Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2023, 11, 536-547
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Dixit Player with Open CLIP

Ryan Wei

Syosset High School, New York, USA
Email: rywbook06@gmail.com

How to cite this paper: Wei, R. (2023) Dixit Player with Open CLIP. *Journal of Data Analysis and Information Processing*, **11**, 536-547.
https://doi.org/10.4236/jdaip.2023.114027

Received: September 26, 2023
Accepted: November 26, 2023
Published: November 29, 2023

Abstract

A computer vision approach through Open AI's CLIP, a model capable of predicting text-image pairs, is used to create an AI agent for Dixit, a game which requires creative linking between images and text. This paper calculates baseline accuracies for both the ability to match the correct image to a hint and the ability to match up with human preferences. A dataset created by previous work on Dixit is used for testing. CLIP is utilized through the comparison of a hint to multiple images, and previous hints, achieving a final accuracy of 0.5011 which surpasses previous results.

Keywords

Computer Vision, AI, CLIP, Dixit, Open AI, Creative Gameplay, Open CLIP, Natural Language Processing, Visual Models, Game AI, Image-Text Pairing

1. Introduction

In recent years, various board games such as chess have served as benchmarks for progress in AI. However, this research has focused primarily on logical, deterministic games, creating a void in AI research centered on creative and social gameplay [1]. We attempt to begin filling this void by creating an AI which can play the game Dixit [2].

Dixit is a complex game which demands logical, creative, and social ability. It is a challenging benchmark for the creative capabilities of AI and serves as a platform to improve models which connect images and text. In each episode of the game, a storyteller must carefully choose a card and a corresponding description for other players to base their card selections on. Each player then votes for which card they believe is the storytellers.

We attempt to build an AI agent which can accomplish the task of guessing the card which either successfully matches up to the storytellers' or matches up to the human choice. Previous work on Dixit [3] has used basic machine learning algorithms, achieving slightly better results than human counterparts on identifying the storyteller's card (which is one of the key tasks for a Dixit player). An important way in which we capitalize on this prior work is the use of the Dixit play data shared by the authors of [3].

The method achieving the best results in [3] is based on the well-established TF-IDF features. In contrast, we propose a new and more modern approach, based on computer vision and natural language processing models, namely CLIP [4] [5].

In our experiments, we consider two key tasks facing a Dixit player: identifying the storyteller's card and predicting which card in a lineup would garner most votes from other players (note that the latter may not be the same as the storyteller's card)! We show that our proposed method, based on evaluating card-to-hint relevance using "historical" play data in the training set, does better on both tasks than a number of previously proposed baselines.

2. Our Approach

2.1. Dixit

Dixit was chosen due to our belief that it was a good test of Open AI's zero-shot capabilities. The 84 cards which Dixit uses are abstract, artistically provocative paintings which often result in less literal descriptions. Hints describe an emotion evoked by the respective cards or are explained through a cultural reference.

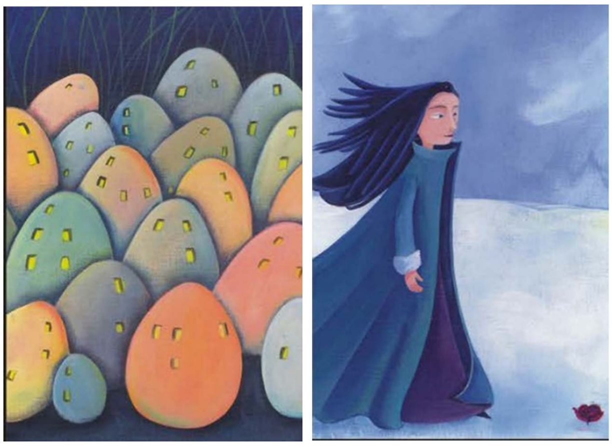
This is magnified by the scoring system. In the game, a storyteller is encouraged to generate a description which is not too obvious, but not too vague since the best outcome for their score occurs when some, not all, players guess the storyteller card. Two examples of a matching hint-card pair are displayed in **Figure 1**. Due to the nature of the game, a player must excel at the task of associating abstract/creative descriptions with the correct image.

