Trojan Bot

The Smart and Social Conversational AI Bot

Yash Phogat, Piyush Chaudhari, Deepali Bhola, Varinthorn Bulakul, Dipesh Chauhan, Nikhil Bagmar, Sathyanaraya Raghavachary

Department of Computer Science, Viterbi School of Engineering
University of Southern California, Los Angeles, CA 900089

Abstract

Trojan Bot is a smart and social conversational AI bot that is intended to engage in social conversations with random users of Amazon Alexa across the world. Our bot will work on a user profiling-based model. Based on the training data we obtain; we intend to classify dialogues/ conversations among few pre defined set of personality types. Going forward, whenever the bot starts a conversation with any random user, it starts populating a personality matrix for the user so that it can pre-emptively decide the flow of the conversation and bring an engagingness quotient to the conversation along with faster dialogue processing. Apart from this we'll use a BERT [Devlin et al., 2018] based dialog manager for Natural Language inference and next sentence prediction, a graph based topical association manager and a sentiment-based conversation quality indicator. Traversing through classical knowledge graphs for sentence intent detection and use of social media channels for opinion-based dialogue formation will be included.

1.Introduction

The high-level flow and the details of the major components/ managers used by the team are discussed here forth. At this point, we intend to use the Amazon Co-Bot kit along with our custom managers for an engaging socialbot.

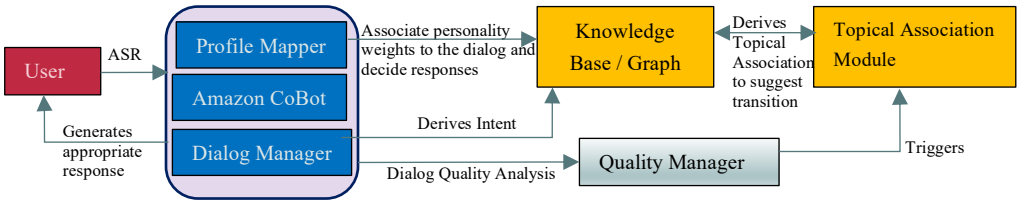
The unique feature of our bot will be the personality profile manager. During the training phase of the challenge we intend to tag conversations from the training set into the four dichotomies of the Myers–Briggs Personality Indicator model [1]. Based on these tags we intend to create a neural network model that learns to add weights for every dichotomy in an incoming dialogue instance. This will help us create a profile matrix for the conversation topics. Using the profile matrix, we'll try to drive the conversation and establish an expected flow of conversation. This matrix will ideally improve response time and quality, through every dialogue.

Further, the dialog manager of our bot that is intended for Natural Language Understanding and Next Sentence prediction will be modelled using BERT [2]. The details of the manager are mentioned in Section 2.3. Apart from the usual conversation intent detection, the dialog manager will summarize the predicted sentence and present them in a more humanly way (since usual Wiki data is complicated for all). For every dialogue in the conversation we assign a quality-heuristic value that will be based on emotion/sentiment classification. We establish a cushioned-threshold value of the conversation quality to proceed through the conversation. For a good quality conversation, we keep going deeper into the current conversation topic and for a quality value below the threshold limit, we trigger a Graph Clustering Association Rule Mining Algorithm (GCAR). This model will suggest making an associative topical transition. The topical association model will provide candidate topics for transition and rank the candidates based on the graph edge weights and our profile matrix.

2.System Architecture

We intend to use the Amazon Conversational Bot Toolkit (cobot) along with our custom managers to create our social bot. Through the initial research phase, we picked up few

modules we intend to improve for a more fun, engaging, simplified and knowledge-based conversation. In the following section we briefly discuss each module:



2.1. Profile Mapper Module

The Profile Mapper module will be a neural network trained to learn weights for the Myers – Briggs Personality Indicator dichotomies. Based on the pre-learned weights we create a profile matrix during the conversation that classifies the dialogues among the dichotomies and help us rank, related topics. This will also help us derive related knowledge and present it to the user in a more personal way.

We can use the profile matrix and build a subset of responses aligning to the user profile. This will fasten the response rate due to smaller set of responses to traverse. This will in fact bring scalability to the social bot that will be able to handle even larger sets of data and knowledge graphs.

