American Express Challenge 2020

Soyeong Bak, Dabeen Oh, Emmanuel Ren

Artificially intelligent
ChatBot
for travel recommendation

Table of Contents

Business insights of American Express: Values added by an AI chatbot

Key Partners

- Banks
- Airlines (ex. Delta Sky Miles)
- Merchants
- Hotels (ex. Hilton)
- Investors
- Travel Guides
- Different tours in Travel Guides can get more exposure

Key Activities

- Marketing
- Customer Support
- Partnerships
- IT developments

Key Resources

- 59000 employees
- Bargain Power
- Contracts with Business
- Acquisitions
- Travel resources

Value Propositions

- Travel advice
- corporate travel mana gement
- Credit cards for small businesses
- Means of payment accepted widely
- Give more personalize d travel advice

Customer Relationships

- Convenience
- find the right tour
- Membership reward
- Assistance
- Security trust
- Make it much more easier and faster to

Channels

- Travel Guides (ex. Lonely Planet)
- Bank branches
- Internet
- Mobile App

Customer Segments

- Retail Customers
- Merchants
- Different size of businesses for card service and travel related service
- Travelers who are having a hard time planning

Cost Structure

- Contra revenue
- Customer rewards
- Salaries
- Marketing costs
- Can reduce customer service costs up to 30%

according to Chatbots Magazine

Revenue Streams

- Products : charge cards, credit cards, traveler's check
  - discount revenues
  - interest
  - net card fees
- Service: insurance, travel, finance
  - travel commission
  - improve overall travel service experience

Neural Network for classification problems

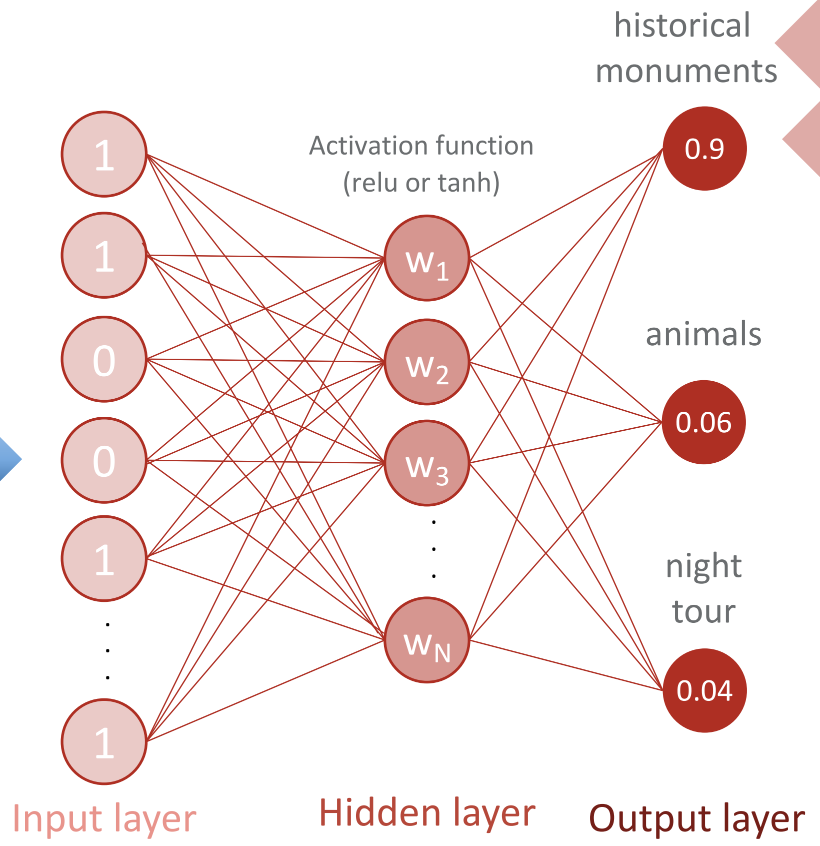
Sentence to classify
We have a list of intents and want to classify sentences into these classes

"Which places provide the best historic walking areas in Beijing?"

Bag of words vector
that contains the frequency of each word of a vocabulary bag

Which	1
the	1
what	0
why	0
places	1
.	.
.	.
.	.
walking	1

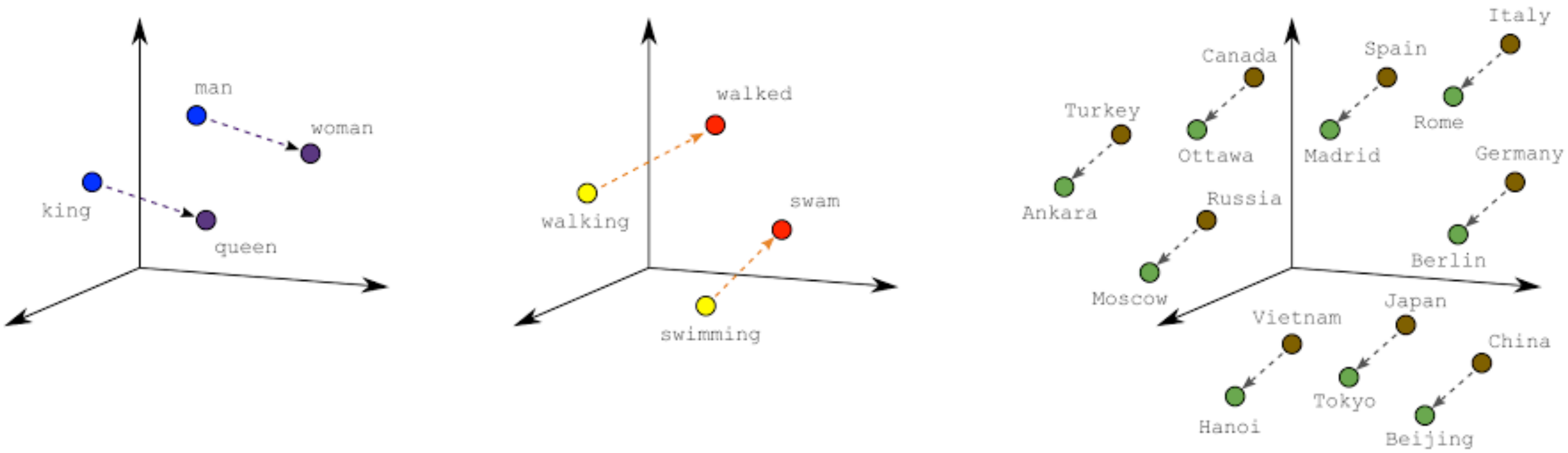
Conventional neural network
It only contains fully connected layers