2.2. The CLIP Model(s)

Open AI's CLIP [4] is a model which attempts to align image and text. It is trained on a large dataset (originally of 400 million text-image pairs, although subsequent efforts trained CLIP on even larger datasets) acquired from the internet, utilizing contrastive representation learning to maximize the cosine scores of the correct image-text pairs.

CLIP simultaneously trains an image encoder (mapping images to a vector in a 512-dimensional embedding space) and a text encoder (mapping text to a vector in the same embedding space). The training objective is to project the matching pairs close to each other and farther away from others. The similarity is measured by the cosine between two vectors. Note that while the training objective focuses on matching images and text, the cosine similarity can also be used to judge association between two images or two text strings. A concise summary is displayed in **Figure 2**.

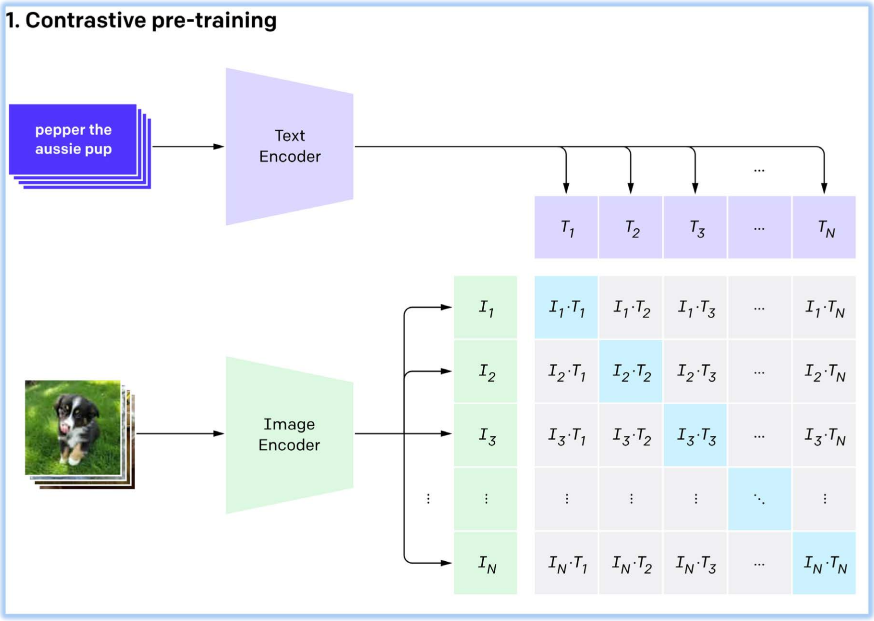
Due to the dataset and its training, CLIP has displayed impressive zero shot performance achieving state of the art image recognition abilities [4] [5].



(a)

(b)

Figure 1. (a) "Gotham City's sidekick" referring to Robin, Batman's, sidekick. Painting interpreted as robin eggs; (b) "Finally!" A look of many emotions is displayed on the girl's face as she comes across a flower.



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Figure 2. Simple summary of CLIP training, taken from [4].

However, CLIP has primarily been tested and trained on standard datasets of text-image descriptions, meaning that its ability for image recognition with more abstract/creative descriptions such as in Dixit has not been extensively evaluated.

2.3. Dataset

Our proposed approach uses the data set collected by the authors of [3]. Each "episode" provides information from a single turn of Dixit played by human players on an online platform. The data include: the storyteller's card ID, storyteller's hint, IDs of other cards played in response to the hint, and the votes received by each card. Each episode comes from a game played by 4, 5 or 6 players. The dataset includes a partition into train, validation, and test sets, with 92,981, 11,624 and 11,624 episodes respectively; we maintain this partition in our experiments.

Formally, let the set of cards played in an episode i be $\left\{c_1^i,\cdots,c_{p^i}^i\right\}$, where p^i is the number of players in the game (4, 5 or 6), and each c_j^i is an index into the 84 Dixit cards. The storyteller's hint is h^i (a text string), for the storyteller's card c_1^i. For each c_j^i we have v_j^i the fraction of the players who voted for c_j^i based on hint h^i. The episode data then includes $\left\{p^i,c_1^i,\cdots,c_{p^i}^i,h^i,v_1^i,\cdots,v_{p^i}^i\right\}$.

