CS 520 : Introduction to Artificial Intelligence

Final Exam: Essays

RUTGERS
UNIVERSITY

Naga Vamsi Krishna Bulusu (nb847)

War Games

1. **Describe what goes on in this scene in terms of algorithms discussed in the class**

   In the climax scene of the 1983 AI Thriller ‘War Games’, Matthew Broderick tries to stop the AI from playing a game of global thermonuclear using the same approach to train tic-tac-toe Model.

   While playing the tic-tac-toe game, he inputs the number of players as zero, which essentially says that the machine is playing with itself to find a winning strategy. It is essentially learning about all the utility of possible state spaces.

   As the state space of tic-tac-toe is finite (3^9), we can easily compute the utility of a state using knowledge algorithms discussed in class. We can co-relate the process of finding the best utility of a state space as a **Min-Max Problem** or as a **Markov Decision Process**.

In the Min-Max problem we back propagate from the leaf node (Terminal State) to the root node (Starting State) computing the utility of each action.

Whereas in the Markov decision process we can pre-compute all the utilities of state space using the value iteration algorithm until convergence.

2. **What is realistic, what is not? What is feasible, based on the kind of algorithms you know to underly AI? What seems like a stretch?**

   It is realistic to apply knowledge algorithms on a finite state space and in a uniform environment. In the case of tic-tac-toe we have a limited number of states, finding the utility of state is fairly simple. Whereas in case of global thermonuclear war the state space is unknown and infinite, it is not feasible to apply the same algorithms.

   Using the same approach to train the model of global thermonuclear war seems like a stretch. As the state space is infinite and the environment is stochastic.

Chat GPT

1. **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.**

   - Large Language Models are trained on large corpus using advanced algorithms. LLM generalizes answers based on the knowledge base. If the query is completely out of context and unknown (0% confidence) to the knowledge base. The LLM uses fall back policies to unknown queries.
   - When the confidence of the knowledge base is below some threshold, the LLM picks up keywords from the utterance and generalizes data from sources which could be untrue.
   - For example, if we query about directions to a place without much context about the location, the LLM will infer about our intent properly but generalizes direction to the place with highest confidence which is irrelevant to us.

2. **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.**

Data Collection

- Collect a large dataset of prompts and corresponding Google search prompts that are known to produce good, targeted results. This dataset could be created by manually annotating prompts or by using existing datasets of search queries and clicks. Data can be represented in a vector of vectors.

Preprocess Data

- Preprocess the data by tokenizing, lemmatizing the prompts and search queries and synonym replacement or stemming to expand the coverage of the model.

Splitting Into Training, Validation and Test sets

- We split the data into training, validation, and test sets with uniform distribution.

Training neural network

- Train a LSTM, to generate a Google search prompt given a natural language prompt as input. The model should be trained to minimize the loss between the generated search prompt and the ground-truth search prompt in the training data.

- Fine-tune the model by adjusting the hyperparameters, such as the learning rate and the size of the hidden layers, and by using techniques such as dropout and early stopping to prevent overfitting.

Testing the model

- Evaluate the model on the validation and test sets to measure its performance. Deploy the model in a system that allows users to input natural language prompts and receive generated Google search prompts in return.