds to the tenth training session, reflecting the model's performance after it had undergone significant refinement through the preceding nine sessions. This approach allowed the model to be improved progressively, minimizing loss and maximizing accuracy with each subsequent session. The rigorous training process, supported by a well initialized and diverse dataset, contributed to the model's robust performance in predicting the secret word as the Chameleon, achieving high accuracy.

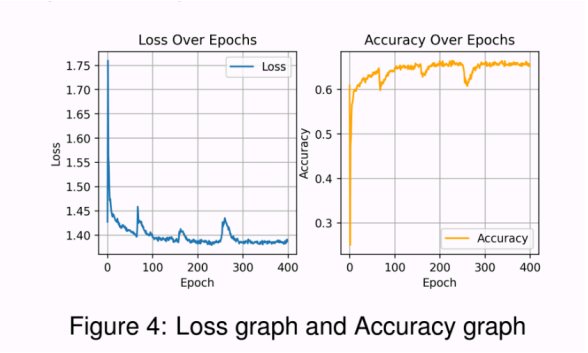


Figure 4: Loss graph and Accuracy graph

Testing Model

After the training phase, we created an additional 2,000 games for testing. In these test games, the Chameleon achieved a win rate of over 90% and was voted out in 64% of the games. The bot selects a keyword based on the rule outlined in the training section. The Chameleon's high win rate demonstrates its effectiveness in a controlled environment where the bots, due to their predictable and in formative cues, allow the Chameleon to blend in and avoid detection more easily. However, this success is expected to drop significantly in a different configuration—specifically, when five people are playing with our model. People, with their less predictable and more varied hints, introduce greater complexity and additional variables, making it more challenging for the Chameleon to remain undetected.

The moderate voting rate of 64% likely results from the stochastic nature of attribute selection among the bots, introducing a degree of randomness that may assist the Chameleon in evading detection. The interplay of these factors is depicted in Figure 5, illustrating how the Chameleon's win rate and the effectiveness of the voting process evolve under different conditions.

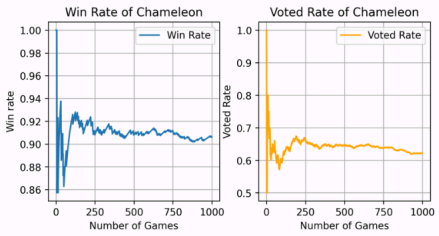


Figure 5: Chameleon Win and Voted Rates after Training

Conclusion

Developing the AI model for Chameleon Games has revealed several strengths and areas for improvement. A significant challenge was the lack of readily available, real-world datasets recorded after each game. This limitation highlights the need for future efforts to expand the dataset by collecting more real game data, enhancing the model's ability to simulate and predict player behavior more accurately.

Despite these challenges, the model demonstrates promising capabilities. The model is designed to learn and adapt after each match, allowing it to refine its accuracy and effectiveness over time continuously. With these strengths and a commitment to addressing the identified limitations, the AI model has the potential to become a robust and versatile tool for simulating and analyzing gameplay in Chameleon Games and beyond.

Additionally, the current implementation uses a simple neural network architecture. Future work should focus on exploring more sophisticated neural network models to improve the model's performance and adaptability.

Code

Source code available at https://github.com/DatNguyen2003/EAAI 2025.

Appendix A: Chameleon Game Simulation

Game Theme: Food
Secret Word: Salad

System: Gathering clues

Player 1 (Human): "coleslaw"

- **Player 2 (Human)**: "eggplant"
- **Player 3 (Human)**: "tangy"
- **Player 4 (Human)**: "servings"
- **Player 5 (Chameleon)**: "frying"
- **Player 6 (Human)**: "tomato"

System: Voting

- **Player 1**: "I think Player 5 is the Chameleon."
- **Player 2**: "I think Player 5 is the Chameleon."
- **Player 3**: "I think Player 5 is the Chameleon."
- **Player 4**: "I think Player 5 is the Chameleon."
- **Player 5 (Chameleon)**: "I think Player 4 is the Chameleon."
- **Player 6**: "I think Player 5 is the Chameleon."

Chameleon Guess: "My guess for the secret word is: Salad"
Outcome: Correct guess! Chameleon es cape!

References

Alpaydin, E. (2020). *Introduction to machine learning* (4th ed.). MIT Press.

Arora, R., Basu, A., Mianjy, P., & Mukherjee, A. (2018). *Understanding deep neural networks with rectified linear units.* arXiv:1611.01491 [cs.LG].

Baldi, P., & Sadowski, P. (2013). Understanding dropout. In *Proceedings of the 27th international conference on neural information processing systems - volume 2* (pp. 2814–2822). Red Hook, NY, USA: Curran Associates Inc

Bjorck, J., Gomes, C., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 7705–7716). Red Hook, NY, USA: Curran Associates Inc.

Brown, N., & Sandholm, T. (2019). Super

human AI for multiplayer poker. *Science*, *365*(6456), 885–890.

Chowdhury, G. (2010). *Introduction to modern information retrieval* (3rd ed.). Facet Publishing.

De Deyne, S., Navarro, D. J., Perfors, A., Brysbaert, M., & Storms, G. (2019). The “small world of words” English word association norms for over 12,000 cue words. *Behavior Research Methods*, *51*(3), 987–1006.

Guresen, E., & Kayakutlu, G. (2011). Definition of artificial neural networks with comparison to other networks. *Procedia Computer Science*, *3*, 426–433.

Haarhoff, L. J., Kok, S., & Wilke, D. N. (2013). Numerical strategies to reduce the effect of ill-conditioned correlation matrices and underflow errors in kriging. *Journal of Mechanical Design*, *135*(4), 044502.

Hasan, B. A. S., & Gan, J. Q. (2012). Hang man BCI: An unsupervised adaptive self paced brain-computer interface for playing games. *Computers in Biology and Medicine*, *42*(5), 598–606.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2Vec model analysis for semantic similarities in English words. *Procedia Computer Science*, *157*, 160– 167.

Martinez, L., Gimenes, M., & Lambert, E. (2023). Video games and board games: Effects of playing practice on cognition. *PLoS ONE*, *18*(3), e0283654.

Mercier, M., & Lubart, T. (2021). The effects of board games on creative potential. *The Journal of Creative Behavior*, *55*(3), 875–885.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space.* arXiv:1301.3781 [cs.CL].

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltz mann machines. In *Proceedings of the 27th international conference on machine learning* (pp. 807—814). Madison, WI, USA: Omnipress.

Ornaghi, V., Brockmeier, J., & Gavazzi, I. G. (2011). The role of language games in children’s understanding of mental states: A training study. *Journal of Cognition and Development*, *12*(2), 239– 259.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics.

Pratiwi, H., Windarto, A., Susliansyah, S., Aria, R., Susilowati, S., Rahayu, L., . . . Rahadjeng, I. (2020). Sigmoid activation function in selecting the best model of artificial neural networks. *Journal of Physics: Conference Series*, *1471*, 012010.

Richards, M., & Amir, E. (2007). Opponent modeling in Scrabble. In *Proceedings of the 20th international joint conference on artificial intelligence* (pp. 1482—1487). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Shi, G., Zhang, J., Li, H., & Wang, C. (2019). Enhance the performance of deep neural networks via L2 regularization on the input of activations. *Neural Processing Letters*, *50*, 57–75.

Shi, L., Chen, Y., Lin, J., Chen, X., & Dai, G. (2023). A black-box model for predicting difficulty of word puzzle games: a case study of Wordle. *Knowl. Inf. Syst.*, *66*(3), 1729–1750.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., . . . Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., . . . Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go

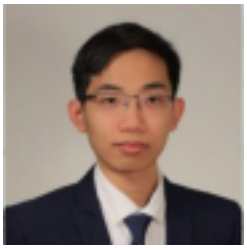
through self play. *Science*, *362*(6419), 1140–1144.

Uzair, M., & Jamil, N. (2020). Effects of hidden layers on the efficiency of neural networks. In *2020 IEEE 23rd international multitopic conference (INMIC)* (pp. 1–6).

Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Na¨ıve bayes. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of machine learning* (pp. 713–714). Boston, MA: Springer US.

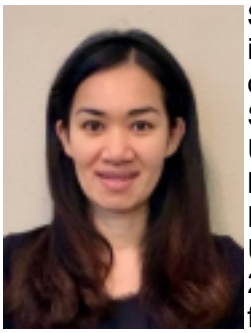
Tri Dang is a senior Computer Science and Mathematics double major at DePauw University. His research interests include Data Mining, Machine Learning, and Data Science. He's also open to new aspects of research. He will start his Ph.D. studies in Fall 2025.

Dat Nguyen is a senior at DePauw University double majoring in Computer Science and Mathematics. His main area of interest is Computer Science, in specific Neural Networks, Large Language Models and Natural Language Processing.



Hieu Tran is a Ph.D. student in Computer Science at Purdue University. His research focuses on creating artificial agents that establish long-term connections with individuals, aiming to enhance their well-being, health, and learning.

Brian Howard is an Associate Professor of Computer Science at DePauw University, where he has taught since 2002. He earned his Ph.D. in Computer Science from Stanford University in 1992. His research interests include programming languages, logic, and educational tools to help students learn about functional programming and proofs.



Sutthirut Charoenphon is an Assistant Professor of Mathematical Sciences at DePauw University. She earned her Ph.D. in Applied Mathematics from the University of Memphis in 2020. Her research focuses on partial differential equations, wave equations, and control theory, and the development of mathematical models and computational techniques in these areas.

Adapting BERT for ‘Apples to Apples’ Gameplay

Authors

Arikka Cherniwchan (MacEwan University; cherniwchana7@mymacewan.ca) Austin Countaway (MacEwan University; countawaya@mymacewan.ca) Chunyang Ding (MacEwan University; dingc4@mymacewan.ca)
Brandon Funk (MacEwan University; funkb22@mymacewan.ca)
Calin Anton (MacEwan University; antonc@macewan.ca)
DOI: 10.1145/3774399.3774404

Introduction

“Apples to Apples” is a word association game where players match red noun cards with a given green adjective card in the hopes of winning the round through the selection of their card by the round judge. There are many possible selection criteria for determining which red card to play in a round. Combinations may focus solely on the direct association between a red and green card, on the humour elicited by a card pair, or through a wide variety of other selection criteria. The game’s difficulty comes from the anticipation and matching of the card played by a player to the criterion of the current round judge. Our aim was to create an agent capable of playing “Apples to Apples,” able to discover the current judge criterion and play the card in hand that best matches that criterion. To this end, we decided to explore the use of BERT pretrained models and evaluate their viability in word game agents. In this study, we leveraged fine tuned BERT models and developed a Naive Bayes classifier to simulate and predict judge personalities in “Apples to Apples.” Beyond these main contributions, this work also involved significant efforts in creating supporting tools for both training and testing processes. Specifically, we manually annotated nearly 1000 card pairs using a custom annotation program to reduce human error and fatigue and developed an environment testing tool to ensure platform consistency during training and testing. These contributions highlight our focus on both data quality and system robustness.

Background and Related Work

Since the introduction of transformers (Vaswani et al., 2017), they have been implemented and shown high performance in a variety of Natural Language Processing (NLP) tasks (Patwardhan, Marrone, & Sansone, 2023), becoming one of the go to architectures in NLP (Lin, Wang, Liu, & Qiu, 2022)). BERT (Devlin, Chang, Lee, & Toutanova, 2019) added greater functionality and performance to the base transformer on a variety of NLP tasks as well (Koroteev, 2021). BERT has shown strong performance in comparison to traditional machine learning text classification (Garrido-Merchan, Gozalo Brizuela, & Gonzalez-Carvajal, 2023) and has been used with high performance in tasks related to sentiment analysis (Sayeed, Mohan, & Muthu, 2023). Zhang et al. (Zhang et al., 2020) extend the application of BERT to semantic association, proposing a model, SemBERT, offering improved capability to the base BERT model in the areas of reading comprehension and language inference. Reimers and Gurevych (Reimers & Gurevych, 2019) offer another extension, SBERT, allowing for greater efficiency in sentence comparisons in a much shorter time. Previous success found using BERT in NLP and semantic association tasks offers a strong motivation for adapting a similar pre-trained model for use in an “Apples to Apples” agent. While there has not yet been a published study adapting BERT to this purpose, a few instances of using pre-trained models for word games require comparable semantic similarity analysis. Koyyalagunta et al.

(Koyyalagunta, Sun, Draelos, & Rudin, 2021) implemented an algorithm to generate clues for use in the game "Codenames" using various embedding methods, one of which was BERT. Although BERT underperformed compared to other word embedding models, their work highlights potential avenues to explore in adapting a pre-trained model to "Apples to Apples." The work of Koyyalagunta et al. (Koyyalagunta et al., 2021) also offers some insight into potential areas of improvement when using a BERT pre-trained model with respect to contextual embeddings and fine-tuning. To adapt BERT to match the "Codenames" scoring metric, their team used an average of the BERT contextual embeddings for each word as that word's new embedding. Contextual embeddings for BERT perform similarly to non-contextual models for large training sets but show increased performance with ambiguous words and words unseen in training (Arora, May, Zhang, & Re´, 2020). Maintaining the contextual embeddings during implementation in "Apples to Apples" may allow for a more versatile agent that can better work with untrained data. Additionally, their use of a pre-trained model as the word embedder without further task-specific fine-tuning may have led to suboptimal results in the model performance. Fine-tuning improves task performance, though excessive divergence from training and test sets may lead to suboptimal results as well (Zhou & Srikumar, 2022). Ensuring sufficient fine-tuning to increase task specificity while avoiding excessive divergence between training and test sets may increase agent performance. The winning card in "Apples to Apples" is often the one that the judge finds funny in some way. Playing this card involves detecting and ranking humorous associations between card pairs with respect to the judge's sense of humour. BERT has shown success in the realm of humour detection using a generic pre-trained model and a corpus of annotated tweets (Mao & Liu, 2019), suggesting a model may be used for both semantic association and humour with changes to fine-tuning to reflect the judge. A particular challenge in creating a playing agent is adapting to previously unseen cards or card pairings. Zero-shot learning is one method that may ameliorate this problem. The capacity for BERT to understand context may allow for the incorporation of zero-shot learning, which has shown successful implementation in the word game "Taboo" (Isaak, 2022). Contextual understanding of each card would allow BERT to make associations with Adjective-Noun pairs, even if that pair had not been seen in training.

Agent Objectives

In "Apples to Apples," a pivotal skill is the player's ability to adapt to the current round judge, playing card pairs that best match the theme or judging method used by that judge in previous rounds. The creation of an "Apples to Apples" playing agent, therefore, necessitates mimicking the adaptation shown by human players to achieve high performance. Modelling various personalities that the judges may display offers one potential starting point to facilitate this adaptation. Refining the model by selecting common, broad personalities, collecting data for those personalities, and training the model on that data would help in the creation of archetypes to serve as a foundation for potential personality models. After the creation of the personalities, the agent would need some method of determining which personality archetype a judge best matches during gameplay. Modelling and predicting user personality has been used in a variety of applications to achieve high success, such as advertising applications (Shumanov, Cooper, & Ewing, 2022). Their team focused on Big Five personality traits in the realm of marketing, showing an overall increased effectiveness of their advertising program after incorporating personality prediction. The use of deep learning (An & Levitan, 2018) and BERT-based (Lucky, Zain Nabiilah, Hendrik Jeremy, & Suhartono, 2023) models have shown similar success in predicting Big Five personality traits, further motivation for incorporating personality modelling and prediction in other platforms. A core facet of our approach was the belief that the incorporation of judge personality detection to an "Apples to Apples" agent would likely rely on more game-specific personality categories to apply to as many players as possible. The easiest personality

to implement is a semantic association judge, favouring card pairs that directly relate. BERT models for semantic as sociation. (A. Rodriguez & Merlo, 2020) and (Delmonte & Busetto, 2022) have shown that comparisons using word embeddings and co sine similarity replicate the human association between words well, offering a good basis for a primary personality model. We reasoned that further judge archetypes would achieve success in focusing on various forms of hu mour due to the large part that humour plays in judging the winner of any round. Humour is a complex and adapting field, making it difficult to create an agent that can reflect the nuanced types found in humans. Nevertheless, nine broad categories determined through humour style questionnaires (Heintz & Ruch, 2019) offered a starting point for judging archetypes that can be subsequently narrowed down based on feasibility. The primary identified categories were fun/affiliative (often relying on shared experience), benevolent humour/self enhancing (laughing at self/life situations in a good-natured manner), sarcasm/aggressive (ridicule or mocking), nonsense (absurd or surreal), wit, irony, satire, cynicism, and self defeating. Creation of datasets for humour focused training would likely feature considerable overlap between these styles, though the work performed regarding the specific humour types of wit using wordplay (Palma Preciado, Sidorov, & Preciado, 2022) and irony (Potamias, Siolas, & Stafylopatis, 2020) provide insights and a starting point in the creation of specific archetypes for use in judge personality detection. The next logical step after creating our archetypes was to create a method to determine which archetype a judge most likely falls under. Naive Bayes classifiers offered one potential approach, having seen success in personality classification from text (Pratama & Sarno, 2015) and offering a simple but powerful classifier (Berrar, 2019).

Methodology

Data Preparation

To ensure high-quality training data, we developed a custom annotation program (Figures 1 and 2) for manual training of card combinations. The program was designed to be user friendly, featuring progress saving and resuming capabilities. Although initially intended for use by additional testers, time constraints limited its deployment. Moreover, to maintain consistency across different platforms during training and testing, we created an environment testing tool (Figure 3). This tool ensures compatibility by checking operating system details, Python versions, software dependencies, available resources, and hardware acceleration options. Its intuitive interface allows users to quickly identify any system in compatibilities, enabling smoother execution of training and testing processes.

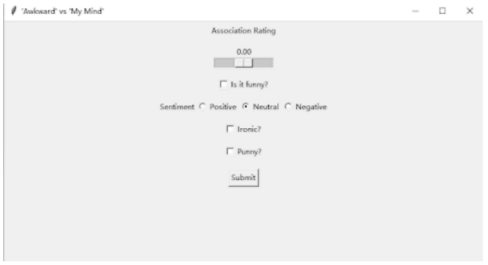


Figure 1: *Screenshot of the custom annotation program for manual card labeling. The interface is designed for ease of use, supporting progress saving and resuming capabilities.*

Figure 2: *Screenshot of the custom annotation program for manual card labeling.*

Pre-trained Models

For this study, we utilized pre-trained BERT models available via the Hugging Face

library. These models were selected for their state-of-the-art performance in natural language understanding tasks.

Fine-tuning Process

To adapt the pre-trained models to the "Apples to Apples" gameplay, fine-tuning was performed on a custom dataset of 800+ labeled pairings could be judged; second, to evaluate whether a BERT model would be capable of reflecting a more niche and difficult to define personality in the form of irony or puns/wordplay combinations, which were manually annotated by our team. This dataset captured the nuances of four distinct judge personalities (semantic association, sentiment analysis, irony detection, and pun detection).