2.4. Hint History

For the final CLIP model, we use the hint history for each card, meaning that the hints of each episode in the training data are encoded and stored in the corresponding storyteller card list. Formally, in each episode i of the training set, the encoded hint, h^i is stored in the list corresponding to card c_1^i. After our computation, we end with a dictionary of 84 terms, each containing the embedded versions of all associated hints.

For each episode in the validation test set, containing p^i cards, hint h^i, and storyteller card c_1^i, h^i's similarity score is compared with the list of previous hints used for card c_1^i in the training set. We test a few different evaluations on the resulting scores.

Min Similarity between the card c and the hint h is determined as

$$\min_{i:c_1^i=c}\cos\left(CLIP\left(h^i\right),CLIP\left(h\right)\right) \tag{1}$$

Intuition: High minimum value indicates that no training play episode in which c was the storyteller's card had a hint that was very different from h, thus suggesting that h may be a good match to c.

Max Similarity between the card c and the hint h is determined as

$$\max_{i:c_1^i=c}\cos\left(CLIP\left(h^i\right),CLIP\left(h\right)\right) \tag{2}$$

Intuition: High maximum value indicates a training play episode in which c was the storyteller's card had a hint that closely resembled h, thus suggesting that h might be a good match to c.

Top 5 Similarity between the card c and the hint h is determined as

$$\max \sum_{\substack{i=0\\ rsorted\left(i:c_1^i=c\right)}}^{4} \cos\left(CLIP\left(h^i\right),CLIP\left(h\right)\right) \tag{3}$$

Intuition: A high sum of the top 5 values indicates multiple training play episodes in which c was the storyteller's card which had a hint that closely resembled h, thus suggesting that h might be a good match to c.

Average Similarity between the card c and the hint h is determined as

$$\sum_{i:c_1^i=i}\cos\left(CLIP\left(h^i\right),CLIP\left(h\right)\right)\Big/\left|i:c_1^i=c\right| \tag{4}$$

Intuition: High average value indicates a good overall similarity between h and all hints associated with c in training, thus suggesting that training episodes in which c was the storyteller's card had generally high similarity scores to h.

Range Similarity between the card c and the hint h is determined as

$$\max_{i:c_1^i=c}\cos\left(CLIP\left(h^i\right),CLIP\left(h\right)\right)-\min_{i:c_1^i=c}\cos\left(CLIP\left(h^i\right),CLIP\left(h\right)\right) \tag{5}$$

Intuition: A high range indicates higher maximum scores, but a lower minimum score could cause range to perform poorly. This is not an evaluation which is expected to improve the overall accuracies, it is meant to depict how influential maximum and minimum are.

3. Experiments and Results

Our approach yields successful results in comparison to baselines and other methods for both accuracy in selecting the storyteller's card and for matching up to human preferences. Card selection accuracy is displayed in **Table 1** while the accuracy of the model matching up to human behavior is displayed in **Table 2**. We also calculate the KL divergence between the distributions of card choices made in each episode of the dataset and the distributions created by the AI, displayed in **Table 3**.

3.1. Baseline Approaches

Before testing how capable open CLIP is with Dixit data, we first implement two basic strategies of selection to serve as a baseline. The first is to randomly select a card. The second is a recreation of the baseline described in Vatsakis *et al.*—the selection of the card which was most frequently chosen as the storytellers in the dataset.

Naïve CLIP

Our third baseline is a naive use of CLIP. We embedded the hint and the 4, 5, or 6 images before calculating the cosine similarities between the hint and images for each episode. This gives us 4, 5, or 6 values which we normalize into a softmax distribution. The card with the corresponding max probability in the softmax is the AI's choice.

3.2. Ability to Choose the Correct Card

The AI performs the task well, surpassing the numbers achieved by the baselines, humans, and the Vatsakis model. All accuracies are displayed, split into 5091, 1947, and 4585 episodes of 4, 5, and 6 players respectively, as well as the overall mean accuracy. The 5 different evaluations of our resultant matrix are displayed, with top 5 achieving the best accuracy. To cut down significantly on computation time, we precompute the embeddings of all 84 cards and corresponding hint histories, combined into a 512 × n_i matrix, where n_i represents the number of hints chosen for card i. This precomputation took about one hour to process,

Table 1. Accuracies of guessing the storyteller's card selection.

