AI BASED VIRTUAL TRAVEL AGENT SYSTEM

Conference Paper · January 2022

CITATIONS
0

READS
157

3 authors, including:

Pallavi Tawde
Nagindas Khandwala College
12 PUBLICATIONS **5** CITATIONS

SEE PROFILE

 ISBN:978-93-5607-568-9

AI BASED VIRTUAL TRAVEL AGENT SYSTEM

Mitsinh Rajendrasinh Rajput
Student
Vidyalankar school
of information technology
mitsinh.rajput@vsit.edu.in

Wasim Ayub Khan
Student
Vidyalankar school
of information technology
wasim.khan@vsit.edu.in

Pallavi Devendra Tawde
Guide,
Vidyalankar school
of information technology
pallavi.tawde@vsit.edu.in

ABSTRACT:

Nowadays, people want a vacation to explore more about world and different culture which they are not aware of. People tends to explore more wonder touring places without any help of any human intervention. They want a virtual assistant which can assist about the tours. AI-powered chatbots are motivated by the need of traditional application to provide a chat facility where a bot is required to be able to chat with users and solve queries. Where live agent can handle only two or three operations at a time, chatbots can operate with an upper limit which really scales up the operations. Having a chatbot clearly improves the response rate compared to human support team. Through the study of various research paper and online course we have built an application called “AI Based Virtual Travel Agent System”. This application will interact with users through chatbot and provides output. Application also uses recommendation to provide accurate prediction according to user’s preference.

INTRODUCTION:

Artificial Intelligence in machines is a very challenging discussion. It involves the creation of machines which can simulate intelligence. This paper discusses some of the current trends and practices in AI and subsequently offers alternative theory for improvement in some of today’s prominent and widely accepted postulates. The paper shows how current approach towards AI is not adequate and offers a new theory that discuss machines intelligence, throwing light to the future of intelligent systems The research on recommender system gained importance after the emergence of collaborative filtering algorithm. It uses information from many sources to make prediction and to suggest an asset for the user. Filtering mechanisms play an important role in the recommendation process. The most used filtering techniques are collaborative filtering, content-based filtering, knowledge-based filtering.

The main aim of using personalization techniques is to generate customized recommendation according to the user preference and interest. The recommendation system as an objective is to filter unwanted information and to provide a specific result for the user.

COLLABORATIVE FILTERING:

K – Nearest neighbour One of the most common forms of collaborative filtering is the k-nearest neighbour approach. There are two main variants of nearest neighbour for collaborating filtering, user-based and item-based.

Both methods generally assume that no user or item features are provided, so here x and y are simply respective user and item indices. When the number of users is far fewer than the number of items, it has been found that the user-based approaches usually provide better prediction as well as being more efficient in computation.

This hold for the evaluation in this paper so we focus on user based approach. Given a user x and an item y. Let N(x:y) be the set of N user nearest neighbour of x that have also given a rating for y, where "nearest" is determined by the cosine similarity Simx,z between two vectors of rating for user x and y.

The predicted rating $\bar{R}$ x,y ∈ [0,1] that the user x gives item y can then be calculated as:

$$\hat{R}_{\mathbf{x},\mathbf{y}} = \frac{\sum_{\mathbf{z} \in N(\mathbf{x}:\mathbf{y})} Sim_{\mathbf{x},\mathbf{z}} R_{\mathbf{z},\mathbf{y}}}{\sum_{\mathbf{z} \in N(\mathbf{x}:\mathbf{y})} Sim_{\mathbf{x},\mathbf{z}}}$$

Figure 1: Collaborative Filtering

SEQUENCE PREDICTION

Given a sequence of items that the user interacted with in his previous search history. What will he/she interact in his next visit? The idea behind this type of problem is straightforward, learning the evolution of user affinity's change in the timeline from the tour selection sequence. The good news is that it's very similar to NLP tasks, and we can apply NLP techniques like embedding, attentions into 19 the models. Basically, the embedding can represent a user or item information in the model, and the attention can decide what in the previous sequence should play a more important role in the future sequence.

One naive approach is to use time-decay on the product embedding to differentiate the importance of more recent events and more previous events. However, the RNN (LSTM /GRU) based models could do better.

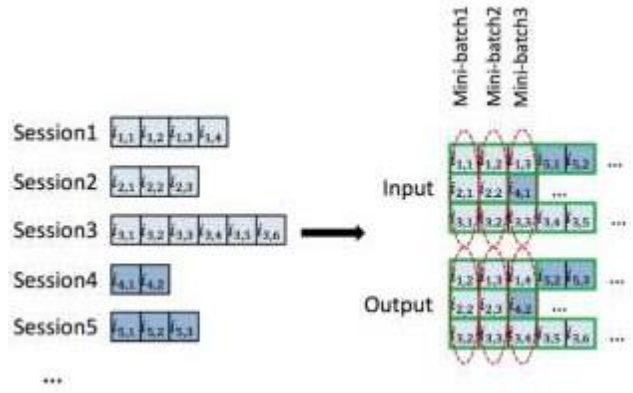

Figure 2: Problem Definition

TENSOR FLOW:

Tensor Flow is a machine learning system that operates at large scale and in heterogeneous environments. Tensor-Flow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices, including multicore CPUs, general-purpose GPUs, and custom designed ASICs known as Tensor Processing Units (TPUs). Tensor Flow enables developers to experiment with novel optimizations and training algorithms. Tensor Flow in production, we have released its an open-source project, and it has become widely used for machine learning research.