Figure 3: *Screenshot of environment checker.*

Dataset Similarity and Isolation

While the training dataset overlapped thematically with the test set (e.g., similar topics and card categories), we strictly held out the test data during training to ensure no data leakage occurred.

Pre-trained vs Fine-tuned

Pre-trained BERT models provided general language understanding capabilities, while fine-tuning adjusted the models to prioritize personality-specific card selection tasks, enabling them to better simulate judge behavior

Selection of Personality Subtypes

The general ‘archetypal’ personalities chosen for implementation were Semantic Association, Sentimental, and Humorous. A judge with the Semantic Association personality would prefer red cards that had a close conceptual or contextual relationship with their given green card, creating a coherent, logical pairing. A judge with a Sentimental personality favours combinations that elicit a strong positive or negative emotional sentiment. The Humorous archetypal personality would look for any combinations that were considered generally humourous. These three personality types serve as the foundation for the typical range of personality traits one would expect to encounter in “Apples to Apples.” We divided the humour archetype into two additional niche types, in the form of ‘Irony’ and ‘Puns/Wordplay.’ The motivation for this sub division was twofold: first, due to the subjective nature of different types of humour, to create more specific categories in which card

Choice of Base Models and Datasets

Four pre-trained models were selected to represent the judge personality types, each trained on datasets relevant to its respective personality. These models serve as the ‘base’ for the Semantic Association, Senti mental, Puns/Wordplay, and Ironic personal ities. The models utilized for each personality type are ”all-mpnet-base-v2” (Reimers & Gurevych, 2019), ”sentiment-roberta-large english” (Hartmann, Heitmann, Siebert, & Schamp, 2023), ”roberta-small-pun-detector v2”, and ”twitter-roberta-base-irony” (Barbieri, Camacho-Collados, Espinosa Anke, & Neves, 2020), respectively. All models were sourced from Hugging Face, a well-established machine learning community which offers a wide range of models, datasets, and applications. These four specific models were chosen because they offer a general yet comprehensive foundation for analyzing text in relation to each personality type and the ability to provide relevant scores for each. An additional dataset for irony was used from gimmaru/tweet eval irony to further train our irony model on pre labelled data. As all data provided so far to the models was based on more general pieces of text, we created a program to generate randomly paired card combinations to manually rank each based on our interpretations of the four personality types. This served not only to train each model on approximately 800 points of data in the specific context of “Apples to Apples" but also to provide subjective data from human players, which was viewed as especially important for detecting puns and

irony.

Naive Bayes Approach

We attempted to incorporate a Naive Bayes classifier to enhance the ability to classify judge personalities based on their exhibited card selections. Naive Bayes was chosen due to its simplicity and efficiency, granting our model the ability to analyze multiple features simultaneously while attempting to determine a specific judge's personality type. The features provided to the classifier were the individual ratings generated by each personality model for a given card combination, aggregated into a single feature vector consisting of five ratings: Semantic Association, Positive Sentiment, Negative Sentiment, Pun, and Irony. The feature vector, after being created by a certain personality model, was labelled with the name of that personality. These labelled feature vectors were used to train a Naive Bayes classifier, which would then be used to predict the most likely personality type of future card combinations. The feature vectors of the winning red and green cards chosen by the judge were aggregated for every round where that judge appeared. The goal was to have a changing overall feature vector that the agent would use to predict the most likely personality that a given judge possesses - reasoning that the card combinations from certain personalities would show patterns over time.

Card Combination Ranking and Selection

The approach to ranking and selecting the most suitable card combinations for each judge went as follows. Both red and green cards were tokenized using the tokenizer functions and embedding layers associated with each personality model. The obtained encodings were then processed by the model to produce logits, whose raw scores indicate how strongly the card combination aligns with a specific personality type. The model architecture necessitated treating each personality output as a multiclass problem rather than binary. Therefore, logits were then passed to a SoftMax function, which converted them into probabilities indicating how likely (or not) a given combination adhered to a particular personality type. Once the probabilities for each personality trait were determined, the card combinations were ranked based on the highest probability of matching the target personality type. The combination with the highest confidence score was chosen as the best match for that judge, as high confidence scores indicate the best possible alignment with the desired personality type in each hand. The Semantic Association model followed a slightly different process due to using SBERT architecture. Instead of using logits and the Soft Max function, the model computes the cosine similarity between the green and red card embeddings. This measures how closely related the two cards are in vector space, with higher similarity scores indicating a stronger semantic connection. The combination with the highest cosine similarity score is then selected.

Testing

Testing the "Apples to Apples" playing agent proved to be time intensive and difficult due to inherent subjectivity in the game. The approach to testing, therefore, consisted of both human and automated testing to generate further data on the viability of the methods used. The first testing came from the datasets used for training. Splitting the datasets with training and testing allowed for quick and superficial analysis of the overall performance of the models on labelled data. Human testing consisted of multiple rounds of gameplay using a hand of seven red cards and a single green card. The tester then selected the red card from their hand that would best correspond to each of the personality models - association, positive sentiment, negative sentiment, irony, and pun. The same hands of red cards and green cards given to the human tester were then given to the agent, where each of the personalities would choose one card from the hand to match the green card. The amount of human/playing agent agreement on card selection was then tallied. Automated testing provided another challenge in that finding an agent that portrayed a certain personality type would be needed but not available. Therefore,

automated testing did not allow for the evaluation of personality effectiveness. Instead, we focused testing on the overall viability of an agent playing “Apples to Apples” by examining situations where the agent should win. To this effect, we gathered additional testing models with similar personalities but pre-trained on different data sets to act as the judges each round. Each round then had eleven players total - the five agent personalities, five complements (not fine-tuned), and random. The winners of each round and the amount that each personality won when their complement was the judge was tallied. Finally, a control test was conducted to examine the overall effect that the randomness of hand assignment would have on the outcomes in “Apples to Apples.” There are often situations where there aren't any strong red cards to play for a given green card for the round judge personality. To examine how much this would impact an agent over time, similar automated testing was performed, but with the same personalities instead of complement models, i.e. there were two copies of each personality each round for a total of ten players, with the judge rotating through distinct personalities. Again, the winners of each round and the amount that each personality won when their duplicate was the judge were tallied.

Results

The result of the dataset testing (Table 1) gave the accuracies obtained by the sentiment, irony, and pun models.

Personality	Testing Accuracy
Association	-
Sentiment	0.84
Irony	0.69
Pun	0.92

Table 1: The association model was not able to undergo the same testing due to the continuous nature of the training label.

Human testing (Table 2) on 48 data points showed some differences in pre-trained and fine-tuned models. The pre-trained association model showed 23% accuracy in selecting the same card as a human tester, compared with 29% for the fine-tuned model. Positive sentiment showed relatively similar results at 21% and 23% for the pre-trained and fine-tuned models, respectively, while negative showed 25% and 35%. Irony (13%, 10%) and pun (17% for both) showed comparatively lower accuracy scores.

Personality	Amount Correct	
	Pre-trained	Fine-tuned
Association	11	14
Positive	10	11
Negative	12	17
Irony	6	5
Pun	8	8

Table 2: Human Expert Round Testing

Automated testing (Table 3) revealed that after 1000 rounds of testing, the complement judge accounted for 34% of the association model wins, 28% of the positive sentiment wins, 23% of the negative sentiment wins, 21% of the irony wins, and 15% of pun wins. Further, the association model won 12% overall, the positive sentiment 11%, 10% for negative sentiment, 8.9% for irony, and 9.4% for pun. The control player, which played a random card every round, won 8.3% of the rounds. These total 594 rounds. The remaining 406 rounds were won by the complement models.

Personality	Rounds Won	
	Complement Judge	Total
Association	39	115
Positive	31	109
Negative	24	104
Irony	19	89
Pun	14	94
Random	-	83

Table 3: Automated Game Round Testing

Personality	Rounds Won	
	Complement Judge	Total
Association	25	72
Positive	27	62
Negative	26	61
Irony	12	61
Pun	9	53

Table 4: Automated Game Round Testing - Control

The control automated testing (Table 4) revealed that after 500 rounds of testing, the complement judge accounted for 35% of the association model wins, 44% of the positive sentiment wins, 43% of the negative sentiment wins, 20% of the irony wins, and 17% of pun wins. Further, the association model won 14% overall, the positive sentiment 12%, 12% for negative sentiment, 12% for irony, and 10% for pun of a total of 309 rounds. The remaining 191 rounds were won by the complement models.

Naive Bayes data showed an accuracy of about 25% for determining the personality based on a given feature vector of a ranked card combination.

Discussion

Our results show that while there is better than random selection in the use of BERT based models to play “Apples to Apples,” further refinement in methods for reflecting human personality and judgement is needed. Testing performed on items set aside from the team-made training dataset shows that there is relatively good prediction accuracy by the models for detecting the presence of their respective personalities in a red and green card combination. The effectiveness of the models dropped quite a bit when asked to choose between multiple options, however. Further, some models showed much higher capability in card selection than others. Random selection would result in an overall success rate of around 14%, which the fine-tuned association and sentiment models were able to defeat but which the irony and pun models were not. This may reflect the more subjective nature of irony and pun detection and may indicate the need for a more significant number of human testers to capture this nuance. One notable limitation of our work was the under-utilization of the annotation and environment testing tools. These tools were specifically designed to be distributed to external testers for broader data collection, but time constraints restricted their use to the internal team. In future work, refining these tools for wider deployment could facilitate larger-scale testing and validation efforts The automated testing shows that despite having similar personalities, the large amount of chance involved in “Apples to Apples" may still result in a win about 34% of the time. This is supported by the results of the Automated Game Round Testing Control group, where the highest contribution of ex act copies only accounted for a maximum of 44% of any model’s wins. This suggests that if a player is dealt a hand that does not have any good cards for a given judge’s personality, they may not be able to win the round despite knowing precisely what that personality is. The problem compounds as there is only one new card drawn per round, and if a player cannot recycle their hand, they are effectively stuck. The result is a difficult situation for human players to overcome, reflecting the results of a playing agent attempting the same. The automated testing further revealed poor performance from the irony and pun players, even for the control. This likely stems from the relatively rare nature of a truly ‘ironic’ or ‘punny’ combination, resulting in an almost random selection from the two models. Further adaptability of a playing agent would be needed to find success moving forward - such as having alternative, broader personalities in the humour category to revert to should the more niche personalities (e.g. irony and pun) be unable to find a satisfactory match. Incorporating additional human testers would allow for greater nuance capture for niche personality types. An additional benefit would be to ask testers to perform the same task as the models and select the card that best matches each model. This would give a direct comparison between the human player and the playing agent based on criteria selected by a human

judge. Finally, Naive Bayes showed limited success with our given implementation, likely due to similar confounding principles as above. The feature vector for any one card combination would have high variance, and only with aggregation from multiple rounds would a pattern emerge in the rankings seen from models. Our implementation trained on single card combinations, and therefore each label was associated with a high degree of variance in each feature. Future implementations would likely benefit from training the Naive Bayes classifier on aggregate data instead.

Conclusion

The creation and implementation of an "Apples to Apples" playing agent poses significant challenges, many of which are insurmountable due to the chance involved in the red cards provided to players. Different personality models showed varying success depending on how niche their selection criteria were, with broad selection categories showing better than random chance of correctly identifying the ideal card from a hand for a given judge. Future work in the creation of playing agents for word games would need a greater degree of focus on alternative selection criteria based on the current hand rather than an ideal personality match.

References

An, G., & Levitan, R. (2018). Lexical and acoustic deep learning model for personality recognition. In *Interspeech 2018* (pp. 1761–1765). doi: 10.21437/ Interspeech.2018-2263

A. Rodriguez, M., & Merlo, P. (2020, November). Word associations and the distance properties of context-aware word em beddings. In R. Fernandez & T. Linzen ´ (Eds.), *Proceedings of the 24th conference on computational natural language learning* (pp. 376–385). Online: Association for Computational Linguistics. Retrieved from https://aclanthology .org/2020.conll-1.30/ doi: 10 .18653/v1/2020.conll-1.30

Arora, S., May, A., Zhang, J., & Re, C. (2020, July). Contextual embeddings: When are they worth it? In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 2650–2663). Online: Association for Computational Linguistics. Retrieved from https://aclanthology .org/2020.acl-main.236/ doi: 10 .18653/v1/2020.acl-main.236

Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., & Neves, L. (2020, November). TweetEval: Unified bench mark and comparative evaluation for tweet classification. In T. Cohn, Y. He, & Y. Liu (Eds.), *Findings of the association for computational linguistics: Emnlp 2020* (pp. 1644–1650). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.finding s-emnlp.148/ doi: 10.18653/v1/2020.findings-emnlp.14 8

Berrar, D. (2019). Bayes' theorem and naive bayes classifier. In S. Ranganathan, M. Gribskov, K. Nakai, & C. Schonbach ¨ (Eds.), *Encyclopedia of bioinformatics and computational biology* (p. 403-412). Oxford: Academic Press. Retrieved from https://www.sciencedirect .com/science/article/pii/B978012809 6332204731 doi: https://doi.org/10.1016/B978-0-12-8 09633-8.20473-1

Delmonte, R., & Busetto, N. (2022, June). Measuring similarity by linguistic features rather than frequency. In H. Bunt (Ed.), *Proceedings of the 18th joint acl - iso workshop on interoperable semantic annotation within lrec2022* (pp. 42–52). Marseille, France: European Language Resources Association. Re trieved from https://aclanthology .org/2022.isa-1.6/

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT:

Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171– 4186). Minneapolis, Minnesota: Association for Computational Linguistics. Re trieved from https://aclanthology .org/N19-1423/ doi: 10.18653/v1/ N19-1423

Garrido-Merchan, E. C., Gozalo-Brizuela, R., & Gonzalez-Carvajal, S. (2023, Apr.). Comparing bert against traditional machine learning models in text classification. *Journal of Computational and Cognitive Engineering*, *2*(4), 352–356. Retrieved from https:// ojs.bonviewpress.com/index .php/JCCE/article/view/838 doi: 10.47852/bonviewJCCE3202838

Hartmann, J., Heitmann, M., Siebert, C., & Schamp, C. (2023). More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Market ing*, *40*(1), 75-87. Retrieved from https://www.sciencedirect.com/scien ce/article/pii/S0167811622000477 doi: https:// doi.org/10.1016/j.ijresmar.2022.05.0 05

Heintz, S., & Ruch, W. (2019). From four to nine styles: An update on individual differences in humor. *Personality and Individual Differences*, *141*, 7-12. Retrieved from https://www.sciencedirect .com/science/article/pii/S019188691 8306421 doi: https:// doi.org/10.1016/j.paid.2018.12.008

Isaak, N. (2022, 06). *A zero-shot classification approach for a word-guessing challenge.* doi: 10.48550/arXiv.2206.13099

Koroteev, M. (2021, 03). *Bert: A review of applications in natural language processing and understanding.* doi: 10.48550/ arXiv.2103.11943

Koyyalagunta, D., Sun, A., Draelos, R. L., & Rudin, C. (2021, September). Playing codenames with language graphs and word embeddings. *J. Artif. Int. Res.*, *71*, 319–346. Retrieved from https://doi .org/10.1613/jair.1.12665 doi: 10.1613/jair.1.12665

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111-132. Retrieved from https://www.sciencedirect .com/science/article/pii/ S2666651022000146 doi: https:// doi.org/10.1016/j.aiopen.2022.10.001

Lucky, H., Zain Nabiilah, G., Hendrik Jeremy, N., & Suhartono, D. (2023, Feb.). A three-order ensemble model for user level big five personality prediction on twitter dataset. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(2), 283–292. Retrieved from https://www.ijisae.org/ index.php/IJISAE/article/ view/2630

Mao, J., & Liu, W. (2019). A bert based approach for automatic humor detection and scoring. In *Iber lef@sepln.* Retrieved from https:// api.semanticscholar.org/ CorpusID:199448318

Palma-Preciado, V. M., Sidorov, G., & Preci ado, C. P. (2022). Assessing wordplay pun classification from JOKER dataset with pretrained BERT humorous models. In G. Faggioli, N. Ferro, A. Han bury, & M. Potthast (Eds.), *Proceedings of the working notes of CLEF 2022 - conference and labs of the evaluation forum, bologna, italy, september 5th - to - 8th, 2022* (Vol. 3180, pp. 1828–1833). CEUR-WS.org. Retrieved from https://ceur-ws.org/Vol-3180/paper-1 42.pdf

Patwardhan, N., Marrone, S., & Sansone, C. (2023, 04). Transformers in the real world: A survey on nlp applications. *Information*, *14*, 242. doi: 10.3390/ info14040242

Potamias, R. A., Siolas, G., & Stafylopatis, A. . G. (2020, Dec 01). A transformer-based approach to irony

and sarcasm detection. *Neural Computing and Applications*, *32*(23), 17309-17320. Retrieved from https://doi.org/10.1007/s00521-020-05102-3 doi: 10.1007/s00521-020-05102-3

Pratama, B. Y., & Sarno, R. (2015). Personality classification based on twitter text using naive bayes, knn and svm. In *2015 international conference on data and software engineering (icodse)* (p. 170-174). doi: 10.1109/ICODSE.2015.7436992

Reimers, N., & Gurevych, I. (2019, November). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 3982–3992). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1410/ doi: 10.18653/v1/D19-1410

Sayeed, M. S., Mohan, V., & Muthu, K. S. (2023). Bert: A review of applications in sentiment analysis. *HighTech and Innovation Journal*. Retrieved from https://api.semanticscholar.org/CorpusID:264954476

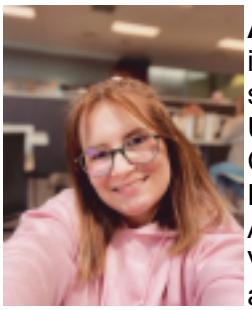
Shumanov, M., Cooper, H., & Ewing, M. (2022, Jan). Using ai predicted personality to enhance advertising effectiveness. *European Journal of Marketing*, *56*(6), 1590–1609. Retrieved from https://doi.org/10.1108/EJM1220190941 doi: https://doi.org/10.1108/EJM1220190941

Vaswani, A., Shazeer, N., Parmar, N., Uszkor eit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st inter national conference on neural information processing systems* (p. 6000–6010). Red Hook, NY, USA: Curran Associates Inc.

Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., & Zhou, X. (2020, Apr.). Semantics-aware bert for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 9628-9635. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/6510

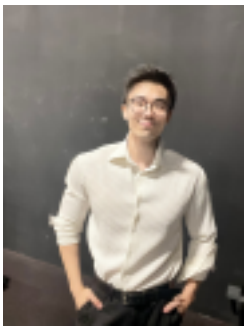
doi: 10.1609/aaai.v34i05.6510

Zhou, Y., & Srikumar, V. (2022, May). A closer look at how fine-tuning changes BERT. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1046– 1061). Dublin, Ireland: Association for Computational Linguistics. Re trieved from https://aclanthology.org/2022.acl-long.75/ doi: 10.18653/v1/2022.acl-long.75

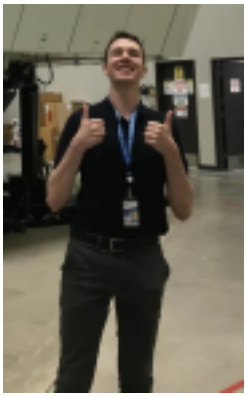


Arikka Cherniwchan is an undergraduate student at MacEwan University, majoring in Computer Science. Her interests include AI, Machine Learning, Video Game Design and Development, and Software Design and Development.