Method	Total	4 players	5 players	6 players
Random baseline	0.2086	0.2509	0.2025	0.1657
Frequency baseline	0.2090	0.2931	0.2332	0.2076
Naive CLIP	0.2767	0.3239	0.2661	0.2236
Human, from [3]	0.4782	0.542	0.472	0.410
Keyword model [3]	0.4042	0.443	0.407	0.360
Full model [3]	0.4793	0.523	0.488	0.427
Hint history (max)	0.4590	0.4970	0.4504	0.4207
Hint history (min)	0.2117	0.2497	0.2013	0.1740
Hint history (avg)	0.3180	0.3571	0.3063	0.2604
Hint history (top 5)	**0.4995**	0.5343	0.4972	0.4619
Hint history (max range)	0.4223	0.4614	0.4304	0.3754

Table 2. Accuracies of selecting card(s) preferred by human players.

Method	Total	4 players	5 players	6 players
Random baseline	0.2086	0.2509	0.2025	0.1657
Frequency baseline	0.2090	0.2931	0.2332	0.2076
Naive CLIP	0.3969	0.4585	0.3878	0.3324
Hint history (max)	0.5045	0.5608	0.4992	0.4443
Hint history (min)	0.2893	0.3518	0.2717	0.2275
Hint history (avg)	0.4137	0.4722	0.4299	0.3418
Hint history (top5)	**0.5647**	0.6154	0.5547	0.5125
Hint history (max range)	0.4632	0.5258	0.4520	0.3754

Table 3. KL divergence numbers.

Method	Total	4 players	5 players	6 players
Random baseline	2.299	3.302	2.196	1.229
Frequency baseline	0.2090	0.2931	0.2332	0.2076
Naive CLIP	1.361	2.019	1.282	0.6653
Hint history (max)	1.750	2.514	1.698	0.9227
Hint history (min)	2.422	3.507	2.306	1.266
Hint history (avg)	2.142	3.083	2.052	1.134
Hint history (top 5)	**1.634**	2.359	1.570	0.8561
Hint history (max range)	1.887	2.717	1.826	0.9914

cutting down the run time of testing to 10 seconds. The results for the different evaluations are displayed in Table 1.

While running on the validation test set, each hint is encoded (into a 512 × 1 vector), transposed, and multiplied with the hint history matrices of $\left\{c_1^i,\cdots,c_{p^i}^i\right\}$.

3.3. Comparison to Human Behavior

The second metric is to judge the AI's ability to match up with human behavior. Recall from the Introduction that this is related to the strategic goals of Task B. For each episode, the cards that were chosen with the highest frequency in the data were stored. If multiple cards shared the highest probability, they would all be viable choices for the AI. There is no benchmark to compare to, but more than half of the AI's choices matching up with the preferences of human players is an impressive start. Additionally, the ranking of the methods is the same as shown in Table 2.

3.3.1. KL Divergence

If we consider the goal of "replicating human judgment" by the AI player, we need to look beyond selecting the winner of the vote. The vote data available to us from [3] is indeed more detailed. For instance, if 3 players out of 6 vote for card 1, 2 vote for card 4 and 1 for card 3, then we have information beyond "card 1 is the winner"—we can also aim to estimate the full scope of human preferences.

We can treat the vote distribution in a game episode (that sums to 1 over the cards) as a "true distribution" of the human vote; intuitively, if the vote fraction for a card is $x \in [0, 1]$ we treat it as "the probability of a human player voting for this card is x". We would like the AI player to predict this probability. To this end, we compare the probability distributions generated by the AI, converted into a vote distribution summing to 1 through applying the softmax distribution, and the "human probability distributions" in the dataset through KL divergence [6]. We use KL divergence as opposed to other calculation types to tell us how much information is lost, giving us a quantifiable number. Again, the method rankings are the same, as indicated by Table 3.

3.3.2. Combining Naïve CLIP with Hint History

In order to get the best possible outcome, we combined the two methods tested with CLIP: comparing the hint to each image and comparing the hint to the hint history for each card. This would also more closely resemble a human's thought process when playing Dixit. We added the two probabilities, weighing their influences, before converting the sums into a softmax distribution again. The differences weren't significant, but there was still a small increase. We adjusted the weightings by 0.01 for 100 iterations. The best results were 0.75 and 0.25 for the naive CLIP and the hint history, respectively, raising the accuracy to **0.5020** from 0.4995.