Tensor Flow execution model

Tensor Flow uses a single dataflow graph to represent all computation and state in a machine learning algorithm, including the individual mathematical operations, the parameters and their update rules, and the input pre-processing.

Dataflow Tensor Flow differs from batch dataflow systems in two respects:

The model supports multiple concurrent executions on overlapping subgraphs of the overall graph. Individual vertices may have mutable state that can be shared between different executions of the graph

Operations:

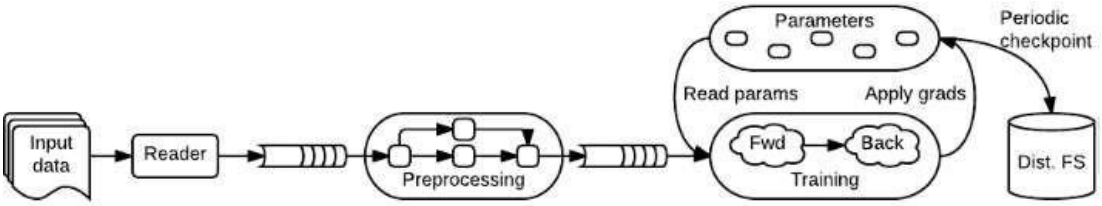

Figure 3: A schematic Tensor Flow dataflow graph for a training pipeline contains subgraphs for reading input data, reprocessing, training, and check pointing state

An operation takes m≥0 tensors as input and produces n≥0 tensors as output. An operation has a named "type" (such as Const, MatMul, or Assign) and may have zero or more compile-time attributes that determine its behaviour. An operation can be generic and variadic at compile-time: its attributes determine both the expected types and arity of its inputs and outputs.

For example, the simplest operation, Const has no inputs and a single output. Const has an attribute T that determines the type of its output, and an attribute Value that determines the value that it produces. AddN is variadic: it has a type of attribute T, and an integer attribute N that defines how many inputs (of type T) it accepts.

SYNCHRONOUS REPLICA COORDINATION:

Though we designed Tensor Flow for asynchronous training, we have begun experimenting with synchronous methods. The Tensor Flow graph enables users to change how parameters are read and written when training a model, and we implement three alternatives.

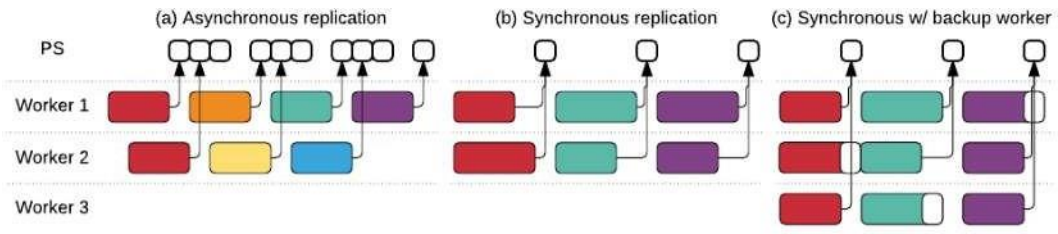

Figure 4: Three parameter synchronization schemes for a single parameter in data-parallel training: (a) asynchronous, (b) synchronous, and (c) synchronous with backup workers.

In the **asynchronous** case, each worker reads the current value when the step begins and applies its gradient to the different current value at the end: this ensures high utilization, but the individual steps use stale information, making each step less effective.

The **synchronous** cases use queues to coordinate execution: a blocking queue acts as a barrier to ensure that all workers read the same parameter version, and a second queue accumulates multiple gradient updates to apply them atomically. The simple synchronous version accumulates updates from all workers before applying them, but slow workers limit overall throughput.

In the **Synchronous with backup workers**, we implement backup Map Reduce backup tasks. Whereas Map Reduce starts backup tasks reactively, after detecting a straggler our backup workers run proactively, and the aggregation takes the first m of n updates produced.

MACHINE LEARNING ALGORITHMS

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values Recommendation engine are a common use case for machine learning.

1) Supervised learning: In this type of machine learning, data scientists supply algorithms with labelled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

2) Unsupervised learning: This type of machine learning involves algorithms that train on unlabelled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

3) Semi-supervised learning: This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labelled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

4) Reinforcement learning: Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

Seq2seq MODEL:

This strategy selects the most probable word (**i.e. argmax**) from the model's vocabulary at each decoding time-step as the candidate to output sequence.