Austin Countaway is an undergraduate Computer Science student at MacEwan University. His interests include natural language processing and emergent behaviour in multi-agent systems using artificial intelligence, as well as video game development and script writing.



Chunyang Ding is an undergraduate Computer Science student at MacEwan University, with a minor in Statistics. He is interested in AI and data-driven methods, and was recognized at DataFest with the Best Use of External Resources award.



Brandon Funk is a recent graduate of MacEwan University. He is currently working as a developer at an Edmonton startup, where he's part of a project that brings together his interests in aviation and software development.

Calin Anton holds an MSc from the University of Bucharest, Romania and a Ph.D. from the University of Alberta, Canada, both in Computer Science. He has been teaching in different roles at postsecondary institutions for more than 25 years. For the last 15 years, he has taught at MacEwan University in Edmonton, Canada, where he is currently an associate professor of Computer Science. His current interests reside in Computer Security and Artificial Intelligence.

Interpolating Humour – Can Lines Be Funny?

Authors

Oscar De Leon (MacEwan University; deleono@mymacewan.ca)
Isaac McCracken (MacEwan University; mcrackeni@mymacewan.ca)
Kevin Ulliac (MacEwan University; ulliack@mymacewan.ca)
Calin Anton (MacEwan University; antonc@macewan.ca)
DOI: 10.1145/3774399.3774405

Introduction

The use of AI to play games has a long history. Some of the earliest examples were deterministic, fully observable games with clearly defined rules and possible game states. AI has proven best in games of this nature. Games such as checkers are considered solvable (Schaeffer et al., 2007) because there is always an optimal move to be made at any given point in the game, which means an AI agent can mathematically learn the best move.

On the opposite side of the spectrum, there are games like Apples to Apples, where AI is not so easily applied since there are elements that cannot be straightforwardly quantifiable and thus are not amenable to mathematical approaches. The goal of Apples to Apples is to satisfy the preferences of the judging player by playing the ”best” card of their seven red cards - nouns, which should match the judge’s green card - an adjective. The game involves dealing seven red cards from a randomly shuffled deck to each player. Each round, a new judge is chosen to draw a green card, whereby all other players will play one red card to match the judge’s green card best. The judge chooses a winning red card based on their own criteria of what is the “best” red card in play. The judge awards one point to the chosen winner, and this cycle continues until one person wins the total number of points needed to win the game. Apples to Apples is an interesting case because it involves three factors that make it far different from games like chess: randomness, use of natural language, and subjectivity. Games with any of these elements are challenging because simulating subjective preferences and determining the “best” choice to make is complex and difficult to quantify.

Research Question

Our goal is to create an Apples to Apples AI agent that can quickly and effectively learn the preferences of its opponent judges to win more games. For this, the AI agent must be able to model and appeal to various kinds of criteria, such as subjective measures like hu mour. We also want our AI agent to have its own unique preferences and to make decisions that are logically sound and mimic human creativity and humour in some way. Human creativity and humour are exceptionally difficult to model but are core attributes of these types of multiplayer word games. This leads us to some very difficult technical questions. Are there ways to quantify human elements like creativity and humour? Are there existing models that we can leverage and/or modify? A review of existing models did not appear to make full use of the vector dimensionality that encodes the semantic meaning of words. Therefore, to address the questions above, we hypothesize that using Linear Interpolation on word embedding vectors on an element-by-element basis provides an effective estimate of each opposing player’s preferences and, therefore, is an effective model for AI learning in Apples to Apples.

Problems to Solve

To address the research questions, we need

to solve several challenging problems. The first major challenge in creating an AI agent for Apples to Apples is representing words as numerical values - word embeddings. By encoding words into multidimensional vectors, an agent can calculate the “best” red card to choose as both a judge and a player. The agent must model the preferences of all op ponents and select cards based on what is in their hand and in play. We also need a way for players to choose a winning red card as the judge based on their modelled pref erences. Additionally, we must recreate the game logic, pre-process all cards, store and update the game state, and record the agent’s performance for analysis. Lastly, we need to optimize our agents to win as many games as possible.

Background

Encoding and Processing Natural Language

Turney & Pantel (Turney & Pantel, 2010) discuss the Word-Context Matrix, where words are represented as vectors, and each dimension represents a type of context in which a given word could be found. The value of each dimension is proportional to the frequency that the given word appears in that context; the frequency also determines the word’s meaning. Using differing clustering techniques, they extracted some meaning from the target word, suggesting that the relationship between verbs and nouns can play a role in determining both a verb and a noun’s meaning.

Mikolov et al. (Mikolov, Chen, Corrado, & Dean, 2013) proposed two novel word em bedding models: Continuous Bag-of-Words (CBOW), which averages the word vectors in a context and predicts the current word based on the context, and Continuous Skip-gram, which predicts surrounding words based on the current word. They tested their model’s effectiveness through a word similarity task, demonstrating significant improvements over previous techniques. CBOW and Skip-gram are the primary algorithms for creating word embeddings in Word2Vec.

Aside from Word2Vec, other popular word embedding models include GloVe and Doc2Vec. Pickard (Pickard, 2020) describes how Word2Vec outperformed GloVe at multiword expressions, which are word combinations that exhibit one or more idiosyncrasies. Pickard also describes how Bidirectional Encoder Representations from Transformers (BERT) are used for similar Natural Language Processing (NLP) tasks, but it considers the sentence in which the words appear. The BERT model goes beyond what GloVe and Word2Vec use, so he omitted BERT from the comparison. Pickard found that Word2Vec outperformed GloVe by a substantial degree.

Related Work

Representing, Measuring, and Predicting humour

While most existing humour detection models use feature extraction to identify humorous words or phrases, Guo et al.(Guo et al., 2022) introduced an alternative approach called Fedhumour. This model uses BERT and fine-tunes pre-trained weights to update its understanding of humour based on individual preferences. Guo et al. demonstrated that Fedhumour outperformed other humour recognition models, including Doc2Vec (with the bag of words approach), Word2Vec (with a Random Forest classifier), and two BERT model variations.

Gultchin et al. (Gultchin, Patterson, Baym, Swinger, & Kalai, 2019) examined how word embeddings can represent word based hu mour. They used embeddings like GNEWS, WebSubword, Webfast, and WebGlove, all of which trained on different token sizes, to predict humour ratings from three datasets. Using a least-squares LR model, they could predict the mean humour ratings from the datasets for all four-word embeddings. GNEWS, with the smallest dataset, performed the worst. Despite this, all embeddings showed significant correlations with humour ratings, proving their effectiveness. Cluster analysis also revealed demographic differences in humour, highlighting the utility of word embeddings in humour detection.
Weller & Seppi (Weller & Seppi, 2019) extended the AI’s ability to determine whether a joke is humourous or not. Their

agent learned to identify funny jokes based on ratings from Reddit's r/Jokes thread. They also refer to Convolutional Neural Networks (CNN) and another method which uses a transformer.

In their analysis of humour in the card game Cards Against Humanity, Ofer et al (Ofer & Shahaf, 2022) utilized the online version of the game to train several models to predict winning jokes, which focused primarily on the punchline cards. Their findings suggested that the context surrounding the punchline had minimal influence on the perceived humour compared to the punchline itself. Further more, short punchlines of 5 words or less were referred, with a 9% higher win rate.

Cosine Similarity

A common NLP method is Cosine Similarity, which calculates the cosine of the angle between vectors in multidimensional vector space. Cosine Similarity can be used to determine the relative similarity between words. For example, an output of 1 means the two vectors are identical, an output of 0 means they are orthogonal to one another, and an output of -1 means they are diametrically opposed to one another. This method seems plausible for generating two possible AI archetypes for the game of Apples to Apples: a literalist and a contrarian. The literalist would have a strong preference for similar word pairings. The contrarian would have a strong preference for dis similar word pairings. However, it is not clear how one could use Cosine Similarity to design effective preference types beyond those two, such as a preference for humourous card pairrings.

Methods

Word2Vec and Dataset Selection

During our research, we learned about Word2Vec, which vectorizes words into n dimensional vectors. Each dimension theoretically represents a feature of a given word, though the features are more abstract in nature. Word embeddings generated from models like Word2Vec have the useful property of placing semantically similar words close in vector space, allowing measures like Cosine Similarity to gauge word resemblance.

We chose Word2Vec for several reasons. First, it outperforms GloVe for multiword expressions, as noted in our background section. Second, BERT and Doc2Vec appear to be better suited for broader context language processing, which is not necessary for the short text on Apples to Apples cards. Lastly, we had access to Google's pre-trained Word2Vec model, trained on 100 billion words from Google News, providing 300-dimensional vectors for about 3 million words and phrases. This pre-trained model is ready to use and expected to cover all words in the Apples to Apples sets of cards.

However, using the Google News pre-trained Word2Vec model has drawbacks: the binary file is large (3.5 GB uncompressed) and is not customizable in its training. Apples to Apples has a more limited vocabulary than what is available in the Google News dataset, so most of the data is unused. A smaller, customized dataset would reduce file size and runtime computation. Furthermore, with a custom pre-trained dataset we could have more control over how the AI agents select word pairings. Despite these tradeoffs, we decided that the advantages of the pre-trained model outweighed the downsides.

Implementation

We hypothesized that instead of boxing every judge into a preference type and modelling those types only using Cosine Similarity or other similar methods, the agent should directly access the float values and leverage known machine-learning techniques or other mathematical or statistical tools. Thus, the agent effectively maps the player's preferences directly to the abstract encoding of meaning in the word embeddings. We concluded we could combine the green and red card vectors by multiplying them, feature by feature, and that the most highly correlated features between the two would result in the highest values, while low correlated features would get much smaller values. Furthermore, we could use a simple linear equation $y = mx + b$ where x represents whether the green and red card pairing was a winning one, y

represents whether the green and red card pairing was a winning one (y = 1) or a losing one (y = -1). Therefore, the opponent's preferences could be modelled using the slope (m) and intercept (b). In summary, our agent interpolates its opponent's preferences as a linear function and uses linear regression to determine the slope and intercept of the decision boundary. We call this design a Linear Regression Agent (LRA).

Initially, the LRA represents the preferences of opposing players by assuming that they have the same preferences as itself. Therefore, the opponent models begin with the same slope and intercept vector values and are then replaced by the slope and intercept vector values produced by LR on that opponent player's continuously updated history of green and red card pairings. The starting preferences are static.

Linear Regression Model

At runtime, we calculate the score using the LRA's model of the current judge. For each card, we calculate: score = compsum(mrg + b), where m is the judge's slope vector, r is the red card vector, g is the green card vector, and b is the judge's bias vector. The compsum function adds every component of the vector, thus producing a scalar. The LRA chooses the red card with the highest score.

We keep a history of all green and red card pairings and their respective word embed dings for each of the opponent judges. We then use LR on each combination of the green card, the winning red card, our initial/previous representation of the judge's modifying vector (m) and the judge's bias (b); LR is performed on each vector dimension individually, and the resulting linear model allows us to calculate how far off our representation of the judge's preferences was. We then update that judge's slope vector and bias vector accordingly based on the winning and losing cards' vector values.

LRA Archetypes

For testing purposes, we devised 3 AI archetypes: A Literalist (Ltrls) - who likes highly associated words that are logical to pair together; a Contrarian (Cntrn) - who likes the least associated words, and a Comedian (Cmdn) - who likes humourous combinations. We created each model archetype in different ways. The Literalist's slope vector values were set to positive 1s, and the bias vector values were set to 0s. With this model, the more closely related a pair of green and red cards is, the more similar each component in word embedding will be. For example, when the same component is similar between two words, they will both be positive or both be negative, so multiplying them together will result in a positive number, and more of these similar components result in a higher score value. In this way, we use the embedding directly to achieve the highest association value. Conversely, the Contrarian's slope vector values were set to negative 1s, and their bias vector values were set to 0s. This results in the model choosing the lowest association values.

For the Comedian archetype, slopes and biases were calculated using LR on preselected green and red card pairings. For each green card, we manually selected a "winning" red card that was humourous as a pairing and a "losing" red card that was not and then serialized them. We did Linear Interpolation on all the components upon start-up, producing a static linear model that should emulate the preferences of the card pairings chosen in the file. We used 30 pairs for testing. This was the minimum number of pairings needed to reasonably emulate a humour preference. We chose winning pairings based on what we considered to be ”funny” pairings, such as the green card ”disgusting” with the red card ”my high school prom.” We chose losing pairings that seemed unlikely to appear humourous to most people, such as the green card "disgusting," with the red card "nosebleeds."

Performance Improvements

Following our implementation, we investigated improvements that could be made to enhance the LRA's performance. The LRA solely considered the green and red cards and the judge's preferences.

There are many potential factors that could have improved the LRA's performance. We identified one such major factor: we were resetting the models in between every game. With this setup, the LRA performed no better than chance and many times performed even worse, getting an aggregate LRA round win rate of less than 50% and sometimes a game win rate of less than 40%.

When multiple games of Apples to Apples are played, and no new players are introduced between games, humans carry over their knowledge of another player's preferences between games. We implemented a similar setting for the LRA, allowing them to preserve learned opposing model vectors between games. This improved the LRA's performance by around a 10-15% increase in aggregate round win rate. We added a feature to the LRAs to store the red words that were revealed in each round but not chosen by the judge; we refer to these as losing red cards. The observation was that the losing red cards have just as much information about the opponent judge's preferences as the winning red cards. We modified the agent so that after each round, the green card, the accompanying winning red card and all the losing red cards were stored in the model as an aggregate set of data points. We take the component-wise product of the green card vector with all combinations of the losing red card vectors, just as we do with the winning red card vector, but the corresponding y values are set to negative 1s instead of positive 1s. To match this new implementation, we added 30 losing red card pairings in addition to the 30 existing Comedian preselected green and winning red card pairings.

After fully implementing this change, there were some noticeable differences. The LRA's aggregate win rate increased significantly, but the program's runtime grew exponentially with each round, taking longer to process the losing red cards. However, we concluded that the increase in win rate outweighed the impact on performance, so we opted to keep this change.

Experiments

In all test cases, we ran games with the same number of LRAs and random agents (Rand). This ensures a balance of LRAs to random agents, which avoids any unfair advantages for the LRAs. If the LRAs are able to win more rounds than the random agents with statistical significance, it would be plausible to conclude that they exhibit partial learning the preferences of opponent players. We ran 22 sets of 100 games, with varying points to win and different numbers of players in the game, to determine whether the LRA could perform consistently well across many different scenarios. We played all possible combinations of games with either 5, 7, or 10 points to win and 4, 6, 8, or 10 players, which makes up 12 different game scenarios. We played at least one of each of the 12 different game scenarios; the remaining 10 of the 22 games were used to experiment with different LRA archetypes. Some games had a single LRA archetype, while others had differing archetypes. The total possible combinations between game scenarios and combinations of LRA archetypes were much too large to test, so some variations were not included due to their similarity with other game scenarios or how long they would take to run. These dynamic elements in our games are our manipulated variables.

We used the same game settings across all variations of the game sets: the same player types are used throughout all 100 games in each set, the judge for each round is cycled, which follows the game rules, and the starting judge for each game is also cycled. As mentioned previously, we did not reset the models representing the LRA's opponents within any set of 100 games. This means the models are continuously updated for 100 games until the program ends, and only when the program is run again will these values be reset back to the default initialized values. The losing red cards feature was used for all games. We used all the base sets and expansion sets for both the green cards and red cards rather than limiting the deck sizes to that of a normal Apples to Apples deck size. All these elements of the game are our control variables and were consistent throughout all sets of 100 games.

Results

Win Rates

The results show that our agent is very capable of interpolating a static preference type. For all Round Summary tables, we list the total round win counts and total round win rates for the entire 100 game set. We also provide mean wins per game, and the standard deviation, and the confidence intervals and p values for the round win rates. For all Game Summary tables, we list the total game win counts, total game win rates for the entire 100 game set, and the confidence intervals and p values for the game win rates.

We calculated 95% confidence intervals for each agent's win rate using a normal approximation to the binomial distribution. The confidence interval quantifies uncertainty in the win rates observed over the total number of games. A z-score of 1.96 was used to compute the intervals. These intervals were used to assess the reliability of the win rate estimates before performing hypothesis tests.

The p-values were calculated using a one tailed binomial test. The alternative hypothesis assumed that the win rates for our LRA were larger than the random agent's win rate. A significance level of 0.05 was used for the tests. In the tables, a p-value of 0 signifies $p < 0.00001$. For each agent, the number of wins was compared to the total number of rounds/games played, and the corresponding values were determined to evaluate whether the observed win rates were larger. All game sample sizes were n=100 while round sample sizes varied depending on the total number of points to win, the number of players in the game, and how fast a player was able to win the game. The round sample sizes varied between n=1137 and n=4905. We observed a trend where the total number of players in the game is positively related to aggregate LRA agent round wins. This trend holds true for all points-to-win variations. Table 1 shows that for 100 games, 10 points to win, 5 LRAs and 5 random agents, the LRA aggregate round win rate is 68.14%, while Table 2, with 100 games, 10 points to win, 2 LRAs and 2 random agents, shows an LRA aggregate round win rate of 61.12%.

Player	Win Cnt	Win %	Mean Wins per Game	Std Dev	Conf Int	p-Value
LRA Model Cmdn 1	664	13.54	6.64	2.156	12.58 to 14.49	0
LRA Model Cntrn 1	692	14.11	6.92	2.378	13.13 to 15.08	0
LRA Model Cntrn 2	666	13.58	6.66	2.732	12.62 to 14.54	0
LRA Model Ltrls 1	663	13.52	6.63	2.560	12.56 to 14.47	0
LRA Model Ltrls 2	657	13.39	6.57	2.495	12.44 to 14.35	0
Rand Ag 1	302	6.15	3.02	1.833	-	-
Rand Ag 2	300	6.12	3.00	1.892	-	-
Rand Ag 3	339	6.91	3.39	1.984	-	-
Rand Ag 4	306	6.24	3.06	1.719	-	-
Rand Ag 5	316	6.44	3.16	2.023	-	-

Table 1: Round Summary for 100 games, 10 points to win, 5 LRAs and 5 random agents

Player	Win Cnt	Win %	Mean Wins per Game	Std Dev	Conf Int	p-Value
LRA Model Cntrn 1	350	30.17	3.50	1.520	27.53 to 32.81	4e-5
LRA Model Ltrls 1	359	30.95	3.59	1.357	28.29 to 33.61	0
Rand Ag 1	233	20.09	2.33	1.607	-	-
Rand Ag 2	218	18.79	2.18	1.627	-	-

Table 2: Round Summary for 100 games, 10 points to win, 2 LRAs and 2 random agents

We used statistical analysis on the round win rates and game win rates to determine the confidence intervals and p-values. For the round wins, we found that with 100 games, most of the p-values for the LRAs were well below 0.05, which was especially true for a higher number of players and for a higher number of points to win. The highest p-value obtained was 0.05877.

