Intelligent Chatbots: Implementation Tools and Techniques.

Introduction

Chatbots are intelligent conversational computer systems designed to mimic human conversation to enable automated online guidance and support. The increased benefits of chatbots led to their wide adoption by many industries in order to provide virtual assistance to customers. Chatbots utilise methods and algorithms from two Artificial Intelligence domains: Natural Language Processing and Machine Learning [1]. In this review, we discuss in brief about some of the various techniques used to implement an intelligent chatbot that goes far beyond just a FAQ answering bot and later we introduce you to spaCy[1] a free, open-source library for advanced Natural Language Processing (NLP) in Python.

Background

Although the quest for being able to create something that can understand and communicate with its creator has deep roots in human history, Alan Turing is thought to be the first person to have conceptualised the idea of a chatbot in 1950, when he proposed the question: "Can machines think?". Turing's description of the behaviour of an intelligent machine evokes the commonly understood concept of a chatbot. [1].

Chatbots are generalized under Conversational AI which can be seen as the process of automating communication and creating a personalized customer experiences at scale.

- It is the process of receiving input data in the form of unstructured conversational data.
- Then in turn structuring this data, to derive meaning, intent and entities from it.

There are multiple ways to design a chatbot. It may be designed as a simple rule-based template system or may involve complex artificial neural network (ANN) architectures. [2]
Neural Network Language Models (NNLMs) such as the Re- current Neural Network (RNN) and Long Short Term Memory (LSTMs) are also popular techniques used in designing a chatbot [3]. Since Deep learning and neural networks are gaining prominence in the field of NLP, with hidden states between input and output and robust networking to produce the best performance, lets look at each method in brief.

A. Recurrent Neural Network

RNN is designed to take text sequences as inputs or return text information sequences as outputs, or both. They are referred to as recurrent because the hidden layers of the network have a loop in which the output and the cell status of each step turn out to be entered in the

[1] https://spacy.io/

next step. RNN can recall just that since it's inside the memory. It generates data, copies the data, and loops returned to the network. The primary strength of the RNN is the ability to memorize the consequences of previous computations and to use certain records in the current computation. Unlike conventional translation models, where only a finite window of previous words can be considered to condition the language model, RNN is effective in conditioning the model to all the preceding words in the corpus. We should consider the sentence as a mini-batch and the sentence as a mini-batch. k words would have k word vectors to be stored in memory. Fig. 1 [3]

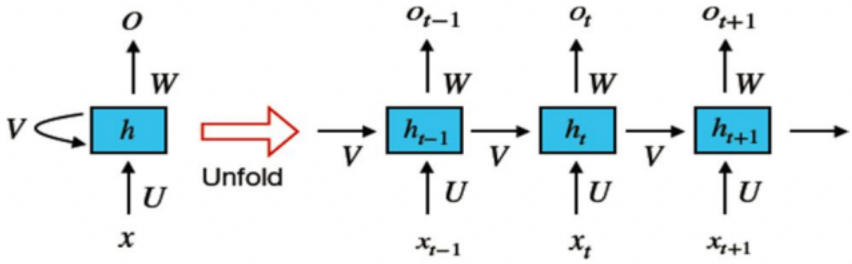
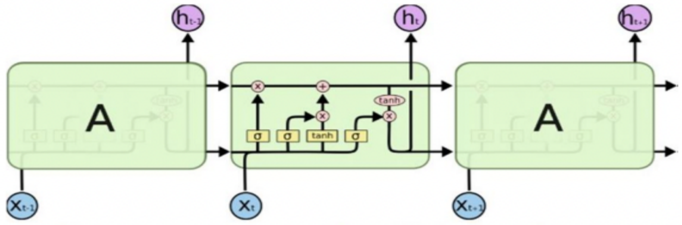


Fig. 1. RNN architecture for sequence to sequence

B. Long Short Term Memory (LSTM)

Sequence-to-sequence (SEQ2SEQ) model. In deep learning (DL), there are 2 key assignments. The first is to derive from the input the value. The second is to produce output from that, in the case of a chatbot application, either a translation or an answer. The Principal The difficulty of creating a decent model is that it produces an appropriate sense of meaning and output inputs that are essentially connected. In deep recurrent neural networks (DRNN) with an attention mechanism, the sequence-to-sequence (seq2seq) model. The capacity of the deep neural network to communicate in human spoken language, although a number of the restrictions of applied mathematics models and implementation mechanisms are side-tracked at the same time. Long Short Term Memory networks are a special form of RNN, capable of learning long-term dependencies, usually only called "LSTM".