3.3.3. Restricting Hint History

We also wanted to investigate how big of an effect the amount of training data the AI could see would have on the overall accuracy. We randomly selected 500, 250, 100, 25, 10, and 5 hints from the hint history and calculated the scores using the algorithm described in section 3.3.2. Results are displayed in Table 4.

3.4. Final Results

Table 5 displays the result of our best model, combining naive CLIP and hint history in a 0.75 and 0.25 weighing. On the validation, the overall accuracy is 0.5020 and on the test dataset, it is 0.5011, surpassing the overall accuracy of humans and Vatsakis as displayed in Table 5.

4. Discussion

In our experiments there were some surprising results. The most notable of these was that averaging the scores does relatively poorly, while range does decently well, a result which we didn't expect. To understand the accuracies, we closely examined a few specific episodes, finding one particularly illustrative of the trends in the data. In the 48th episode of the training dataset, we are given a hint of "Don't trust Fibonacci." along with 6 cards [11, 48, 12, 25, 19, 79] (displayed in Figure 3) where card 11 is the storyteller's card. The image/text matching gives us a softmax distribution of [0.2673, 0.1433, 0.2467, 0.1455, 0.1957, 0.2184] respectively. The softmax for the history matching is [1, 0, 0, 0, 0, 0]. The top 5 terms for card 11 and the 5 other cards are displayed in Table 7 and Table 8. Combining this through addition and weighing the two probabilities in a 0.75, 0.25 split, we get a final softmax distribution of [0.4505, 0.1075, 0.1850, 0.1091, 0.1468, 0.1638]. The first probability is the largest and that is the AI's answer, the correct answer.

From this data, we can draw a few conclusions. Fibonacci is a common term associated with explaining card 11 and other cards. Card 19 has 4 descriptions of "Fibonacci," but its last hint is "Golden ratio". These differences illustrate the effectiveness of the top 5 method over taking the max. For card 79 and 25, the hints don't bear much resemblance to the storyteller's hint. Other cards don't have the same volume of use of the term, "Fibonacci," while card 11 does. Extending the history evaluation to the top 10 scores gives us an even closer look at the AI's process.

The corresponding similarity scores in Table 6 show that after there are no longer any hints which contain the word fibonacci for card 11, the scores begin to drop off. One of the first hints below this batch is "Golden ratio." Other hints include "Freebonacci", "mathematics", and "Either way could be interesting," which had similarity scores between 0.6 and 0.8. There was a noticeable difference for these hints, but they still maintain some sense of similarity with the original hint.

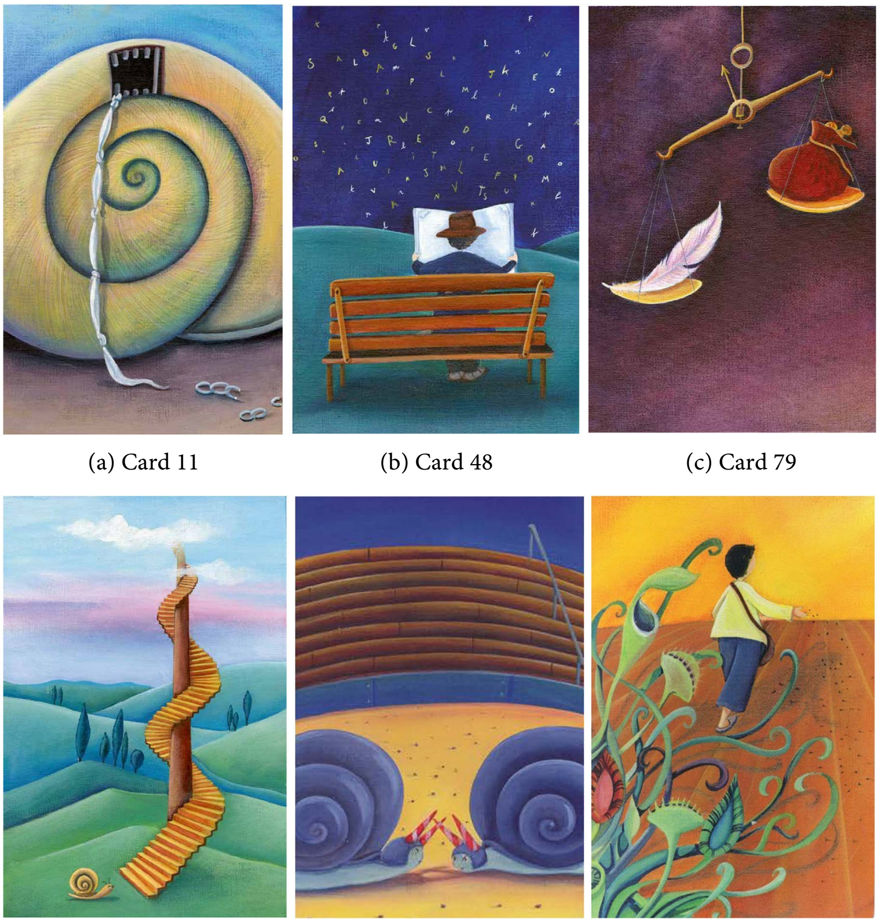
This episode, along with others, confirmed our observation that for each card, there are multiple different elements and emotions evoked, and each one could be described in multiple different ways. The fibonacci example showed that descriptions tend to come in batches, each batch describing a specific element of the card in a certain way.