The p-values for game win rates exhibit a different behaviour. In a game set with 10 players per game, the highest p-value of one LRA was 0.21824; for another LRA was 0.07257, but most of the p-values across all game sets were below 0.05, many of which were much closer to 0. As discussed earlier,

all our 22 test sets comprised 100 games; increasing the total number of games played in each set would have decreased the LRA's p-values, but that would have required longer processing time.

As shown in Table 3 in the case of only four players, our LRAs still had good results, but less so compared to when there were more players and points to win. This demonstrates that even at one of their worst performances, the LRAs still performed high above the probability of random chance. Other tests also portrayed similar outcomes in both the round and game win rates. The game set for Table 3 has the fourth lowest round and game win rate out of all 22 variations we did during testing, further proving how consistent and well the LRAs play. The lowest score for both game and round win rates during testing came from a set of 100 games, with 5 points to win, 2 Comedian LRAs, and 2 random agents, where the aggregate LRA game win rate was 75%, and the round win rate was 59.40%.

Player	Win Cnt	Win %	Mean Wins per Game	Std Dev	Conf Int	p-Value
LRA Model Cmdn 1	346	29.67	3.46	1.513	27.05 to 32.30	1.7 e-4
LRA Model Cntrn 1	355	30.45	3.55	1.519	27.80 to 33.09	1e-5
Rand Ag 1	227	19.47	2.27	1.475	-	-
Rand Ag 2	238	20.41	2.38	1.580	-	-

Table 3: Round Summary for 100 games, 5 points to win, 2 LRAs and 2 random agents

Table 4 provides further evidence that as the player count and/or number of points to win goes up, so does the aggregate LRA performance. The game set for Table 4 was a particularly interesting set because there was only one of each LRA archetype in the game, which meant that the LRAs could potentially deviate from their initial understandings of their opponents' preferences since all archetypes start every set of games under the assumption that all other players have the same preferences as their own. The fact that this game set has such high win percentages demonstrates that the LRAs could identify opposing players as having completely different preferences from their own at some point throughout the set of games.

Player	Win Cnt	Win %	Confidence Interval	p-Value
LRA Model Cmdn 1	33	33.00	23.78% to 42.22%	5e-5
LRA Model Cntrn 1	28	28.00	19.20% to 36.80%	3.1e-3
LRA Model Ltrls 1	34	34.00	24.72% to 43.28%	2e-5
Rand Ag 1	2	2.00	-	-
Rand Ag 2	2	2.00	-	-
Rand Ag 3	1	1.00	-	-

Table 4: Round Summary for 100 games, 10 points to win, 3 LRAs and 3 random agents

Judge Choices

Throughout our development process, we monitored each judge agent's preferences to declare other agents as the winner, and we represented those choices as a heatmap. In Figure 1, we provide an example of such a heatmap; this is the heatmap for the same 100-game set displayed in Table 1 (10 points to win, 5 LRAs and 5 random agents). On the y-axis, we have all players in the game as judges; on the x-axis, we have all players in the game as regular players. The value in the matrix portion denotes the percentage of rounds for which the judge (y-axis) chose each player (x-axis) as the winner of that round. The color scaling (white to black) ranges from 0 to 28.8%, and are the lower and upper bounds, respectively, of the total win selections. Of the 22 game sets that were simulated, every heatmap had a similar distribution to Figure 1.

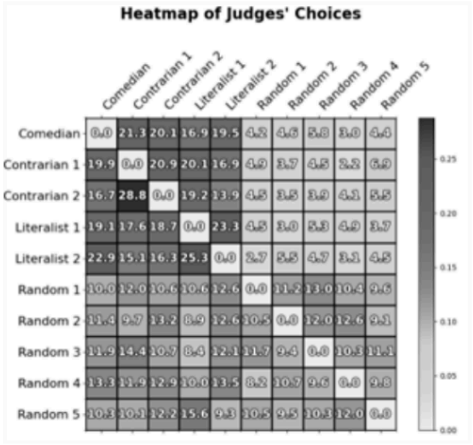


Figure 1: Heatmap of Judge Selections for 100 games, 10 players, 10 points to win, 5 LRAs, 5 random agents. Row labels represent each individual judge. Column labels represent the opponent players chosen as the winner. The value intersecting rows and columns represent the percentage that the judge chose that given player as a winner.

Discussion

Overall, our results indicate that the agents we implemented were successful in playing Apples to Apples. Our Linear Interpolation model showed promising potential for a significantly high round and game win rate, which appears to indicate that our LRAs were able to learn opposing player preferences. Although we wanted the Comedian to make creative choices to achieve humour, we cannot assert that we succeeded in this regard since humour is subjective. The LRA's ability to be humourous could have partially been due to the limited number of cards at their disposal in each given round. A more thorough measurement of how humourous the LRAs are would be to test them against many human players with differing preferences, but this is beyond the scope of our project. Further research could yield more significant findings in this regard.

Further Considerations

Continuing our research into this topic would entail experimenting with BERT for word embeddings, as it excels in context-dependent material, which could be helpful for humour analysis. This could improve our agent's ability to interpret meaning from context. Additionally, training our own Word2Vec model using custom datasets will allow us to reduce the number of vector dimensions, thus lowering computations during training or increasing them to represent more features of a word.

We would also like to explore more ways to determine the most important features for accurately representing a judge's preferences. More research might lead to an effective solution, possibly choosing features dynamically throughout the game and tracking the most important ones for each judge separately.

Another possible improvement involves tracking all red cards played in each round, even when the LRA is a judge. This could help us understand if opponent judges used the same criteria for choosing a red card as they do for selecting from their hand. Collecting these extra data points to regress linearly could potentially help our agents converge faster on each opponent's preferences.

Conclusion

We gathered the results and investigated a novel approach for AI agents to play Apples to Apples using Linear Interpolation to estimate which red card a given judge would choose as the "best." Additionally, we aimed for the LRAs to learn their opponents' preferences and perform well in Apples to Apples and tried many variations to optimize the aggregate LRA win rate. Our analysis showed that the LRAs were able to achieve high round win rates and very high game win rates, all of which were statistically significant. Although we could not extrapolate how well the LRAs would perform against human players, our results demonstrated potential for a new, unique approach to word association games that provides insight into the area of automated playing of natural language card games.

References

Gultchin, L., Patterson, G., Baym, N., Swinger, N., & Kalai, A. (2019, 09–15 Jun). Humor in word embeddings: Cockamamie gobbledegook for nincompoops. In K. Chaudhuri & R. Salakhutdinov

(Eds.), *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 2474–2483). PMLR. Retrieved from https://proceedings.mlr.press/v97/gultchin19a.html

Guo, X., Yu, H., Li, B., Wang, H., Xing, P., Feng, S., . . . Miao, C. (2022, May). Federated learning for personalized humor recognition. *ACM Trans. Intell. Syst. Technol.*, *13*(4). Retrieved from https://doi.org/10.1145/3511710 doi: 10.1145/3511710

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient estimation of word representations in vector space. In the International *conference on learning representations.* Retrieved from https://api.semanticscholar.org/CorpusID:5959482

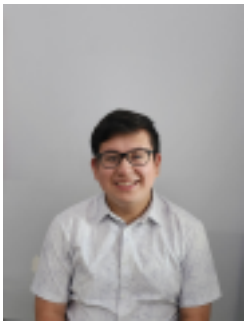
Ofer, D., & Shahaf, D. (2022, December). Cards against AI: Predicting humor in a fill-in-the-blank party game. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Findings of the association for computational linguistics: Emnlp 2022* (pp. 5397–5403). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.findings-emnlp.394/ doi: 10.18653/v1/2022.findings-emnlp.394

Pickard, T. (2020, December). Comparing word2vec and GloVe for automatic measurement of MWE compositionality. In S. Markantonatou et al. (Eds.), *Proceedings of the joint workshop on multiword expressions and electronic lexicons* (pp. 95–100). online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.mwe-1.12/

Schaeffer, J., Burch, N., Bjornsson, Y., Kishimoto, A., Muller, M., Lake, R., Sut- ¨ phen, S. (2007). Checkers is solved. *Science*, *317*, 1518 - 1522. Retrieved from https://api.semanticscholar.org/CorpusID:10274228

Turney, P., & Pantel, P. (2010, 03). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*. doi: 10.1613/ jair.2934

Weller, O., & Seppi, K. (2019, November). Humor detection: A transformer gets the last laugh. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 3621–3625). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1372/ doi: 10.18653/v1/ D19-1372



Oscar De Leon is currently studying for his Bachelor of Science at MacEwan University. He is majoring in Computer Science and minoring in Psychology. His fields of interest include database management systems and artificial intelligence. Additionally, he wants to apply his interests in behavioural and social psychology to computer science in order to help humans better understand each other.

Isaac McCrackenis an undergraduate student in Computer Science at MacEwan University. His academic interests include programming languages, compiler design, mathematics, and artificial intelligence. He is particularly focused on exploring the theoretical and practical aspects of software development.

Kevin Ulliac graduated from MacEwan University with a Bachelor of Arts in Philosophy and a minor in Psychology in Spring 2017, and more recently with a Bachelor of Computer Science with a minor in Mathematics in April 2025, also from MacEwan. His major areas of interest are in machine learning and artificial intelligence, data science/analysis, music-related software like plugins and DAWs, and standalone/desktop applications including custom debugging tools. He also has a passion for linear algebra and calculus and their application in software and engineering.

Calin Anton holds an MSc from the University of Bucharest, Romania and a Ph.D. from the University of Alberta, Canada, both in Computer Science. He has been teaching in different roles at postsecondary institutions for more than 25 years. For the last 15 years, he has taught at MacEwan University in Edmonton, Canada, where he is currently an associate professor of Computer Science. His current interests reside in Computer Security and Artificial Intelligence.

Semantic, Orthographic, and Morphological Biases in Humans' Wordle Gameplay

Authors

Jiadong (Gary) Liang (University of Toronto; esgary.liang@mail.utoronto.ca)
Adam Kabbara (University of Toronto; adam.kabbara@mail.utoronto.ca)
Cindy Liu (University of Toronto; cindyjy.liu@mail.utoronto.ca)
Ronaldo Luo (University of Toronto; ronaldo.luo@mail.utoronto.ca)
Kina Kim (University of Toronto, IBM; giyeon.kina.kim@ibm.com)
Michael Guerzhoy (University of Toronto; guerzhoy@cs.toronto.edu)
DOI: 10.1145/3774399.3774406

Abstract

We show that human players' game play in the game of Wordle is influenced by the semantics, orthography, and morphology of the player's previous guesses. We demonstrate this influence by comparing actual human players' guesses to near-optimal guesses, showing that human players' guesses are biased to be similar to previous guesses semantically, orthographically, and morphologically.

Introduction

Wordle is a daily word-guessing game where players attempt to identify a hidden five-letter word within six attempts (Wardle, 2021). Players usually attempt to minimize the number of guesses they make. Players also usually want to maintain a "streak" of having solved the game within at most 6 guesses for several days.

We explore the difference between near optimal play and human gameplay, which may be influenced by cognitive shortcuts and bi ases. In order to estimate near-optimal plays, we use the maximum-entropy heuristic. We verify that the heuristic is near-optimal. In settings where word association is important, humans are known to be influenced by salient past information, a phenomenon known as *priming* in psychology (Schacter & Buckner, 1998). We conjecture that priming effects exist in the game of Wordle as well. Additionally, we conjecture that humans will tend to depart less from previous guesses in order to minimize cognitive load.

We review the prior work on priming in psychology, and in particular on how priming influences future word choice. We then review the optimal strategy in Wordle, as well as heuristics that approximate it. We introduce our human guess data. We then present our approach to measuring human biases in Wordle gameplay and demonstrate the systematic differences between human plays and near optimal play.

Background: Human Cognitive Processes

Priming is a phenomenon in psychology where past experience influences behavior without the person's explicit knowledge of the influence (Schacter & Buckner, 1998). Specifically, one aspect of priming is word association. Prior works have demonstrated that the grammatical class, semantic meaning and rhyme of the previous (cue) word would influence the later (response) word by humans.

Deese (1962) conducted early research on word association, exploring the influence of

the grammatical class of cue words over word association on the next word. De Deyne and Storms (2008) followed up the study and suggested that no matter whether a noun, a verb, or an adjective are given as cues, the resulting association is most likely to be a noun. Furthermore, for noun cues, while still being dominant, the effect of paradigmatic association (associating with the same class of noun) would decrease when changing from first to second and third responses.

Steyvers and Tenenbaum (2005) demonstrate that an undirected free association network — constructed from data by D. Nelson (1999) that collects human participants' first responses associated with given cue words — where each word is a node and two words are connected if there exists a cue-response pair consisting of those two words — reveals that, on average, each word is connected to only 0.44% of the overall dataset. This finding underscores the sparseness of the association network where the probability of each word being the response given a cue word is not equally distributed.

Steyvers and Tenenbaum (2005) also use data collected by Miller (1995) and Fellbaum (1998), and found that the word network constructed based on semantics of words ex hibits sparseness, connectedness, neighbor ing clustering and power-law degree distribu tion, which are the same characteristics exhibited in the free association network, just to a varying degree.

Bullinaria and Levy (2007) and McDonald and Lowe (2022) observe the connection between information regarding lexical semantics and patterns of word co-occurrence. De Deyne and Storms (2008) also illustrate that the basic semantic features (coded in Wu and Barsa lou (2009)): "taxonomic," "entity," and "situation" are influential in terms of association responses, with "situation" being the most prominent.

D. L. Nelson et al. (1987) demonstrated the effect of rhyme on memory and word association. They run an experiment where subjects would initially study (read aloud) the cue target pair of a given rhyme; then 1.5-2 minutes after they finished studying, a meaning related cue word and its semantic relation with the target word would be given and the participants would be required to read it aloud and recall the word they studied (D. L. Nelson et al., 1987). In the experiment, cue words that rhyme with many other words would decrease the accuracy of the respondent, regardless of the meaning-related cue word (D. L. Nelson et al., 1987). Through conducting a further experiment that changed all the cue-target pairs studied to be meaning-related and only half to be also rhyme-related, D. L. Nelson et al. (1987) showed that the effect of rhyming appears only if the subjects actively attend to it when studying the word pairs.

Matusevych and Stevenson (2018) studied human word association based on word attributes.

Background: Wordle solving mechanisms

The objective of Wordle depends on the player — it can be maintaining the streak (i.e. try not to lose today's game), winning in as few guesses as possible, or even winning the game using funny words.

Most of the solving mechanisms are designed to optimize objectives regarding the number of guesses, such as minimizing the average number of guesses, minimizing the number of guesses in the worst case, etc. Those mechanisms can be classified into two classes: the exact optimization approach and heuristic approaches. The best approaches based on heuristics achieve results that are only marginally inferior to exact methods.

Bertsimas and Paskov (2024) found an optimal and efficient solution for Wordle that minimizes the average number of guesses using dynamic programming. Bertsimas and Paskov (2024) show that the word "SALET" is the best starting guess and the minimum average number of guesses required is 3.421. They demonstrate that under this approach the program never loses (i.e. it always completes the game within 6 guesses).

Heuristic approaches to Wordle do not guarantee an optimal result but are

relatively competitive, and can achieve performance that is very close to optimal. Doddle's minimax heuristic aims to minimize the number of guesses for the worst-case scenario with search depth of 1 (for each guess, it is only optimizing over all the situations after that single guess). For each guess, it iterates through all possible words in the game and chooses the one that minimizes the size of maximum partition (the amount of possible solutions after the current guess) as the guess. Given the starting guess as "SALET", it is guaranteed to finish the game in 5 guesses and have the average number of guesses to 3.482 (Cross, 2022). Doddle's entropy-based heuristic (also with depth 1) reduces the uncertainty at each step by choosing the guess that decreases (on average) the most number of potential solutions after that guess (Shannon, 1948) (Cross, 2022). It is also guaranteed to complete the game in 6 guesses and have the average number of guesses of 3.432.

Data

The human guess data was sourced from Reddit. The machine-generated guesses were obtained using Doddle, an open-source Wordle solver introduced earlier. Although an ideal comparison would be with the optimal model, the Doddle solver was chosen for computational reasons. It's important to note that the performance difference between the exact dynamic programming solution and the heuristic entropy solver is minimal: the exact solution achieves a minimum average of 3.421 guesses, while the heuristic-based solver has an average of 3.482 guesses for its minimax heuristic and 3.432 guesses for its entropy-based heuristic. Doddle's heuristic min-entropy solver that we use will be referred to as *near-optimal*.

Data collection

The data for this research project was collected from the r/Wordle subreddit, where people share their guesses online, contributing to a total of 83,000 data entries (Watchful1, 2023). We used regular expressions to extract guesses and colored square results from posts written in the standard Wordle format.

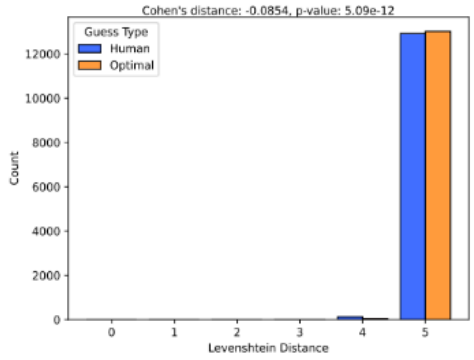
Methods

Measuring Human Biases

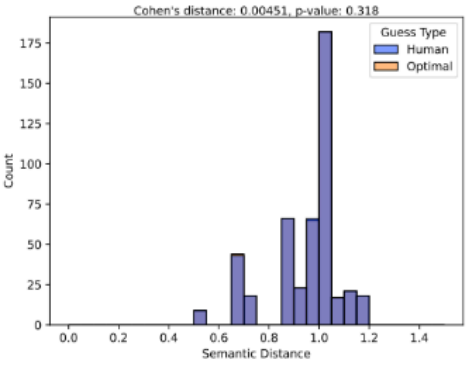
To quantitatively assess the influence of human cognitive biases in Wordle games, human plays are compared to their entropy based near-optimal counterparts, where five different metrics described below are utilized to reveal different aspects of human biases (semantic, orthographic, and morphological). For each guess in the data, the metrics below are computed through comparing that guess with the previous one (instead of comparing with all prior guesses) unless otherwise stated.

Levenshtein Distance The Levenshtein Distance measures the minimum number of edits — insertions, deletions, or substitutions — needed to transform one word into another (Levenshtein, 1966). This feature captures how closely a player's subsequent guesses align with their previous ones in terms of structural similarity. A smaller Levenshtein distance indicates that the player is selecting guesses that are more similar to their prior attempts, potentially reflecting a reluctance to explore novel letter combinations or a preference for minimizing cognitive effort.