The problem with this approach is that once the output is chosen at any time-step t, we don't get the flexibility to go back and change our choice. It is seen in practice that greedy decoding strategy is prone to have grammatical errors in the generated text. It will result in choosing the best at any time-step t but that might not necessarily give the best when considering the full sentence to be grammatically correct and sensible.

RESULT:

Based on basic-brinn-seq2seq-greedy model and the data consist of cornell movies-dilaogs corpus.

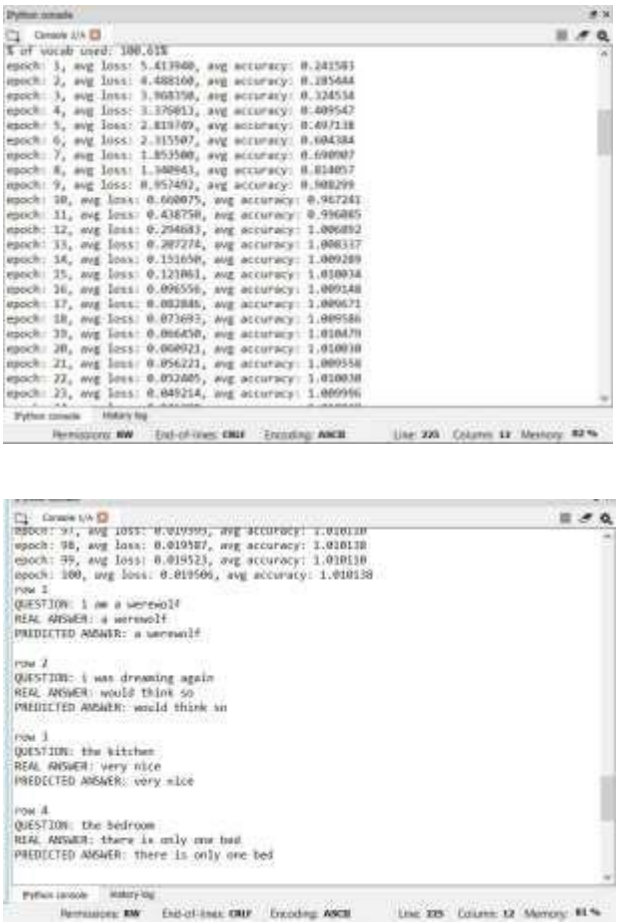

Figure 5: Real and Predicted Output

We are using the Cornell Movie-dialogs corpus as our dataset, which contains more than 220k conversational exchanges between more than 10k pairs of movie characters. “+++$+++” is being used as a field separator in all the files within the corpus dataset.

movie_conversations.txt has the following format: ID of the first character, ID of the second character, ID of the movie that this conversation occurred, and a list of line IDs. The character and movie information can be found in movie_characters_metadata.txt and movie_titles_metadata.txt respectively.

Encoder

The Encoder consists of:

Input Embedding

Positional Encoding

N of encoder layers

The input is put through an embedding which is summed with the positional encoding. The output of this summation is the input to the encoder layers. The output of the encoder is the input to the decoder.

Decoder

The Decoder consists of:

Output Embedding

Positional Encoding

N decoder layers

The target is put through an embedding which is summed with the positional encoding. The output of this summation is the input to the decoder layers. The output of the decoder is the input to the final linear layer.

Seq2seq

Seq2seq consists of the encoder, decoder and a final linear layer. The output of the decoder is the input to the linear layer and its output is returned.

CONCLUSION:

In Summary, “AI Based Virtual Travel Agent System” is an application-based project built on Flutter. The project uses Firebase Authentication Database for data security and integrity. Project contain two main modules named Chatbot and Recommendation System respectively. Chatbot is integrated with IBM Watson Assistant for solving the user queries related to tour and services. The aim of using personalization techniques is to generate customized recommendation according to the user preference and interest. The main goal of this project is to analyse users’ action for accurate result and purpose is to achieve chatbot experiences live up to expectations.

REFERENCES:

1. S. V. Logesh Ravi, "A Collaborative Location Based Travel Recommendation System through Enhanced Rating Prediction for the Group of Users," p. 28, 2016.
2. B. A. H. K. Dandison Ukpabi, "Robots, Artificial Intelligence, and Service Automation in Travel, Tourism and Hospitality," 2019. [Online]. Available: https://www.researchgate.net/publication/336531265_Chatbot_Adoption_in_Tourism_Services_A_Conceptual_Exploration.
3. K. V. Anirudh Khanna, "A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence," 2015.
4. S. S. Joseph Noel, "New Objective Functions for Social Collaborative Filtering," [Online]. Available: http://users.cecs.anu.edu.au/~ssanner/Papers/www12.pdf.
5. Martín Abadi, "TensorFlow: A system for large-scale machine learning," 2016.