Table 4. Effect of restriction of hint history data on accuracy. Random selections of the number of restrictions for each of the 84 card dictionaries were made.

Restriction	Overall	4 players	5 players	6 players
500	0.4600	0.4946	0.4612	0.4209
250	0.4236	0.4569	0.4201	0.388
100	0.3775	0.4109	0.3780	0.3402
25	0.2937	0.3229	0.2964	0.2602
10	0.2512	0.2942	0.2450	0.2061
5	0.2309	0.2754	0.2142	0.1887

Table 5. Final accuracy on validatoin and test datasets for our best method along with Vatsakis numbers.

Data set	Overall	4 players	5 players	6 players
Val	0.5020	0.5343	0.5039	0.4654
Test	0.5011	0.4516	0.4893	0.4592
Val, from [3]	0.4793	0.523	0.488	0.427
Test, from [3]	0.4731	0.516	0.472	0.424



(a) Card 11 (b) Card 48 (c) Card 79

(a) Card 12 (b) Card 19 (c) Card 25

Figure 3. 6 cards [2] in a specific episode of the dataset provided by [3].

Table 6. Top 5 similiarity scores calculated by the AI player for each card in this episode. Corresponding hints are displayed in Table 7 and Table 8.

Card	1st	2nd	3rd	4th	5th
11	0.9165	0.9165	0.9120	0.9120	0.9120
48	0.9120	0.6062	0.6004	0.6004	0.5913
12	0.9120	0.8873	0.8148	0.6622	0.6081
25	0.6227	0.6040	0.6016	0.5966	0.5944
19	0.9120	0.9120	0.9120	0.9120	0.6786
79	0.6384	0.6106	0.6077	0.6071	0.6059

Due to the wide scope of the game, many different cards can result in similar descriptions, meaning that taking the max only was not as effective. Top 5 appeared to be the sweet spot from our testing, as adding more tends to take away from the influence of the most similar hints. This also explains why averaging doesn't work well, as each card may have multiple batches of descriptions which are completely different to the given hint, pulling the score of the correct card down. In our example, Fibonacci was a relatively common description, taking up 10 hints for the drop off from the first batch to the second batch to occur. The more unique descriptions would have much sharper drop-offs.

This reasoning also explains why taking the largest minimum performed poorly. Due to the diversity of descriptions, a very small or large minimum did not mean much. The minimum's ineffectiveness is further highlighted by the relative success of the range. High range indicates a high maximum value, while the minimum doesn't have much of an effect and would likely be similar across the cards. Perhaps unsurprisingly, we observe the same trends for the AI-human evaluation.

Limiting hint history showed a drastic drop off, until it began to hold back the naive CLIP method, as the accuracies show in **Table 4**. On average, each card contained 1100 hints, while the minimum number was 767. The accuracy dropped at a steady rate, and it generally took at least 15 hints to improve on naive CLIP. It took a limitation of 500 to drop the accuracy by 0.042.