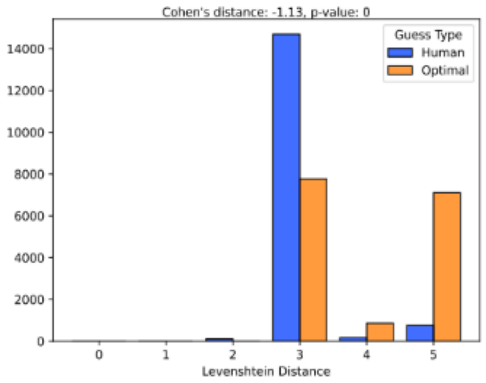
Semantic distance The GLoVe distance is computed using negative cosine similarity between word embedding pairs. Words are represented as vectors using GLoVe, and GloVe distances are computed using negative cosine similarity (Pennington et al., 2014).



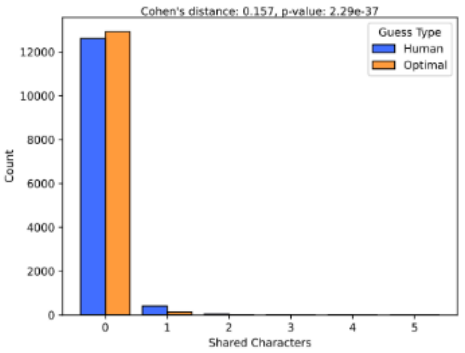
(a) Levenshtein distance between human guesses and near-optimal guesses for 0g0y5b: both choose distance 5 most of the time. Humans subotimally play letters they know aren't there. Note: reference the bottom of this page for Wordle notations.



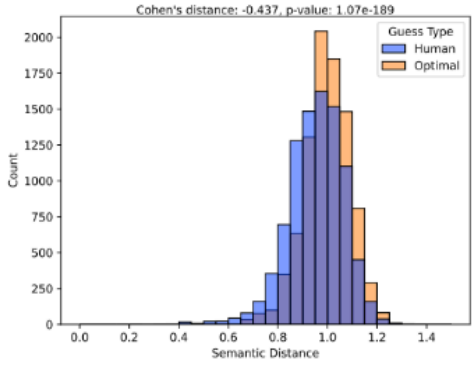
(d) Semantic distance between human guesses and near-optimal guesses for 3g2y0b: no bias.



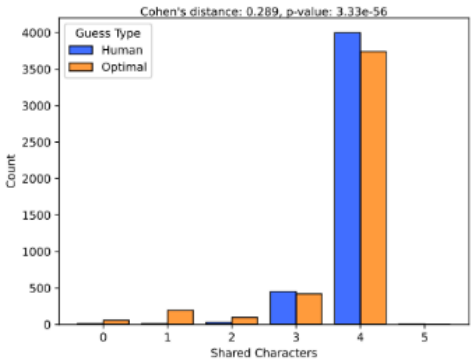
(b) Levenshtein distance between human guesses and near-optimal guesses for 2g0y3b: humans under-explore compared to near-optimal.



(e) Character-level difference between human guesses and near-optimal guesses for 0g0y5b: humans play very obviously suboptimally by reusing characters they know are not there.



(c) Semantic distance between human guesses and near-optimal guesses for 0g0y5b: humans slightly biased towards underexploring.



(f) Character-level difference between human guesses and near-optimal guesses for 2g2y1b: humans play sub-optimally by using new characters.

Figure 1: Notation (c_g, c_y, c_b): c_g - number of "green" guesses (correct letter in the correct place); c_y - number of "yellow" guesses (correct letter in the incorrect place); c_b - number of "black" guesses (incorrect guess).

Character-level difference measures the extent to which players deviate from their initial guesses, quantified by the number of differing characters between subsequent guesses. More character-level difference suggests a greater willingness to explore alternative solutions. Conversely, minimal deviation indicates an over-reliance on early guesses.

Rhyme To determine whether two words rhyme or not, their phonetic transcription was used. This was achieved with the help of the pronouncing library, which provides a phonetic transcription based on the CMU Pronouncing Dictionary "The CMU Pronouncing Dictionary" (2015) . Two words are considered to have a *perfect rhyme* if they have matching phonetic endings which include stressed vowels. We assess if the guess rhymes with the previous one.

Cohen's d Cohen's d is a measure of effect size that quantifies the standardized difference between two means, in this case, human and model performance (Sullivan & Feinn, 2012). Cohen's d transforms the absolute difference between means into standard deviation units, enabling a direct comparison of the magnitude of this difference across various metrics. Effect sizes are traditionally classified as small (d = 0.2), medium (d = 0.5), and large ($d \geq 0.8$) (Carson, 2012)

Experiments

We compare how human guesses/plays differ systematically from near-optimal play. We obtain distributions of human plays and near optimal plays, and compare them. We assess the effect size using Cohen's d, and we computed the p-values based on the t-statistics for the difference between the two distributions.

We analyze separately games starting from different positions. We use the notation $c_g c_y y c_b$, where c_g denotes the number of "green" guesses (correct letter in the correct place), c_y denotes the number of "yellow" guesses (correct letter in the incorrect place), and c_b denotes the number of "black" guesses (letter guesses that are incorrect).

In Figure 1, we present some observations from the results of our comparison of human play with near-optimal play for specific configurations. We observe that in many, though not all cases, humans are biased towards their previous guesses — especially early in the game, when many valid options remain. This suggests that human gameplay, which combines creativity with optimization, tends to reflect more about players' strategic tendencies when the solution space is still large.

We additionally report that the optimal guess rhymes with the previous guess 7.3% of the time, but humans make a guess that rhymes with the previous guess 9.3% of the time (p value < 0.001).

Conclusions

Human gameplay in Wordle exhibits a bias toward previous guesses semantically, orthographically, and morphologically (e.g., sharing the last syllable). The bias is systematic, indicating human gameplay does not merely randomly deviate from optimal play. Initial work indicates that those biases affect human perception of Wordle games (Luo et al., 2025). Our findings can influence game design, especially as it pertains to optimizing the experience of games, as well as understanding how and why people enjoy word games that involve word association.

References

Bertsimas, D., & Paskov, A. (2024). An exact solution to wordle. *Operations Re search*.

Bullinaria, J. A., & Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, *39*, 510–526.

Carson, C. (2012). The effective use of effect size indices in institutional

research. *31st Annual Conference of the North East Association for Institutional Re search*, *41*, 41–48. The CMU pronouncing dictionary [Accessed: October 13, 2024]. (2015).

Cross, A. (2022). Doddle. *https://github.com/CatchemAL/Doddle*.

De Deyne, S., & Storms, G. (2008). Word associations: Network and semantic properties. *Behavior research methods*, *40*(1), 213–231.

Deese, J. (1962). Form class and the determinants of association. *Journal of verbal learning and verbal behavior*, *1*(2), 79– 84.

Fellbaum, C. (1998). Wordnet: An electronic lexical database. *MIT Press*, *2*, 678– 686.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the So viet physics doklady*.

Luo, R., Liang, G., Liu, C., Kabbara, A., Bakhtawar, M., Kim, K., & Guerzhoy, M. (2025). Automatically detecting amusing games in wordle. *Proceedings of the International Conference on Computational Creativity*.

Matusevych, Y., & Stevenson, S. (2018). Analyzing and modeling free word as sociations. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *40*.

McDonald, S., & Lowe, W. (2022). Modelling functional priming and the associative boost. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, 675–680.

Miller, G. (1995). Wordnet: An on-line lexical database [special issue]. *International Journal of Lexicography*, *3*(4).

Nelson, D. (1999). The university of south florida word association norms. *http://w3. usf. edu/FreeAssociation*.

Nelson, D. L., Bajo, M. T., & Canas, J. J. (1987). Prior knowledge and memory: The episodic encoding of implicitly activated associates and rhymes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*(1), 54.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. http : / / www . aclweb.org/anthology/D14-1162

Schacter, D. L., & Buckner, R. L. (1998). Priming and the brain. *Neuron*, *20*(2), 185– 195.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, *27*(3), 379–423.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive science*, *29*(1), 41–78.

Sullivan, G. M., & Feinn, R. (2012). Using effect size—or why the p value is not enough. *Journal of graduate medical education*, *4*(3), 279–282.

Wardle, J. (2021). Wordle [Online game]. https : / / www . nytimes . com / games / wordle/index.html

Watchful1. (2023). Subreddit comments/sub missions 2005-06 to 2023-12. https : / / www . reddit . com / r / pushshift / comments / 1akrhg3 / separate dump files for the top 40k subreddits/

Wu, L.-l., & Barsalou, L. W. (2009). Perceptual simulation in conceptual combination: Evidence from property generation. *Acta psychologica*, *132*(2), 173– 189.

Appendix: Sample Data

This appendix contains a series of visualizations that support the main findings of this study by illustrating key differences between human gameplay and near-optimal model guesses in Wordle. The figures

present histograms for several metrics used in our analysis, including Levenshtein distance, semantic distance (Word2Vec and GloVe), shared syllables, shared characters, and rhyme occur rence. Each figure compares human guesses against the optimal model guesses under various game states, such as those with partial or no feedback (e.g., 0g0y5b, 2g0y3b). Metrics such as Cohen's d and p-values are provided to indicate the magnitude and statistical significance of the observed differences.

Ultimately, these visualizations highlight the extent to which human players rely on structural and semantic similarities in their guesses, favoring familiarity over exploration, particularly when faced with partial confirmation of correct letters.

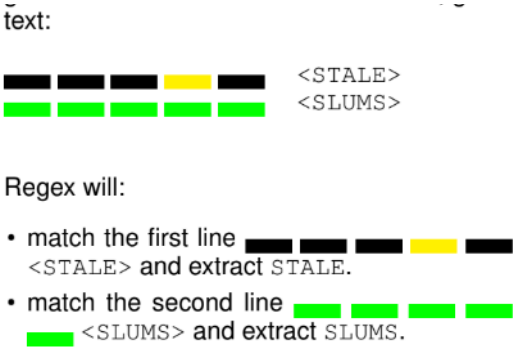
The full data we used is in a further Appendix.

Data cleaning

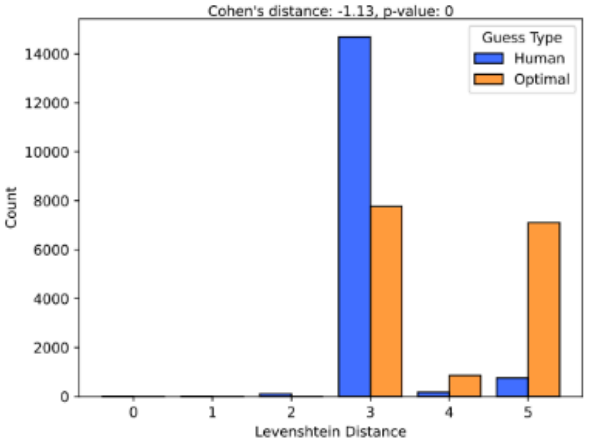
For each guess, the unnecessary parts such as the special symbols (, &!lt;) are removed. To ensure the integrity of the data provided by Reddit users, a cross-referencing process was conducted between the dataset and a Wordle answers database. This approach verified the accuracy of the Wordle IDs submitted and ensured that the answers had not been altered. If a Wordle ID was not provided, the corresponding game was considered illegible, as there was no way to confirm the authenticity of the data. In cases where a Wordle ID was provided without an answer, the last guess was cross-referenced with the Wordle answers dataset. If the last guess matched the correct answer, it was recorded; otherwise, the entry was removed.

To maintain consistency, all guesses were converted to lowercase. The data cleaning process eliminated entries where users did not include their guesses or submitted answers for non-Wordle games. Additionally, any unsolved Wordle games were removed from the dataset. As a result of these cleaning efforts, the dataset was reduced from 83,000 entries to a more manageable 65,000 entries. Ultimately, information about the player, words guessed, and the number of guesses each user made are obtained.

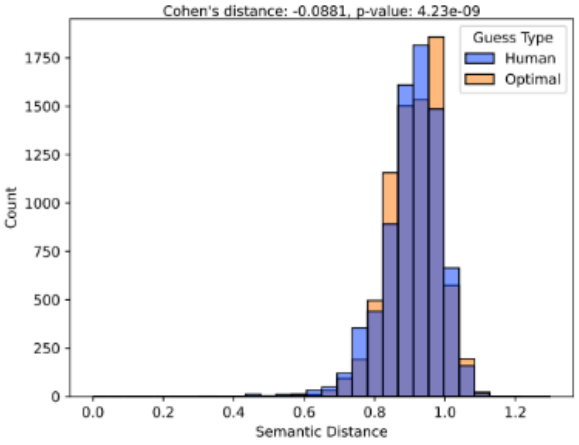
Regex is used to identify lines in Wordle posts where users have displayed both their square results and their guesses. Regex searches for the combination of colored squares and five letter guesses enclosed in special HTML-like tags (<WORD>), ensuring that only complete guess lines are extracted. For instance, given text:



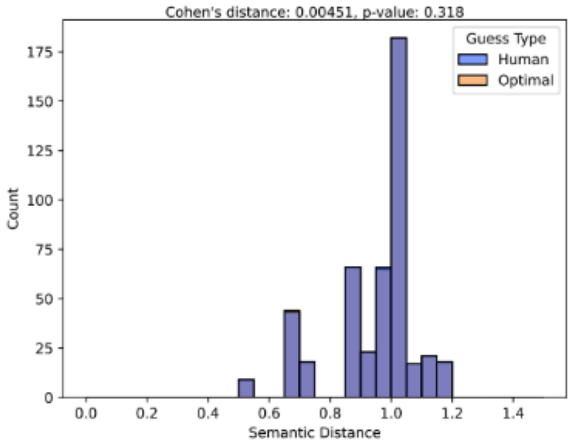
Appendix: more sample histograms



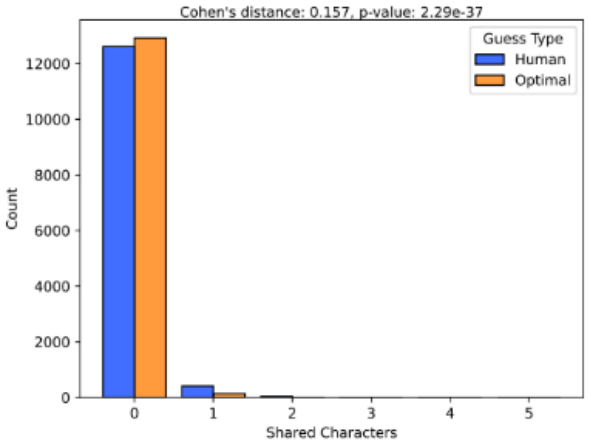
Levenshtein distance histogram for 2g0y3b



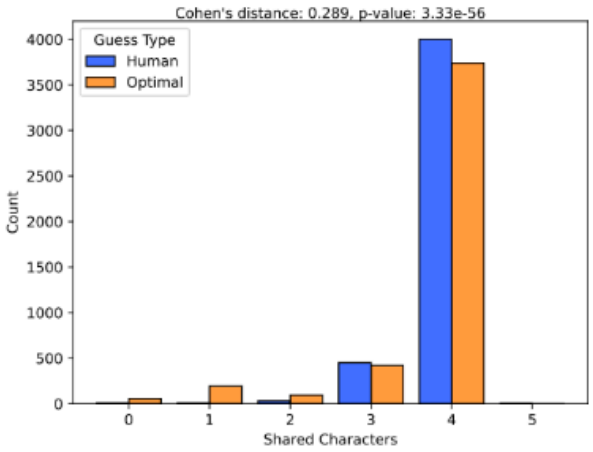
Word2vec distance histogram for 0g0y5b



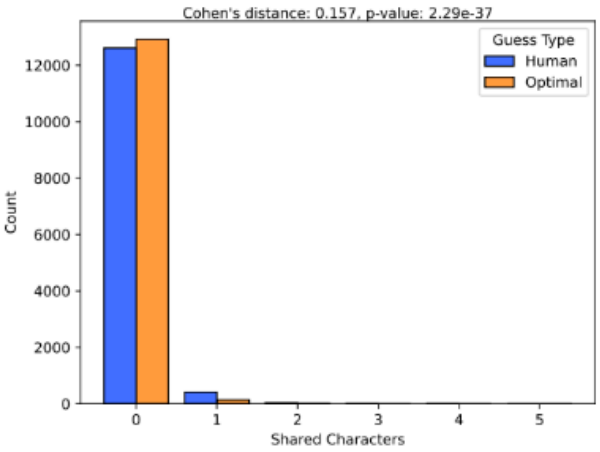
Glove distance histogram for 2g3y0b



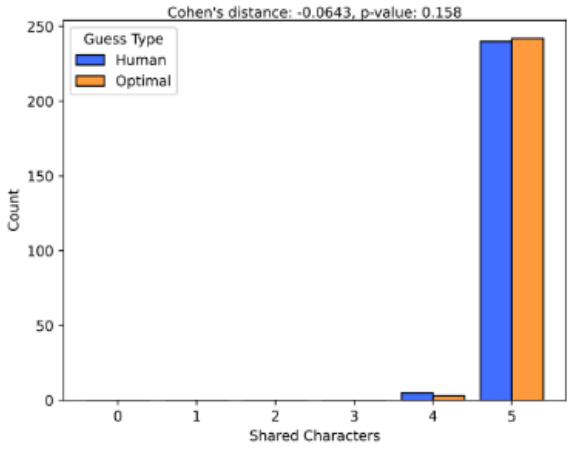
Common syllables histogram for 0g0y5b



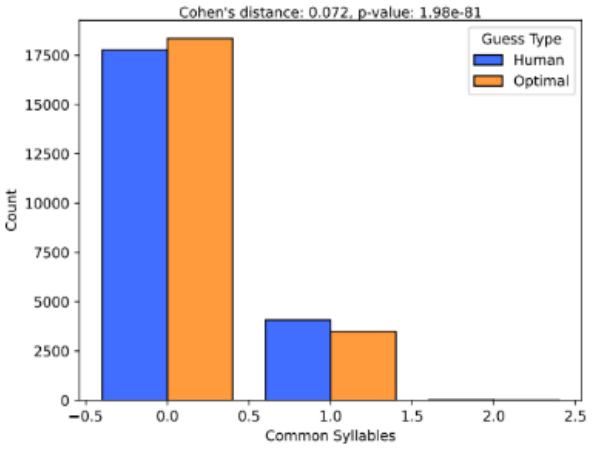
Shared chars histogram for 2g2y1b



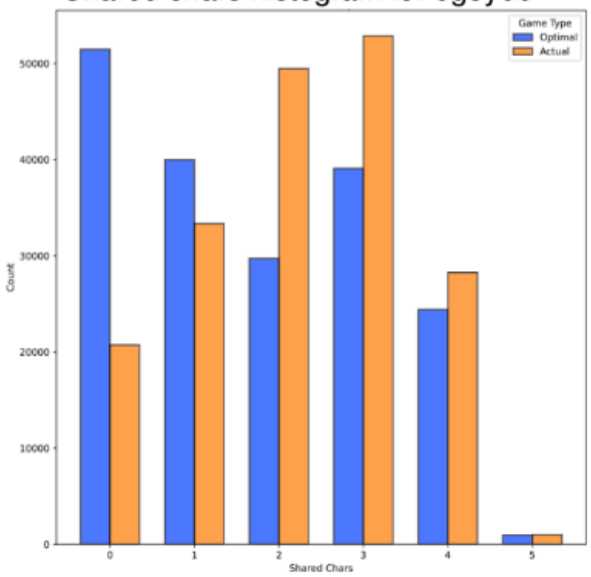
Shared chars histogram for 0g0y5b



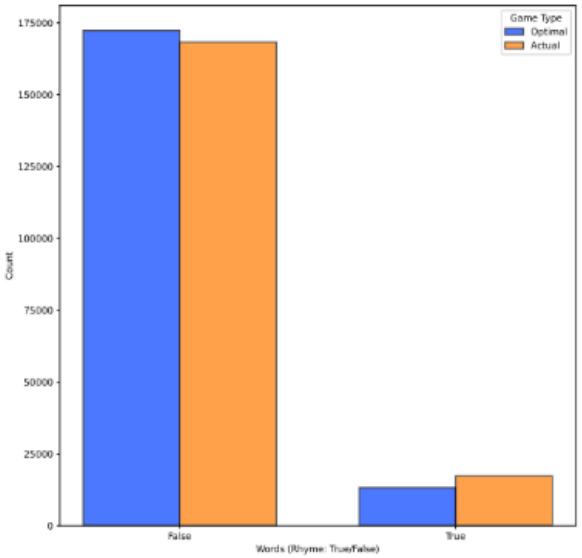
Shared chars histogram for 0g5y0b



Common syllables histogram for 3g0y2b



Total shared chars histogram



Proportion of rhyming guesses

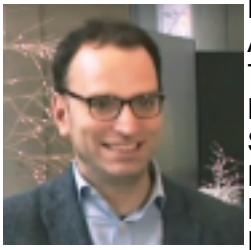
Jiadong (Gary) Liang is a third-year undergraduate student at the University of Toronto, majoring in Machine Intelligence within the Engineering Science program. His research interests lie in machine learning, optimization, and reinforcement learning, with a focus on both theoretical foundations and practical applications of AI.