The repeating module in an LSTM contains four interacting layers.

Fig. 2. LSTM input, forget, and output gates

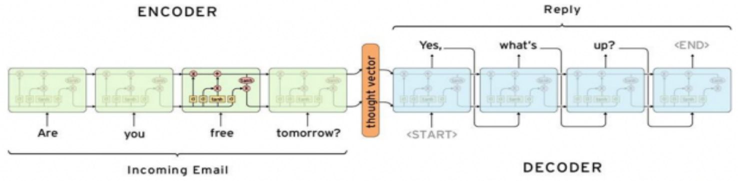


Fig. 3. Encoder Decoder(LSTM) network architecture .

Fig. 2. LSTM input, forget, and output gates [3]
Fig. 3. Encoder Decoder(LSTM) network architecture [3]

The input sequence shown in Fig.3 is ’”Are you free tomorrow? . Thus, when such an input sequence is passed through the encoder decoder network of LSTM blocks (a type of RNN architecture), the decoder produces words one by one at each stage of the iteration of the decoder. The generated output sequence is generated after an entire iteration. “Yes what’s up?”. Various LSTM-based models have been proposed for sequence mapping (via encoder-decoder frameworks) that are suitable for machine translation, text summarization, modelling human conversations, question answering, and image-based language generation, among other tasks. LSTM networks are an extension for recurrent neural networks that broaden their memory.

C. Natural Language Processing(NLP)

Natural Language Processing (NLP) is a branch of computer science and Artificial Intelligence (AI) concerned with human language in the information age. With the rise of voice interfaces and chatbots, NLP has emerged as an essential component of AI as one of the most significant innovations of the information age. NLP, like speech and text, is commonly referred to as natural language processing. NLP applies computers to the words we use to understand human language. NLP is concerned with developing process algorithms in order to mechanically analyse and represent human language.

The process of training an AI chatbot is similar to a human trying to learn an entirely new language from scratch. The different meanings tagged with intonation, context, voice modulation, etc are difficult for a machine or algorithm to process and then respond to. NLP technologies are constantly evolving to create the best tech to help machines understand these differences and nuances better [4]. A popular python framework that helps implement NLP for our chatbot is spaCy, let’s take a look at spaCy.

spaCy

spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. It is a NLP tool and not a chatbot development framework.

Natural Language Processing and Understanding can be light weight and easy to implement. It is within anyone’s grasp to create some Python code to process natural language input, and expose it as an API.

spaCy does not cater for dialog scripts, NLG, dialog state management etc. But, what makes spaCy all the more interesting is that it can be implemented as a language processing API assisting an existing chatbot implementation. Especially in instances where users submit a longer input, the chatbot will do good to work with only a specific span or tokens from the utterance. [5] Also, it can be used for offline post-processing of user conversations.

Conclusion

In this review, we introduced Intelligent chatbots and discussed a variety of modern techniques for designing a chatbot. Finally, we discussed spaCy, a Python library that can aid in the implementation of an NLP-based chatbot. There are numerous gaps in this field, and improved, scalable, and flexible models for industry-specific applications, more human-like model architectures, and improved evaluation frameworks would undoubtedly represent significant advances in the field, i.e. conversational AI.

References

[1] "Caldarini, Guendalina et al. “A Literature Survey of Recent Advances in Chatbots.” Inf. 13 (2022): 41.".

[2] "Adewumi, Tosin P. et al. “State-of-the-art in Open-domain Conversational AI: A Survey.” Inf. 13 (2022): 298.".

[3] "Maher, Santosh et al. “Chatbots & Its Techniques using AI: A Review.” International Journal for Research in Applied Science and Engineering Technology (2020): n. pag.," [Online].

[4] "https://www.analyticsvidhya.com/blog/2021/10/complete-guide-to-build-your-ai-chatbot-with-nlp-in-python/," [Online].

[5] [Online]. Available: https://cobusgreyling.medium.com/using-spacy-in-your-chatbot-for-natural-language-processing-8fa14c26bb51.