2.2. Knowledge Base and Knowledge Graphs

Our social bot would acquire content and make topical associations from sources like:

- Google's Knowledge Graph API: for known topics and summarization.
- Wikidata API for related topics.
- Evi for fact related question and answers, jokes etc.
- Amazon provided trending list.
- Washington Post for news articles: Articles will be scraped and indexed in Elastic search.
-Reddit/Twitter for user-generated content and expressing opinions.

These sources cover a variety of interests and can keep the conversation going.

2.3. Dialog Manager/ Summarization

We propose a self-attention-based approach for dialogue summarization and next sentence prediction which will involve a sequence to sequence based encoder decoder network involving the usage of Google's Bidirectional Encoder Representations from Transformers (BERT) [2].

BERT can be used as an encoder which will generate encoded representations for every word in the dialogue.

Previous research suggests use of two stage decoders to get refined summary. In the first stage, the summary can be generated by using a left-context only decoder. The second stage will involve masking each word of the summary and predicting the refined word using a refine decoder.

Thus, the encoder BERT will give us a word embedding for each word in the dialogue which can be later fed to the decoder BERT to generate a summary and derive the intent of the user dialogue. Since, the decoder in BERT is a left context only decoder, it will not be able to consider the words ahead of the word under consideration. Thus, a second decoder will take this word embedding and would refine it. Thus, the words will pass through a transformer model in BERT which are stacks of Multi-Head Self Attention, Multi-Head Personal Attention etc. along with other layers. The attention model will help us prioritize certain words during the decoding process.

We will be using BERT for abstractive summarization because this model will be embedded in the Alexa bot while giving a response. In a certain scenario, if Alexa needs to crawl through many web pages and needs to summarize the content while crawling webpages, then this model will be useful.

2.4. Topical Association Classifier

In every conversation with the user, our system will find the Intent or 'topic' of the conversation using our Dialog Manager (BERT). We will then utilize this topic to find associations in user's knowledge graphs. For this purpose, we will use a special algorithm – 'Graph Clustering Association Rules Mining Algorithm (GCAR)'. It is an optimization of the Apriori algorithm which is a classic algorithm in association rule mining.

In the GCAR graph G, the topics from each conversation are added as a subgraph G', where each node is a topic and the adjacent nodes are related topics. Each time a topical transition is triggered, the corresponding topic set is invoked and a counter of 1 is added to the edge between the current topic and the next topic. With each conversation, counters keep on increasing and the topical transitions is made to the node with greater counter value and profile state; matching the conversation profile at the time when topical transition is triggered.

Iteratively, the frequent topics set that have been visited may be eliminated from graph G, to reduce redundancy. Thus, a large knowledge graph G with topics as nodes and the counters on edges indicate the best topical transitions in G.

The two main advantages of using GCAR are:

1. The reduction of database scans
2. The elimination of frequent topic set

2.5. Conversation Quality Indicator (Red Flag Manager)

We need to detect red-flags at every step in a conversation to decide whether the user is interested in it or not. This can be done by detecting user emotions and sentiments [3] through text using an LSTM [4] based approach. Emotion detection will classify user conversation as 'angry', 'sad', 'happy' or 'other'. Whereas sentiment analysis would classify the user's view as 'positive' or 'negative'. A conversation detected as angry or sad or negative could be considered as red flags and the social bot would want to drive the conversation away from such topics to make it more engaging to the end user.

2.5.1. For Emotion Detection: -

The input user utterance is fed into two LSTM [4] layers using two different word embedding matrices. One layer uses a semantic word embedding, whereas the other layer uses a sentiment word embedding. These two layers learn semantic and sentiment feature representation and encode sequential patterns in the user utterance. These two feature representations are then concatenated and passed to a fully connected network with one hidden layer which models interactions between these features and outputs probabilities per emotion class.

2.5.2. For Sentiment Analysis: -

The input user utterances are passed to an embedding layer which generate word embeddings. These embeddings are passed to an LSTM layer. The output of the LSTM layers may then be fed to a sigmoid activation function that turns all output values to a 'positive' or 'negative' sentiment.

REFERNCES

[1] https://en.m.wikipedia.org/wiki/Myers%E2%80%93Briggs_Type_Indicator
[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*
[3] Umang Gupta, Ankush Chatterjee, Radhakrishnan Srikanth, Puneet Agrawal, *A Sentiment-and-Semantics-Based Approach for Emotion Detection in Textual Conversations*
[4] Sepp Hochreiter, Jurgen Schmidhuber, *LONG SHORT-TERM MEMORY*