Adam Kabbara is a second-year engineering Science student majoring in Electrical and Computer Engineering at the University of Toronto. His interests lie in artificial intelligence, embedded systems, and applied machine learning, especially in robotic systems. His work spans research, engineering for social impact, and leadership roles in design teams.

Jiaying (Cindy) Liu is pursuing a Bachelor's degree in the Engineering Science program at the University of Toronto, specializing in Machine Intelligence. Her research interests encompass machine learning, mathematical optimization, and operating systems.

Ronaldo Luo is a third year Engineering Science undergraduate student majoring in machine intelligence at the University of Toronto. He is interested in the general area of machine learning and especially in reinforcement learning and AI safety.

Kina Kim is a Db2 Cloud SWE at IBM and a recent graduate from the University of Toronto in Machine Intelligence. Her undergraduate thesis focused on optimal trajectory planning for unmanned ground vehicles using RL and GNN. Her research interests surround AI applications, using Graph and RL approaches for better human-AI interaction.



Michael Guerzhoy is an Assistant Professor, Teaching Stream in the Division of Engineering Science and the Department of Mechanical & Industrial Engineering at the University of Toronto. His principal interests are in teaching introductory computer science and machine learning, as well as in research in applied ML and neural network interoperability.

An N-Gram Framework for Sentiment and Emotion-Aware Word Association Games

Authors

Rohan Dalal (Pennsylvania State University; rxd5491@psu.edu)
Sanjana Menon (Pennsylvania State University; ssm5808@psu.edu)
Sohan Hajra (University of Illinois at Urbana-Champaign; shajra2@illinois.edu) **Jeremy Blum, D.Sc.** (Pennsylvania State University; jjb24@psu.edu)
DOI: 10.1145/3774399.3774407

Abstract

Natural Language Processing (NLP) models have seen impressive advancements in understanding word associations. Still, limited attention has been given to user sentiment and emotions influencing these associations. In this paper, we explore the impact of both sentiment and emotion on the selection of words in an N-gram-based word association game. By integrating GloVe embeddings with SenticNet-based emotion classification and sentiment analysis from VADER, we evaluate how positive and negative sentiments, combined with intense emotions such as delight, ecstasy, terror, and loathing, affect word choices. Our findings suggest that user sentiment and emotion have a significant effect on word selection, with positive players associating positive adjectives with positive nouns, while negative players tend toward the opposite associations. This study highlights the necessity of considering sentiment and emotional intelligence in future NLP systems and presents new applications in areas such as AI-based gaming, behavioral analysis, and human-computer interaction.

Introduction

Word association games are a form of entertainment that challenges players to think creatively while considering the emotional and conceptual context of the words involved. In Apples-to-Apples, players match descriptive adjectives with nouns with the goal of having their selected adjective chosen by a designated judge. The open-ended nature of these games presents a challenge in terms of optimizing gameplay strategy, particularly in how players make decisions based on the emotional and semantic context of words. In particular, the goal is to select words that align with the given target adjective in a way that reflects both semantic similarity and emotional resonance with the judge. This study focuses on the development of an approach that selects words for word association games by simulating a player that can account for sentiment, emotional context, and semantic relationships. This requires not only an understanding of word meanings but also the emotional nuances and subjective preferences of individual players.

The objective of the Refined NGram-based Player (RNP) model is to act as a dynamic player that uses sentiment analysis and emotion classification to align card choices with the given adjective to reflect user biases and emotional profiles. The model uses Valence Aware Dictionary and sEntiment Reasoner (VADER) for sentiment analysis, SenticNet for emotion classification, and GloVe word em beddings to measure semantic similarity between

words, enhancing its ability to select more relevant words based on the target adjective. Through preset tests that reflect diverse user profiles, the model's ability to select nouns that are most likely to align with target adjectives is evaluated. The results showed a measurable improvement in card selection alignment.

Related Work

Research in sentiment analysis, semantic similarity, and artificial intelligence has made significant strides in linguistic and emotion analysis. Our work aligns with studies examining how sentiment and emotion interact with AI-driven models, particularly in contexts requiring human- like decision-making.

As Hutto and Gilbert (2014) note, "Sentiment analysis is useful to a wide range of problems that are of interest to human-computer interaction practitioners and researchers, as well as those from fields such as sociology, marketing and advertising, psychology, economics, and political science" (p. 216). Traditional approaches have predominantly relied on rule-based models and machine learning classifiers. For instance, Zainudin et al. (2019) highlighted the potential of K Nearest Neighbors classifiers (KNN) to out perform Support Vector Machines (SVMs) in precision and recall metrics for sentiment classification tasks. Pang and Lee (2008) explored opinion mining, focusing on challenges like domain dependency, sentiment polarity, and subjectivity detection. Their work emphasized the inherent complexities of sentiment analysis, especially in disambiguating nuanced expressions across diverse contests. These studies focused on relatively static con texts, such as reviews or social media posts. Their integration into dynamic, emotion-driven systems like word association games remains underexplored.

This work also builds on previous work by incorporating VADER, a rule-based sentiment analysis model optimized for short text, such as tweets. VADER has been shown to out perform traditional models, including Linguistic Inquiry and Word Count, in microblogging environments due to its nuanced handling of intensity modifiers, negations, and emoticons (Hutto & Gilbert, 2014). Integrating VADER enables the proposed RNP model to capture sentiment polarity (positive, negative, neutral) with high accuracy.

To further enhance the emotional understanding of words, the RNP model integrates Sen ticNet (Cambria et al., 2016), a concept level sentiment analysis tool that links natural language concepts to fine-grained emotional states such as delight, serenity, and terror. Unlike traditional sentiment analysis frameworks that rely solely on word-level analysis, SenticNet combines cognitive and affective in formation, leveraging commonsense reasoning and psychology to understand both the semantics and connotations of multi- word expressions. By dynamically extracting and applying emotional profiles, SenticNet enables our model to adjust card selection based on the player's emotional context, improving personalization and fostering human-like decision-making.

Word embeddings allow for the quantification of semantic relationships between words. GloVe (Global Vectors for Word Representation) captures co-occurrence statistics, enabling it to model both local and global word contexts (Pennington et al., 2014). Unlike traditional frequency-based methods, GloVe creates dense vector representations that facilitate fine-grained similarity assessments. This is particularly relevant for our study, as semantic alignment between adjectives and nouns is a critical factor in word association games. Despite other alternatives like Word2Vec and FastText offering similar capabilities, GloVe was chosen for its effectiveness in smaller datasets and its pre-trained vectors' ability to generalize across linguistic domains. By integrating GloVe embeddings, our model ensures that selected word pairs maintain both semantic coherence and emotional relevance.

Finally, studies on bigrams provide relevant context for our work. For example, Nguyen et al. (2016) explored bigram entropy analysis to detect significant events on social media, demonstrating how word pair

distributions reveal insights into shifts in meaning and social dynamics. Similarly, our model uses bigrams to assess the alignment between adjectives and nouns, ensuring that player choices are both semantically and emotionally consistent with the target words.'

Model Details

The Refined NGram-based Player (RNP) model enhances word association games by integrating semantic similarity, sentiment analysis, and emotion classification to simulate human-like decision-making. The model selects noun cards that align with a given target adjective while accounting for both the semantic meaning and the emotional preferences of the user.

The development of the model relied on two key datasets: an adjective-noun dataset, compiled from open-source linguistic resources, and a user tweets dataset, curated to capture diverse user sentiment profiles categorized as positive, neutral, and negative.

Datasets

Our research involved two key datasets. An adjective-noun dataset was used as a basis for determining the semantic similarity between words and a curated user tweets dataset which was used for analyzing potential judge sentiments. We compiled CSV files of 4,844 adjectives (Siem, 2019) and 6,800 nouns (Leite, 2018) from Kaggle to be used in the word association game. These datasets were preprocessed to remove duplicates and ensure variety.

To create a robust and reproducible dataset for sentiment and emotion analysis, we curated tweets from publicly accessible celebrity accounts due to their expressive language and accessibility. We began by selecting 12 users to represent three sentiment profiles: positive, negative, and neutral. Sentiments were formally defined using VADER sentiment scores, where tweets with a compound score greater than 0.05 were labeled positive, between - 0.05 and 0.05 as neutral, and less than - 0.05 as negative. For each user, we initially collected 10 tweets and analyzed their sentiment scores to select the 5 best candidates for evaluation and testing purposes—2 users with predominantly positive sentiment, 2 with negative sentiment, and 1 with a neutral profile. To refine each sentiment profile, we manually gathered 20 recent tweets (from the past 5 years) per user, ensuring they were in English for uniformity. From these, we randomly sampled 10 tweets per user to build a balanced and representative sentiment profile.

Card Selection Process

The card selection process, depicted in Figure 1, is a multi- step approach that integrates semantic similarity, sentiment analysis, emotion classification, and bigram validation to make intelligent and user-aligned decisions. The objective is to select the noun card that most closely aligns with the target adjective while accounting for the user's sentiment and emotional profile. This process enables the model to simulate human-like decision-making in word association games.

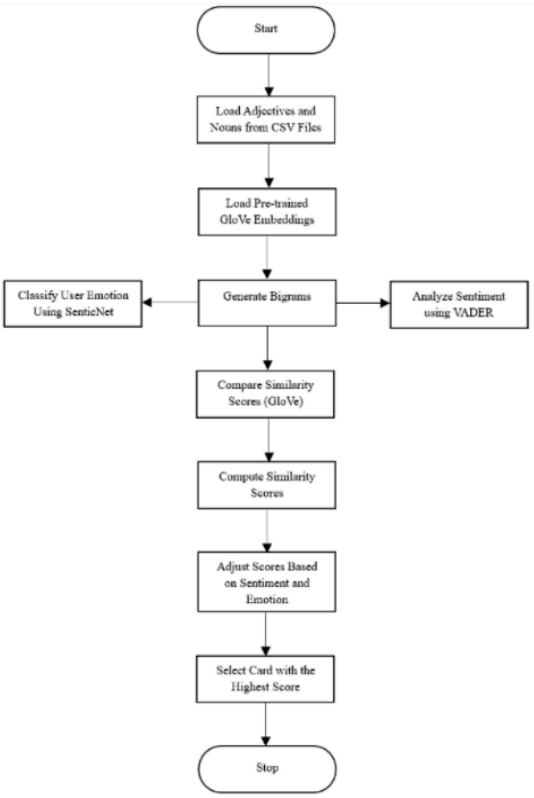


Figure 1: The overall architecture of the Sentiment Analysis Model.

The first step involves computing semantic similarity between the target adjective and

each card in the player's hand using GloVe word embeddings. These pre-trained embeddings map words into a high-dimensional vector space, allowing the model to quantify the semantic closeness between words. The similarity score for each card is calculated using the cosine similarity function, which measures how closely the vector representation of the adjective aligns with the noun. To emphasize the importance of contextual relevance in word associations, this similarity score is amplified by a factor of two—ensuring that cards with higher semantic alignment are prioritized.

Once the semantic similarity scores are established, the model incorporates sentiment analysis using VADER (Valence Aware Dictionary and sEntiment Reasoner). VADER assigns a compound sentiment score to each noun card and the target adjective, classifying the words as positive, neutral, or negative. The user's sentiment profile, derived earlier through VADER analysis of their tweets, is then compared with the sentiment of the cards. The model dynamically adjusts scores based on this alignment. For instance, if the user has a positive sentiment and the target adjective is also positive, the model assigns a higher score to noun cards carrying positive sentiments. Conversely, when a negative sentiment user interacts with a negative adjective, cards with stronger negative connotations are favored. This step personalizes card selection to resonate with the user's sentiment bias. The model further refines the selection process by incorporating emotional context using Sentic Net. Each user is assigned a dominant emotional state (e.g., ”delight,” ”terror,” or ”serenity”) that influences the decision-making process. Cards that align with the user's emotional state are given a boost in score, while cards that conflict with the emotion receive a penalty. Strong emotions, such as ”ecstasy” or ”loathing,” exert a greater influence, amplifying their effect on the scores, whereas milder emotions like ”serenity” lead to more subtle score adjustments. This ensures that the model reflects not only the user's sentiment but also their emotional intensity.

To ensure linguistic validity, the model verifies whether the selected noun card forms a valid bigram with the target adjective. This validation is performed using Part-of-Speech (POS) tagging to confirm that the first word is an adjective and the second is a noun. Additionally, WordNet synset validation ensures that the chosen noun is contextually relevant to the adjective. If the bigram passes both checks, the score for the card receives a further boost, enhancing its likelihood of being selected.

The final step involves aggregating the scores from all the aforementioned components—semantic similarity, sentiment alignment, emotional context, and bigram validation. The noun card with the highest total score is selected as the optimal match for the target adjective. In cases where no card stands out or multiple cards receive identical scores, the model defaults to selecting a random card to ensure smooth and continuous gameplay.

Experimentation and Testing

The experimentation and testing phase of this study was divided into two stages to comprehensively evaluate the effectiveness of the RNP model in a word association game context. The first stage focused on benchmarking the Base RNP Model against traditional approaches for card selection, including models like Word2Vec, Random selection, EditDistance, and SpaCy. This comparison aimed to assess the Base RNP Model's ability to integrate semantic similarity, sentiment analysis, and emotional alignment in decision making relative to these established methods. The second stage of testing introduced personalized sentiment profiles—Positive, Negative, and Neutral—by modifying the Base RNP Model to account for user-specific emotional and sentiment contexts. This stage evaluated the model's ability to adapt its card selection process to mimic user personas and align closely with their emotional profiles. Together, these stages aim to provide a holistic evaluation of the model's performance, both as a general-purpose AI-driven player and as a personalized, emotion-aware participant in the game.

Base RNP Model

The Base RNP Model in our study serves as a foundational approach to card selection in word association games, relying on a combination of semantic similarity and static rules to make decisions. Unlike models integrated with user data, the Base RNP Model operates without personalization, applying generic sentiment scores derived from VADER and focusing solely on static attributes of the words. While it calculates semantic similarity using GloVe embeddings to ensure contextual relevance between adjectives and nouns, it does not account for user-specific emotional or sentiment-driven nuances. The absence of dynamic scoring and adaptability renders it a more traditional, rule-based solution, making it ideal for baseline comparisons. This approach makes the model a static decision-maker, where its output is determined by the pre-defined logic and the hand of cards, without any variability based on user-specific inputs. The general-purpose framework of this model highlights the limitations of non-personalized models and serves as a benchmark to evaluate the enhanced performance of sentiment and emotion-aware models, which dynamically adjust scoring and adapt to user profiles.

RNP Model with User Sentiment Profiles

The RNP Model with User Sentiment Profiles builds upon the Base RNP Model by integrating user-specific sentiment and emotional data to dynamically tailor its decision- making process. Unlike the static approach of the Base RNP Model, this advanced model personalizes card selection by leveraging user sentiment (positive, negative, or neutral) and dominant emotional states (e.g., delight, terror, serenity). Through dynamic weighting, the model adjusts scoring based on the alignment between the user's sentiment and the target adjective's sentiment. For instance, when the user sentiment is positive, the model prioritizes cards with positive sentiment for positive adjectives while penalizing negative ones. Conversely, for a user with negative sentiment, the model reverses this approach, favoring negative cards for negative adjectives.

Additionally, the model incorporates emotion specific adjustments, where dominant user emotions further influence scoring. High arousal emotions like delight or loathing amplify the selection bias, either favoring or penalizing specific cards depending on the con text, while neutral emotions like serenity have subtler effects. This emotion-aware bias enhances the model's ability to mimic human-like decision-making by considering both the semantic relevance of the word pairings and the user's emotional context. By reweighting semantic similarity, sentiment alignment, and emotional biases, the model creates a personalized and context-sensitive experience. Over multiple runs, the model does not ”train” in the machine learning sense (i.e., it doesn't update its parameters or learn from past decisions). However, the presence of a user specific sentiment profile makes the decision making process appear tailored for the user. As a result, this model behaves as a personalized decision-maker where every user will likely get a different outcome based on their profile, even with the same target adjective and hand of cards.

First Stage of Testing

The first stage of testing evaluates the effectiveness of the Base RNP Model against traditional card selection models by using a curated dataset of adjective-noun pairs. After creating the finalized user tweets dataset, which provided sentiment and emotion profiles for selected users; we manually generated 10 target adjectives and corresponding hands of cards for each of the six emotions: delight, ecstasy, serenity, enthusiasm, terror, and loathing. Each target adjective represented a specific emotion, and the hands of cards consisted of nouns that varied in sentiment and semantic similarity to the target adjective.

To establish a ground truth for the testing phase, we categorized these six emotions into positive, neutral, and negative sentiment groups based on psychological and linguistic standards. Positive emotions include de light, ecstasy, and enthusiasm, as they

convey happiness, excitement, and positivity. Neutral emotions, such as serenity, represent calmness or a lack of strong sentiment. Negative emotions, including terror and loathing, were defined by their association with fear, disgust, or negativity. This classification ensured a clear and consistent framework for evaluating the model's performance, as the ground truth determined whether the selected card aligned with the target adjective's emotional and semantic context. A total of 60 examples (10 per emotion) were compiled into a CSV file, providing a structured and reproducible basis for evaluation.

