CS520: Question 2

## Question 1

*🡪Describe what goes on in this scene in terms of algorithms discussed in the class. What is realistic, what is not? What is feasible, based on the kind of algorithms you know to underly AI, and what seems a stretch? Be as thorough and clear as you can be, drawing on as much of the course as you can.*

The provided scene of the 1983 film ‘War Games’ portrays a variety of concepts that describe what an AI can be and do. The scene shows an AI playing tic-tac-toe against itself, which causes a loop of repeated draws, resulting in it using a massive amount of energy. It breaks out of the loop to attempt the same but with a far more complex “game” - war simulation - between USA and USSR. The simulated games result in a loop until the AI shuts down with the result that the ideal outcome is simply not to play.

The AI is coded with the logic to play tic-tac-toe against other players, and by pitting the AI against itself, Matt Broderick makes the AI learn from repeated games of tic-tac-toe. The AI is in a loop of playing against itself with a given heuristic on how to respond to an action done by itself. Tic-tac-toe is fully deterministic and a zero-sum game. It has a maximum of 362,880 game configurations (Tic-tac-toe, n.d.), so the AI playing against itself has a breaking point: once it runs through the possible configurations. There are different algorithms the AI may have been coded with to solve tic-tac-toe: minimax algorithm, alpha-beta pruning, brute force, etc. Since tic-tac-toe is biased towards whoever goes first, the AI playing against itself would imply that it will attempt to draw because the algorithms that train the AI reward for winning, and penalize for losing, with drawing being the middle ground.

The AI then decides to move on to simulating war between USA and USSR. This is a different setting than simple tic-tac-toe: the environment is stochastic, and there may be more than 2 players (historically speaking, in war there have been ally countries). The AI simulates a game of thermonuclear war between USA and USSR, the AI playing against itself again. The way the AI simulates the game can be something simple, like introducing chance into a min-max game. A more fitting method would be a Markov Decision Process or Monte Carlo simulation: run a simulation and get feedback before running the next simulation to improve on the previous simulation’s result. Going through the game repeatedly, the AI notes that there is no winner since both sides keep losing.

War Games presents a very interesting representation of artificial intelligence, yet it is a fictional one. The movie was made in the 80s, and as of today, no true general AI (an AI that can perform multiple tasks like a human) exists, while the AI in the movie appears to be one, having the capability to solve tic-tac-toe and simulate war. Moreover, there is nothing to learn from tic-tac-toe: the state space is finite and pre-determined. On the other end, the state space for the war simulator is infinite and the environment is stochastic. An AI that used up a massive amount of energy to simulate the finite states of tic-tac-toe cannot possibly simulate the infinite discrete states of war and end with a result. After giving up on simulating, the AI remarks, “A STRANGE GAME. THE ONLY WINNING MOVE IS NOT TO PLAY”. The AI is portraying anthropomorphic behavior here, implying that it is perhaps sentient and capable of communicating, which is not possible yet.

## Question 2

*🡪 Why are Large Language Models prone to making things up in this way? Why don’t they know better? Be thorough and draw on our discussions of machine learning. Be wary of anthropomorphizing.*

Languages are an essential for communication and computers are yet to obtain the capacity to coherently communicate. Large Language Models are a revolutionary application of natural language processing in recent times. These models supply systems with the ability to communicate. At least that is what it appears to be. LMs are, in essence, transformer models that are trained over massive amounts of text data. They can generate text, abridging text, and even generating images from input prompts.

The above features of LMs seem flawless at a high level, but a flaw that sticks out is that the model simply gives an output that is biased on what is provided in verbatim as input. There is no semantic meaning to the model, it just generates an output that has some meaning to the input but may not necessarily be the correct output. The model hands out an output with no moral; it replies with a confident output that is generated from what it is trained from, not with what is a necessarily an ethically correct answer. To summarize, such models are statistical learners, so their method of producing outputs is by tracking statistical associations that it can learn from the training sets (McCoy, Pavlick, & Linzen, 2019). Therefore, LMs simulate communication rather than actual communication that it appears to do.

*🡪Describe how you might build and train a system to take a natural language prompt from a user and generate a good google search prompt from it to achieve good, targeted results from google. Be clear about methods, data, representation - all the usual things we discussed in class.*

Training a system to generate goof results from Google through a language prompt involved training a model to understand relations in texts. To begin with, a suitable model would be a recurrent neural network (RNN) since the input length is variable which RNNs are made to handle. Transformers are found to be faster and more convenient compared to RNNs. Instead of sequentially processing input word-wise like an RNN, a transformer processes the entire input at once and encodes it using embeddings like GloVe or word2vec, though embeddings tend to ignore different semantic meanings for a word/phrase and take only one of them.

By taking the entire input at once from left to right to encode, some context is provided: what is to the left or right of a word in the input. Transformers also make use of multi-head self-attention, which evaluates contextuality for each word. The layer architecture along with activation and loss functions needs to be decided. Various models such as Google Bert and GPT have shown great results through their architecture. With the model being able to take in a prompt as input, it can contextualize the input sentence and statistically relate the sentence to one that Google stores to provide the relevant search results.

# Works Cited

McCoy, R. T., Pavlick, E., & Linzen, T. (2019). Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. *arXiv*, 21.

*Tic-tac-toe*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Tic-tac-toe