Overall, however, we were able to improve on the numbers achieved by Vatsakis *et al.*, showing that the recent advancements in computer vision are more effective than traditional machine learning algorithms for image recognition tasks. It is likely that the consideration of both text and image creates a more balanced judgement of each episode. Additionally, analyzing the images means there is always an impartial aspect to the calculation—the image is always the same, unlike the descriptions which can occasionally be difficult to understand for even humans. The amount of training data is another key aspect in the success of the agent, as many obscure hints that are difficult for CLIP to understand are covered due to similar hints in training being recognized by CLIP's zero-shot ability.

Table 7. Top 5 hints for cards 11, 48, and 12.

Card 11	Card 48	Card 12
Fibonacci…	Fibonacci	Fibonacci
Fibonacci…	This doesn't make sense	Fibonacci
Fibonacci	Ignorance	It's a Fibonacci kind of thing
Fibonacci	Ignorance	Fibonacci ascension
Fibonacci	Is that the number 9?	The golden ratio

Table 8. Top 5 hints for cards 25, 19, and 79.

Card 25	Card 19	Card 79
It's not much but it's honest work	Fibonacci	I don't think this is correct
I didn't expect this…	Fibonacci	This doesn't make ANY sense
Not my problem anymore	Fibonacci	It's supposed to do that
Every 6th year, this doesn't happen	Fibonacci	Not correct
I'm not sure how long this will last	Golden ratio	This just doesn't make any sense

Limitations

We improve the accuracy of our AI agent to surpass that of Vatsakis and humans. However, the data is from a casual website, requiring only registration to play. Many rounds are played by casuals and first-timers, meaning that the human accuracy score may not be the best benchmark. Although Dixit is a complex game which requires a specific skill set that test understudied elements of current AI, it is not well known. The quality of the data and our benchmarks/baselines may be questioned. However, our numbers are still impressive for a task that is not easy for both humans and AI.

There are more improvements which can be made for AI Dixit players. The most obvious one is that of fine-tuning CLIP for Dixit. CLIP's zero-shot abilities are great, but it can still improve through optimization of its parameters. This is especially true for image-text pairs in Dixit. Additionally, we did not address the topic of generating descriptions, a task that is much more difficult. It would likely require extensive training and optimizations in order to create a model which works well, along with more time in order to test it.

5. Conclusions

In this paper, we developed a Dixit AI agent, utilizing the capabilities of CLIP in order to obtain the best accuracy possible for choosing the correct card or matching human behavior. We obtain a 0.5003 accuracy rate on the test data, surpassing that of humans (0.4782) and the Vatsakis model, (0.4793). With extensive training and fine-tuning, this number can likely be improved.

Creating an AI agent which can guess the correct card correctly at a greater rate would be an impressive step of advancement for computer vision, widening the scope of its ability to identify abstract and creative image-text pairings. Another task would consist of being able to generate the Dixit descriptions effectively, a challenging, but interesting task. CLIP could be used as a function to calculate how effective certain words and certain strings of words are and trained to prefer the ideal types of descriptions in Dixit. This is a task that would be time-intensive and challenging, but still interesting.

Acknowledgements

We thank Greg Shakhnarovich for guidance and many helpful discussions throughout the project.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Singh, A.K., Ding, D., Saxe, A., *et al.* (2022). Know Your Audience: Specializing Grounded Language Models with the Game of Dixit. arXiv: 2206.08349. https://doi.org/10.48550/arXiv.2206.08349

[2] Roubira, J.-L. and Carpuat, M. (2008) Dixit. [Board game].

[3] Vatsakis, D., Mavromoustakos-Blom, P. and Spronck, P. (2022). An Internet-Assisted Dixit-Playing AI. *Proceedings of the* 17*th International Conference on the Foundations of Digital Games*, Athens, 5-8 September 2022, 1-7. https://doi.org/10.1145/3555858.3555863

[4] Radford, A., Kim, J.W., Hallacy, C., *et al.* (2021) Learning Transferable Visual Models from Natural Language Supervision. arXiv: 2103.00020. https://doi.org/10.48550/arXiv.2103.00020

[5] Cherti, M., Beaumont, R., Wightman, R., *et al.* (2022) Reproducible Scaling Laws for Contrastive Language-Image Learning. arXiv: 2212.07143. https://doi.org/10.48550/arXiv.2212.07143

[6] Han, J. and Kamber, M. (2008). Introduction to Data Warehousing and Mining. https://hanj.cs.illinois.edu/cs412/bk3/KL-divergence.pdf