The goal of the testing against the structured dataset was to assess the extent to which the Base RNP Model could align its selections with the target emotion and sentiment, providing a benchmark for comparing it with other traditional models. This approach highlighted the importance of sentiment and emotional understanding in creating contextually accurate and human-like gameplay

In this stage, we evaluated the performance of the Base RNP Model by comparing it with the following models and each model had its unique approach for selecting cards given a target adjective and hand of cards (nouns):

1. Word2Vec model: Leverages pre-trained Word2Vec embeddings to calculate the semantic similarity between the target adjective and nouns in the hand. The card with the highest similarity score is selected, ensuring contextually relevant choices.
2. SpaCy Model: Utilizes SpaCy's large English language model for word embeddings and integrates Hugging Face's transformers for sentiment analysis. This model incorporates humor-based decision-making for a more dynamic and engaging approach.
3. Random Model: Makes entirely arbitrary selections, offering no semantic or contextual basis for its choices.
4. EditDistancePlayer Model: Chooses the card with the smallest Levenshtein distance to the target, prioritizing lexical closeness.

The testing procedure involved extracting examples from the curated dataset, consisting of predefined emotions, corresponding target adjectives, and their associated noun combinations (hand of cards).

Each model was run across all 60 target adjectives and their respective hand of cards (noun combinations). The chosen card, sentiment score, and emotion for each example were stored, generating a dataset, which was further used for modelling the results.

To evaluate the performance of each model, we utilized Root Mean Squared Error (RMSE) as the primary metric. For this study, we assigned a sentiment value of +1 to positive emotions and -1 to negative emotions. The model- generated sentiment score was then compared against these ground truth values to calculate RMSE.

Model Accuracy

The models were evaluated for accuracy on the test set of curated examples. Figure 2 shows the RMSE values for each model across the six emotions: delight, ecstasy, serenity, enthusiasm, terror, and loathing. The plots illustrate that the RNP model consistently outperformed the others in positive emotions like delight and ecstasy, with notably lower RMSE values. However, the performance dropped for negative emotions such as loathing, especially for negative sentiment users. The RNP model performed best overall, especially in categories with extreme positive emotions.

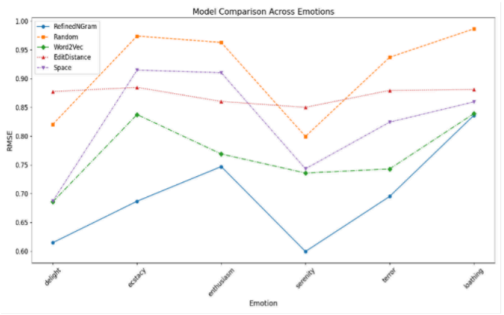


Figure 2: Performance of the models as a function of the underlying emotions.

Second Stage of Testing

The second stage of testing focused on assessing the ability of the RNP with user sentiment profiles to adapt its card selection process to personalized user sentiment profiles. This stage directly addressed our research question: can the model mimic user sentiment in word association games to provide more human-like and engaging gameplay? To achieve this, we introduced five personalized sentiment profiles derived from a curated user tweet dataset (refer to Section 3.1.2), representing three distinct sentiment orientations:

- Positive sentiment profiles: Two users with consistently positive sentiment scores, e.g. Taylor Swift.
- Negative sentiment profiles: Two users with consistently negative sentiment scores, e.g. Piers Morgan.
- Neutral sentiment profile: One user with neutral sentiment scores, e.g. Rihanna.

The model was modified to integrate these sentiment profiles by dynamically adjusting scoring based on user sentiment and emotion. The personalized RNP Model with user sentiment profiles were then compared to the base RNP Model to evaluate the impact of sentiment personalization on decision-making.

Using the same dataset of target adjectives and noun combinations generated in Stage 1, we calculated the sentiment scores for the personalized models and the Base RNP Model across all six emotions: delight, ecstasy, serenity, enthusiasm, terror, and loathing. RMSE values were computed to measure the accuracy of each model's sentiment alignment relative to the ground truth. The inclusion of user-specific sentiment profiles allowed us to evaluate how effectively the RNP Model could refine its selections to align with user sentiment, emotional context, and the semantic relevance of target adjectives.

The results from this stage formed the basis for further analysis, providing insights into the benefits of integrating user sentiment data. By comparing the personalized models against the non-personalized Base RNP Model, we demonstrated the model's potential to enhance user engagement and decision-making in word association games through sentiment aware personalization. This stage highlighted the adaptability and context-sensitivity of the RNP Model with User Sentiment profiles, showcasing its ability to emulate human-like decision-making in emotionally dynamic scenarios.

Evaluation of Model Variability

The RNP model exhibits more consistency in its selections. The box plot of RMSE distribution in Figure 3 highlights the variability in each model's performance, showing which models are consistent and which exhibit greater fluctuation.

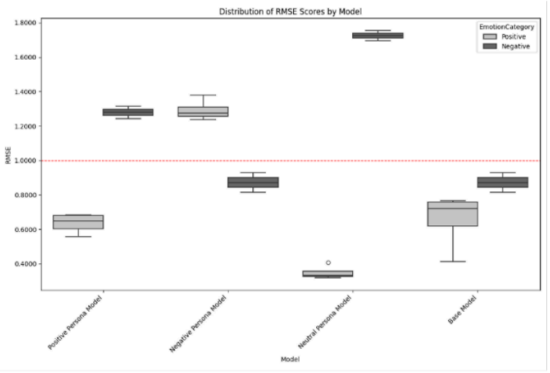


Figure 3: Root Mean Squared Error Distribution across all models and emotions.

As seen in Figure 3, the RNP model has a relatively narrow RMSE distribution, reflecting its consistency in predicting across different emotional categories. In contrast, the Random and EditDistance models exhibit much wider distributions, indicating that their predictions are less reliable. This reinforces earlier observations about the need for models to handle sentiment and semantic context better, particularly in high-variance categories.

Figure 3 illustrates the distribution of RMSE scores for the RNP Model with User Senti ment profiles and Base RNP Model across both positive, neutral, and negative emotional categories during the second stage of testing. The RNP Model with User Sentiment profiles displays a relatively narrow range of RMSE values for both

positive and negative emotions, indicating its robustness and consistency in adapting to user sentiment profiles. This aligns with the model's design to account for semantic, sentiment, and emotional factors during decision-making.

The red dashed line in the plot serves as a reference threshold for acceptable RMSE values. While the RNP Model with User Sentiment profiles consistently performs below this threshold, indicating higher accuracy, the Base RNP Model frequently exceeds it, highlighting their limitations in predicting sentiment-aligned outcomes. The stark contrast between the models underscores the importance of incorporating sentiment and emotion-aware mechanisms in achieving reliable and human-like decision- making in word association games.

Limitations

While the RNP model demonstrated significant strengths in handling positive emotions, its performance in dealing with negative emotions such as loathing and terror was noticeably weaker. This suggests that the model's sensitivity to negative sentiment could be enhanced, particularly in distinguishing between nuanced negative emotions. The difficulty in managing these emotions limits the model's overall applicability in scenarios where users may exhibit more complex negative emotional states.

Additionally, the relatively small dataset, 5 users with 10 tweets each, may limit the model's generalizability. A larger and more diverse dataset would likely offer better insights into the full range of emotional expressions and improve the robustness of the model in real-world applications. The limited dataset makes it challenging to fully assess the model's scalability and adaptability to broader, more diverse user groups.

Finally, the use of static GloVe embeddings for measuring semantic similarity may not fully capture the dynamic contextual nuances present in more sophisticated word relationships. Dynamic embeddings could potentially provide a richer, more accurate understanding of both sentiment and semantics, leading to improved performance in the word association game.

Conclusion and Future Work

This study presented a novel approach to incorporating user sentiments into word selection for word association games, leveraging the RNP model to enhance gameplay personalization and alignment with emotional contexts. The experimentation and testing phases highlighted the model's ability to integrate semantic similarity, sentiment analysis, and emotional profiling into decision-making, offering both strengths and areas for improvement.

The RNP model demonstrated strong performance in handling positive emotions such as delight and ecstasy, consistently aligning with user sentiment and enhancing semantic coherence. However, the model struggled with negative emotions like loathing and terror, revealing limitations in adapting to nuanced negative sentiment profiles. This inconsistency underscores the need for further refinement in handling complex emotional states. While the inclusion of user personas highlighted the model's potential to personalize gameplay, it also exposed gaps in performance balance across different emotional spectrums.

The results suggest that sentiment-aware AI models like the RNP model have the potential to enhance user engagement by tailoring gameplay to emotional contexts. However, user engagement was not directly evaluated in this study. While our results indicate potential for improving interactivity, future studies should explicitly measure user engagement through qualitative or quantitative evaluations, such as user surveys or gameplay metrics, to validate this claim. Additionally, the dataset size was a limitation of this study. A larger and more varied dataset, including real- world user input, would improve the model's generalizability and robustness. Advanced transformer-based architectures like BERT (Devlin, Chang, Lee, & Toutanova, 2018) and GPT (Radford & Narasimhan, 2018) could further enhance performance by capturing dynamic contextual relationships and enabling richer sentiment and semantic

analysis. Future work could incorporate these models alongside multi-modal data sources such as images or videos to expand the emotional scope of gameplay. Finally, integrating real-time sentiment and emotion analysis from platforms like Twitter into word association games presents an exciting avenue for research. This could enable more responsive and emotionally intelligent game play, enhancing personalization and engagement. Expanding the applications of such models to education or mental health contexts could also offer significant societal benefits, demonstrating the versatility and impact of emotion-aware AI.

References

Kaur, Sumandeep, Geeta Sikka, and Lalit Kumar Awasthi. 2018. ”Sentiment Analysis Approach Based on N-Gram and KNN Classifier.” In 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), 38:1–4. IEEE.

Hutto, C., and Eric Gilbert. 2014. ”VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text.” Proceedings of the International AAAI Conference on Web and Social Media 8 (1): 216–25. https://doi.org/10.1609/icwsm.v8i1.14550

Wilson, Theresa, and Janyce Wiebe. n.d. ”Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis.” Aclan thology.org. Accessed September 16, 2024. https://aclanthology.org/H05-1044.pdf.

Erik Cambria, Daniel Olsher, Dheeraj Rajagopal. 2014. ”SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-Driven Sentiment Analysis.” Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pages 1515-1520.

Kanna Akella, N. Venkatachalam, K. Gokul, Keunho Choi and Ramachandraprabhu Tyakal. 2017. ”Gain Customer Insights Using NLP Techniques.” SAE International Journal of Materials and Manufacturing, pages 333-337.

Rahim Dehkharghani, Yucel Saygin, Berrin Yanikoglu and Kemal Oflazer. 2015. ”SentiTurkNet: a Turkish polarity lexicon for sentiment analysis.” Language Resources and Evaluation, pages 667-685.

Preslav Nakov, Sara Rosenthal, Svetlana Kiritchenko, Saif M. Mohammad, Zornitsa Kozareva, Alan Ritter, Veselin Stoyanov and Xiaodan Zhu. 2016. ”Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts.” Language Resources and Evaluation, pages 35-65.

Vishakha Joseph, Chandra Prakash Lora, Narmadha T. 2024. ”Exploring the Application of Natural Language Processing for Social Media Sentiment Analysis.” 2024 3rd International Conference for Innovation in Technology (INOCON), pages 1-6.

Erik Cambria, Soujanya Poria, Rajiv Bajpai, Bjoern Schuller. 2016. SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives. The COLING 2016 Organizing Committee, pages 2666–2677.

Siem, J. (2019). Adjectives List [Data set]. www.kaggle.com/datasets/jordansiem/ adjectives-list

Leite, M. (2019). List of Nouns [Data set]. www.kaggle.com/datasets/leite0407/list-of nouns

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional Transformers for lan guage understanding. In arXiv [cs.CL]. https://doi.org/10.48550/ARXIV.1810.04805

Radford, A., & Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1–135. https://doi.org/10.1561/1500000011

Cambria, E., Zhang, X., Mao, R., Kwok, K., & Chen, M. (2024). SenticNet 8: Fusing emotion AI and commonsense AI for interpretable, trustworthy, and explainable affective computing. In Proceedings of the 2024 Conference on Affective Computing and Intelligent Interaction (ACII). College of Computing and Data Science, Nanyang Technological University.

Rohan Dalal is a student at the Pennsylvania State University majoring in Applied Data Sciences. His academic interests include artificial intelligence, machine learning, and natural language processing. He enjoys exploring the intersection of technology and human behavior, with a strong focus on data driven insights and socially responsible AI applications.

Sanjana Menon is a student at the Pennsylvania State University, majoring in Computational Data Science. Her research interests include natural language processing, sentiment analysis, and computational approaches to human decision-making. She is also interested in the ethical implications of AI, particularly in socially sensitive applications such as evaluation systems.

Sohan Hajra is a student at the University of Illinois at Urbana-Champaign, majoring in Mathematics and Computer Science. His research interests span the development of intelligent financial technologies, including algorithmic trading strategies, fraud detection models, and decentralized finance (DeFi) protocols. He is also passionate about transforming extended reality (XR) through advanced spatial computing and interactive perception algorithms, as well as advancing natural language processing (NLP) for real-world applications in information retrieval, conversational AI, and semantic understanding.

Dr. Jeremy Blum is an associate professor of Computer Science and Chair of the Mathematics and Computer Science programs at the Pennsylvania State University, Harrisburg. Dr. Blum received a D.Sc. in Computer Science and an M.S. in Computational Sciences, both from the George Washington University, as well as a B.A. in Economics from Washington University. His research interests include computer science education and transportation safety.

Conference Reports

DOI: 10.1145/3774399.3774400

One of SIGAI's missions is to promote and support AI-related conferences. Here, we report on the proceedings of recent events sponsored or run in cooperation with ACM SIGAI, up until the end of January 2025. Reports from February 2025 onwards will be published in the next issue. Members receive reduced registration rates to all affiliated conferences. These reports are based on submissions by the conference organisers.

Image Credits: “Data Mining 1” — Hanna Barakat & Archival Images of AI + AIxDESIGN,https://betterimagesofai.org / https://creativecommons.org/licenses/by/4.0/

7th AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society

San José, USA, October 21-23 2024

https://www.aies-conference.com/2024/

The AIES conference aims to encourage talented scholars from Computer Science, Law and Policy, the Social Sciences, Ethics, Philosophy, and related fields to discuss the best work related to the intersection of AI, Ethics, and Society. The conference also explicitly welcomes disciplinary experts who are newer to this topic, and see ways to break new ground in their own fields by thinking about AI.

Keynote Lectures

- Danah Boyd, Microsoft Research, Georgetown University
- David Danks, University of California San Diego
- Diyi Yang, Stanford University

Best Paper
Michael Feffer, Anusha Sinha, Wesley H. Deng, Zachary C. Lipton and Hoda Heidari: *Red-Teaming for Generative AI: Silver Bullet or Security Theater?*

Best Paper Runner-Up
Arianna Manzini, Geoff Keeling, Lize Alberts, Shannon Vallor, Meredith Ringel Morris and Iason Gabriel: *Beyond Interaction: Investigating the Appropriateness of Human-AI Assistant Relationships*

Best Student Paper
Andrea W Wen-Yi, Kathryn Adamson, Nathalie Greenfield, Rachel Goldberg, Sandra Babcock, David Mimno and Allison Koenecke: *Automate or Assist? The Role of Computational Models in Identifying Gendered Discourse in US Capital Trial Transcripts*

Best Student Paper Runner-Up
Rui-Jie Yew, Lucy Qin and Suresh Venkatasubramanian: *You Still See Me: How Data Protection Supports the Architecture of AI Surveillance*

The *Association for the Advancement of Artificial Intelligence* and the *Association for Computing Machinery* organised AIES 2024, and it was hosted by the *Markkula Center for Applied Ethics* at *Santa Clara University.* The conference was sponsored by *Google DeepMind, Google, Google Salesforce, Sony AI, IBM, eBay,* and *JPMorgan Chase*.

AIES 2024 also featured a student program with one-on-one meetings and lunches with senior scholar mentors, as well as the opportunity to present dissertation research in posters alongside main track posters. The student participants from all over the world expressed very positive reactions to the quality of the program and the resources provided. The student track was sponsored by the *ACM SIG on Artificial Intelligence* and the *National Science Foundation.*

The proceedings of AIES 2024 are here: https://ojs.aaai.org/index.php/AIES/issue/view/609

AIES 2025 will take place in Madrid, Spain, October 20-22.

39th IEEE/ACM International Conference Automated Software Engineering

Sacramento, USA, October 27 - November 1, 2024

https://conf.researchr.org/home/ase-2024

ASE is the premier research forum for Automated Software Engineering. Each year, it brings together researchers and practitioners from academia and industry to discuss foundations, techniques, and tools for automating the analysis, design, implementation, testing, and maintenance of large software systems.

Keynote Lectures

The conference featured three excellent keynote talks, all dealing with different aspects of generative models in software engineering. Prof. Koushik Sen (UC Berkeley, US) talked about how LLMs affected symbolic execution and test generation for smart contracts. Prof. Corina Pasareanu (CMU, US) presented how multi-model perception models such as Vision Language Models can assist formal analysis and run-time monitoring of requirements. Finally, Dr. Chales Sutton (Google Deepmind, University of Edinburgh) gave a talk which explored the extent to which LLMs can "understand", rather than generate, code.

Distinguished Paper Awards

- Xihao Zhang, Yi Song, Xiaoyuan Xie, Qi Xin and Chenliang Xing: *Do not neglect what's on your hands: localizing software faults with exception trigger stream*
- Zongze Jiang, Ming Wen, Jialun Cao, Xuanhua Shi and Hai Jin: *Towards Understanding the Effectiveness of Large Language Models on Directed Test Input Generation*
- Zifan Xie, Ming Wen, Tinghan Li, Yiding Zhu, Qinsheng Hou and Hai Jin: *How Does Code Optimization Impact Third-party Library Detection for Android Applications?*
- Chuan Yan, Mark Huasong Meng, Liuhuo Wan, Tian Yang Ooi, Ruomai Ren and Guangdong Bai: *Exploring ChatGPT App Ecosystem: Distribution, Deployment and Security*
- Yin Wang, Ming Fan, Hao Zhou, Haijun Wang, Wuxia Jin, Jiajia Li, Wenbo Chen, Shijie Li, Yu Zhang, Deqiang Han and Ting Liu: *MiniChecker: Detecting Data Privacy Risk of Abusive Permission Request Behavior in Mini-Programs*
- Chengpeng Li, Abdelrahman Baz and August Shi: *Reducing Test Runtime by Transforming Test Fixtures*
- Shiwei Feng, Yapeng Ye, Qingkai Shi, Zhiyuan Cheng, Xiangzhe Xu, Siyuan Cheng, Hongjun Choi and Xiangyu Zhang: *ROCAS: Root Cause Analysis of Autonomous Driving Accidents via Cyber-Physical Co-mutation*
- Huan Xie, Yan Lei, Maojin Li, Meng Yan and Sheng Zhang: *Combining Coverage and Expert Features with Semantic Representation for Coincidental Correctness Detection*
- Elvan Kula, Arie van Deursen and Georgios Gousios: *Context-Aware Automated Sprint Plan Generation for Agile Software Development*
- Jiuang Zhao, Zitian Yang, Li Zhang, Xiaoli Lian, Donghao Yang and Xin Tan: *DRMiner: Extracting Latent Design Rationale from Jira Issue Logs*
- Muhammad A. A. Pirzada, Giles Reger, Ahmed Bhayat and Lucas C. Cordeiro: *LLM-Generated Invariants for Bounded Model Checking Without Loop Unrolling*
- Yiheng Xiong, Ting Su, Jue Wang, Jingling Sun, Geguang Pu and Zhendong Su: *General and Practical Property-based Testing for Android Apps*
- Guangyuan Wu, Weining Cao, Yuan Yao, Hengfeng Wei, Taolue Chen and Xiaoxing Ma: *LLM Meets Bounded Model Checking: Neuro-symbolic Loop Invariant Inference*
- Huan Zhang, Wei Cheng, Yuhan Wu and Wei Hu: *A Pair Programming Framework for Code Generation via Multi-Plan Exploration and Feedback-Driven Refinement*
- Shuncheng Tang, Zhenya Zhang, Jixiang Zhou, Lei Wang, Yuan Zhou and Yinxing Xue: *LeGEND: A Top-Down Approach to Scenario Generation of Autonomous Driving Systems Assisted by Large Language Models*

UC Davis California, ByteDance, and *Google* sponsored ASE 2024. The conference was also supported by the *ACM SIGAI* and the *ACM SIGSoft.*

The proceedings of ASE 2024 are in the ACM digital library: https://dl.acm.org/doi/proceedings/10.1145/3691620

ASE 2025 will take place in Seoul, South Korea, November 16-20.

4th ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization

San Luis Potosí, Mexico, October 29-31, 2024

https://conference2024.eaamo.org/

EAAMO highlights work where techniques from algorithms, optimization, and mechanism design, along with insights from the social sciences and humanistic studies, can help improve equity and access to opportunity for historically disadvantaged and underserved communities.

Keynote Lectures

- Alessandra Fogli, Federal Reserve Bank of Minneapolis: *Beliefs, Social Context, and Macro Outcomes*
- Irene Lo, Stanford University: *Mechanisms, Optimization, and Education Policy.*

Best Paper Award

Johann Gaebler and Sharad Goel: *A Simple, Statistically Robust Test of Discrimination*

Best Student Paper

Marios Mertzanidis, Alexandros Psomas and Paritosh Verma: *Automating Food Drop: The Power of Two Choices for Dynamic and Fair Food*

The conference also featured a month-long Social Hackathon, which looked at the impact of environmental risk factors on breast cancer in Mexico; a Doctoral Consortium, which offered mentoring and skill-building for students; and a panel with the Jamal Poverty Action Lab, connecting researchers and policymakers. It also included an EAAMO Bridges session, where participants explored strategies for building a research community that centers the perspectives and interests of marginalized groups in the design of algorithmic and resource allocation systems.

ACM SIGecom and *ACM SIG on Artificial Intelligence* sponsored EAAMO 2024. The following organisations co-sponsored the conference: *Columbia University,* the *University of Pittsburgh Center for International Studies and Responsible Data Science,* the *Artificial Intelligence Journal, Econmachina, COPOCYT (San Luis Potosí Council of Science and Technology), Santander Universia, Mexican National Network of Councils and State*

Organizations of Science and Technology, an Luis Potosí Municipal Government, San Luis Potosí State Government, Autonomous University of San Luis Potosí, and the *San Luis Potosí Institute of Scientific Research and Technology.* The conference was also supported y the *Association for Computing Machinery,* the *Harvard David Rockefeller Center for Latin American Studies,* and the *Harvard SEAS Office for Diversity, Inclusion and Belonging.*

ACM EAAMO 2025 will take place in Pittsburgh, USA, November 5-7.

2024 International Conference on Artificial Intelligence and Future Education

Shanghai, China, November 1-2 2024

https://www.icaife.net/

ICAIFE is a pioneering platform, uniting scholars, educators, researchers, and industry leaders from around the globe to explore the transformative role of Artificial Intelligence in shaping future educational paradigms.

Keynote Speakers

- Prof. Qinghua Zheng, Tongji University, China
- Prof. Hui Wang, Beijing Normal University, China
- Prof. Bin Zhou, The Second Affiliated Middle School of East China Normal University, China
- Liu Yuxiang, Secretary of the Party Committee and President of Shanghai Education Examination Institute
- Prof. Weimin Zheng, Tsinghua University, China
- Prof. Xingao Gong, Guangdong Technion-Israel Institute of Technology, China
- Prof. Han Yu, Education Examination Institute, Ministry of Education, China
- Prof. Yihui Zheng, Shanghai Open University, China

The proceedings of iCAIFE 2024 are in the ACM Digital Library: https://dl.acm.org/doi/proceedings/10.1145/3708394?tocHeading=heading3

ICAIFE 2025 will take place in Shanghai, China, October 24-26.

2nd International Conference on Mathematics and Machine Learning

Nanjing University, China, November 8-10, 2024

https://www.icmml.org/

ICMML is an international forum for academics, scientists, and engineers in the fields of Mathematics and Machine Learning to exchange ideas, share expertise, and discuss the challenges and future possibilities in their specialisms.

Keynote Lectures

- Wanyang Dai, Nanjing University, China: *3D AI and Spatial Generative AI via Feedback Control Strengthened Quantum Transformer*
- Lin Chen, Sun Yat-sen University, China: *On Batching Task Scheduling: Theoretical Foundation and Algorithm Design*
- Lazim Abdullah, Universiti Malaysia Terengganu, Malaysia: *A Combined Decision-Making Analysis Under Single Valued Neutrosophic Set for Selecting Knowledge Management Strategy*

Nanjing University sponsored ICMML 2024. The proceedings of ICMML are in the ACM digital library: https://dl.acm.org/doi/proceedings/10.1145/3708360

ICMML 2025 will take place in Nanjing, China, November 14-16.

16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management

Porto, Portugal, November 17-19 2024

https://ic3k.scitevents.org/

IC3K consists of three co-located conferences, in the areas of Knowledge Discovery, Knowledge Engineering, and Knowledge Management. It attracts researchers, engineers, and practitioners in these fields.

Keynote Lectures

- Nirmalie Wiratunga, Robert Gordon University, Aberdeen, United Kingdom: *Intelligent Reuse of Explanation Experiences: The Role of Case-Based Reasoning in Promoting Best Practice in Explainable AI*
- João Gama, University of Porto, Portugal: *Recent Advances in Learning from Data Streams*
- Carlo Sansone, University of Naples Federico II, Italy: *Multimodal Deep Learning in Medical Imaging*

Best Paper Award

Colin Daly and Lucy Hederman: *Learning to Rank for Query Auto-Complete with Language Modelling in Enterprise Search*

Best Student Paper Award

Elona Shatri and György Fazekas: *Knowledge Discovery in Optical Music Recognition: Enhancing Information Retrieval with Instance Segmentation*

Best Poster Award

Mahran Jazi and Irad Ben-Gal: *Federated Learning for XSS Detection: A Privacy-Preserving Approach*

Honorable Mention
Ahmed Qarqour, Sahil-Jai Arora, Gernot Heisenberg, Markus Rabe and Tobias Kleinert: *Utilizing Data Analysis for Optimized Determination of the Current Operational State of Heating Systems*

The *Institute for Systems and Technologies of Information, Control and Communication* sponsored IC3K 2024. The conference was also organized in cooperation with the *ACM SIG on Artificial Intelligence*, the *Association for the Advancement of Artificial Intelligence*, and the *Portuguese Association for Artificial Intelligence*.

IC3K 2025 will take place in Marbella, Spain, 22-24 October.

21st International Conference on Informatics in Control, Automation and Robotics

Porto, Portugal, November 18-20, 2024

https://icinco.scitevents.org

ICINCO brings together researchers, engineers and practitioners interested in the application of informatics to Control, Automation and Robotics. The conference covers topics related to Intelligent Control Systems, Optimization, Robotics, Automation, Signal Processing, Sensors, Systems Modelling and Control, and Industrial Informatics.

Keynote Lectures
- John Charles Krumm, University of Southern California, United States: *Personal Data Privacy: Especially Location*
- Antonella Ferrara, University of Pavia, Italy: *Control of Road Traffic Systems: A Multi-Scale Perspective*
- Vladimír Kucera, Czech Technical University in Prague, Czech Republic: *Youla-Kučera Parameterization: Theory and Applications*

Best Paper Award
Duarte Branco, Rui Coutinho, Armando Sousa and Filipe Dos Santos: *Subsurface Metallic Object Detection Using GPR Data and YOLOv8 Based Image Segmentation*

Best Student Paper Award
Emmanuel Alao, Lounis Adouane and Philippe Martinet: *Multi-Risk Assessment and Management in the Presence of Personal Light Electric Vehicles*

Best Poster Award
Elias August, Sigurdur Hafstein, Jacopo Piccini, Stefania Andersen and Anna Bavarsad: *Application of the Schur Complement in Sum of Squares Optimisation*

Best Industrial Paper Award

Assia Belbachir, Antonio M. Ortiz, Erik T. Hauge, Ahmed Nabil Belbachir, Giusy Bonanno, Emanuele Ciccia and Giorgio Felline: *Drone Technology for Efficient Warehouse Product Localization*

The *Institute for Systems and Technologies of Information, Control and Communication* sponsored ICINCO 2025, and it was technically co-sponsored by *IEEE SMC - TC on Evolving Intelligent Systems and International Federation of Automatic Control*. ICINCO 2024 was also organized in cooperation with the *ACM SIG on Artificial Intelligence*, the *Association for the Advancement of Artificial Intelligence*, the *Portuguese Association for Artificial Intelligence*, and the *International Neural Network Society*.

ICINCO 2025 will take place in Marbella, Spain, October 20-22.

1st International Conference on Explainable AI for Neural and Symbolic Methods

Porto, Portugal, November 20-22 2024

https://explains.scitevents.org/

EXPLAINS provides a space for knowledge exchange between different communities in AI, which are all developing explainable AI with different definitions, evaluation metrics, motivations and results. This is important, as in the future, people will collaborate more and more with machines to solve complex problems using AI techniques. Such a collaboration requires adequate communication, trust, clarity and understanding. Explainable AI aims to address such challenges by combining the best of symbolic AI and Machine Learning including neural models, evolutionary computing and fuzzy systems.

Keynote Lectures

- Tome Eftimov, Jožef Stefan Institute, Slovenia: *Trustworthy Benchmarking for Black-Box Single-Objective Optimization*
- Rita P. Ribeiro, University of Porto, Portugal: *Predictive Maintenance for Industry 4.0 & 5.0*
- Gabriela Ochoa, University of Stirling, United Kingdom: *Search Trajectories Illuminated*

The *Institute for Systems and Technologies of Information, Control and Communication* sponsored EXPLAINS 2024. The conference was also organized in cooperation with the *ACM SIG on Artificial Intelligence*, the *Portuguese Association for Artificial Intelligence,* and the *Associação Portuguesa de Reconhecimento de Padrões.*

EXPLAINS 2025 will take place in Marbella, Spain, October 22-24.

Image Credits: “Decision Tree” —Rens Dimmendaal & Johann Siemens / https://betterimagesofai.org / https://creativecommons.org/licenses/by/4.0/

16th International Joint Conference on Computational Intelligence

Porto, Portugal, November 20-22, 2024

https://ijcci.scitevents.org/

IJCCI brings together researchers, engineers and practitioners in the areas of Fuzzy Computation, Evolutionary Computation and Neural Computation. IJCCI is composed of three co-located conferences, each specialized in at least one of the aforementioned main knowledge areas.

Keynote Lectures

- Tome Eftimov, Jožef Stefan Institute, Slovenia: *Trustworthy Benchmarking for Black-Box Single-Objective Optimization*
- Rita P. Ribeiro, University of Porto, Portugal: *Predictive Maintenance for Industry 4.0 & 5.0*
- Gabriela Ochoa, University of Stirling, United Kingdom: *Search Trajectories Illuminated*

Best Paper Award

Jacob de Nobel, Diederick Vermetten, Thomas H. W. Bäck and Anna V. Kononova: *Sampling in CMA-ES: Low Numbers of Low Discrepancy Points*

Honorable Mention

Michele Vannucci, Satchit Chatterji and Babak H. Kargar: *Testing Emergent Bilateral Symmetry in Evolvable Robots with Vision*

Best Student Paper Award

Seyedamirhossein Salehiamiri, Richard Allmendinger and Arijit De: *Real-Time IoMT-driven Optimisation for Large-Scale Home Health Care Planning*

Honorable Mention

Quentin Vacher, Nicolas Beuve, Paul Allaire, Thibaut Marty, Mickaël Dardaillon and Karol

Desnos: *Hybrid Genetic Programming and Deep Reinforcement Learning for Low-Complexity Robot Arm Trajectory Planning*

Best Poster Award
Younes Boukacem, Hatem M. Abdelmoumen, Hodhaifa Benouaklil, Samy Ghebache, Boualem Hamroune, Mohammed Tirichine, Nassim Ameur and Malika Bessedik: *L-SAGA: A Learning Hyper-Heuristic Architecture for the Permutation Flow-Shop Problem*

The *Institute for Systems and Technologies of Information, Control and Communication* sponsored IJCCI 2024. The conference was also organized in cooperation with the *ACM SIG on Artificial Intelligence*, the *Association for the Advancement of Artificial Intelligence*, the *World Federation on Soft Computing*, the *International Neural Network Society,* and the *Portuguese Association for Artificial Intelligence.*

IJCCI 2025 will take place in Marbella, Spain, October 22-24.

2nd International Conference on Information Education and Artificial Intelligence

Kaifeng, China, December 20-22, 2024

www.ic-ieai.org

ICIEAI focuses on how AI can shape education, the advancement of the science and engineering research behind AI-assisted interactive learning systems, and the promotion of the widespread adoption of AI in education.

Keynote Lectures

- Benjamin W. Wah, The Chinese University of Hong Kong: *Objective Modeling of the Big-Data Problem On Perceptual Quality for Fast Interactive Multimedia Games*
- Shadiev Rustam, Zhejiang University: *Advancing Language and Culture Learning through VR: From Basic Immersion to AI-enhanced Multilingual Interaction and Content Engagement*
- Shuai Liu, Hunan Normal University: *Challenges in Classroom Behavior Recognition: Long-term Tracking and Tiny Behavior Recognition*
- Haozhe Jiang, Zhejiang University: *Thinking Styles, STEM Attitudes and Computational Thinking in The 21st Century*

ICIEAI is supported by the *ACM SIG in Artificial Intelligence* and the *South China Agricultural University.* The proceedings of ICIEAI are in the ACM digital library: https://dl.acm.org/doi/proceedings/10.1145/3724504

ICIEAI 2025 will take place in Guangzhou, China, December 12-14.

7th IEEE International Conference on Artificial Intelligence & eXtended and Virtual Reality

Lisbon, Portugal, January 27-29 2025

https://aixvr.tecnico.ulisboa.pt/

IEEE AIxVR brings together researchers and professionals across the spectrum of Artificial Intelligence, Augmented Reality, Mixed Reality, and Virtual Reality. This event serves as a global platform to foster collaboration between these diverse fields, facilitating the presentation of cutting-edge advancements, identifying emerging research avenues, and collectively shaping the future of these dynamic research domains.

Proceedings

There were two keynote speakers, one from each side of AI and XR.

- Anthony Steed: Head of the Virtual Environments and Computer Graphics group in the Department of Computer Science at University College London.
- Elisabeth André: Chair for Human-Centered Artificial Intelligence at Augsburg University

The program also featured 8 technical sessions, 9 thematic workshops, 2 special sessions, demo sessions and social events. Best full papers, short papers, and demos were awarded. The *IEEE Computer Society* sponsored IEEE AIxVR, with in-cooperation support from *ACM SIGAI, ACM In-Cooperation, ACM SIGGRAPH, ACM SIGCHI, Eurographics,* and the *Grupo Português Computação Gráfica Eurographics Portuguese Chapter.*

The proceedings of AIxVR 2025 can be found: https://ieeexplore.ieee.org/xpl/conhome/10895984/proceeding. IEEE AIxVR 2026 will take place in Osaka, Japan, January 26-28.

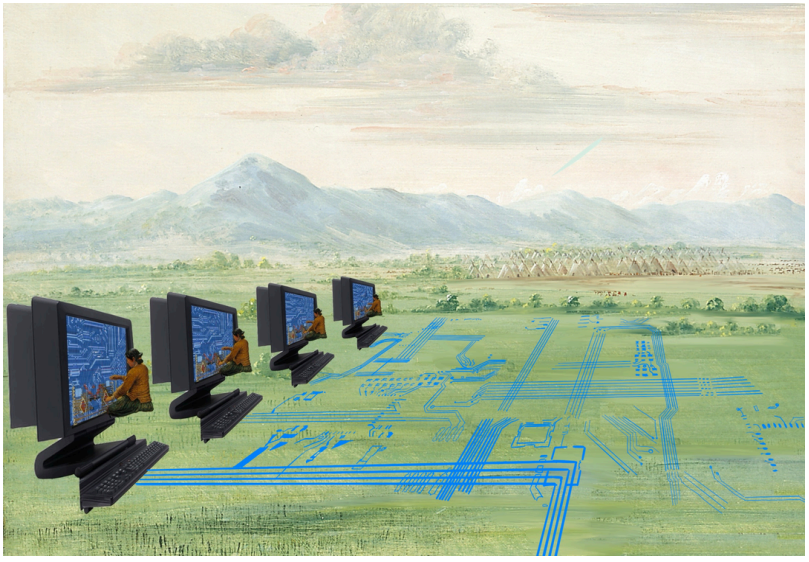


Image Credits: “Weaving Wires 2” — Hanna Barakat & Archival Images of AI + AIxDESIGN, https://betterimagesofai.org / https://creativecommons.org/licenses/by/4.0/

Conference Statistics

Conference	Number of Participants	Number of Submissions	Number of Countries Submissions Were From	Percentage of Papers Accepted / %	Type of Review
ACM EAAMO 2024	203	113	19	23.0	-
AIES 2024	-	468	-	32.1	-
AIxVR 2025	-	-	-	-	-
ASE 2024	430	557	-	26.4	-
EANMO 2024					
EXPLAINS 2024	23	19	13	26.3	Double-blind
IC3K 2024	151	175	47	21.1	Double-blind
ICAIFE 2024	500	-	-	-	-
ICIEAI 2024	70	-	-	-	-
ICINCO 2024	135	160	45	22.0	Double blind
ICMML 2024	30	-	-	-	-
IJCCI 2024	77	78	33	32.1	Double-blind