Conclusion

- For a given input vector a neural network can fit proper weights so that it predicts the class it belongs to
- A conventional neural networks presents 3 main problems, it doesn't take account of the meaning of the words, the context of the words and the order of the words

Word embeddings to encode meaning



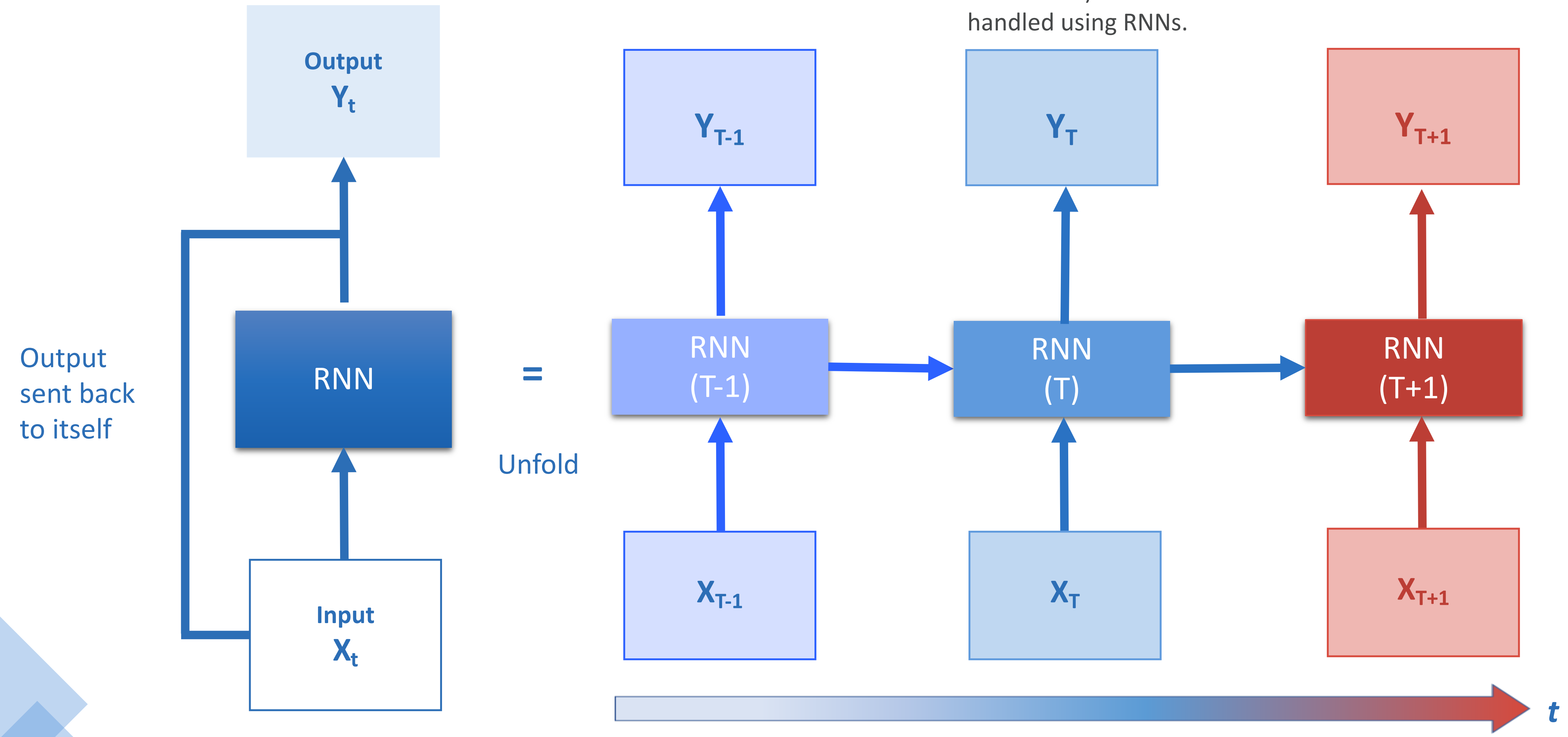
Male-Female Verb Tense Country-Capital

Conclusion

- Each word is encoded into a vector that describes the meaning of each word in a N dimension space
- This meaning is fixed for a given embedding matrix, this approach solves the meaning issue but not the context

Why RNN for intent classification?

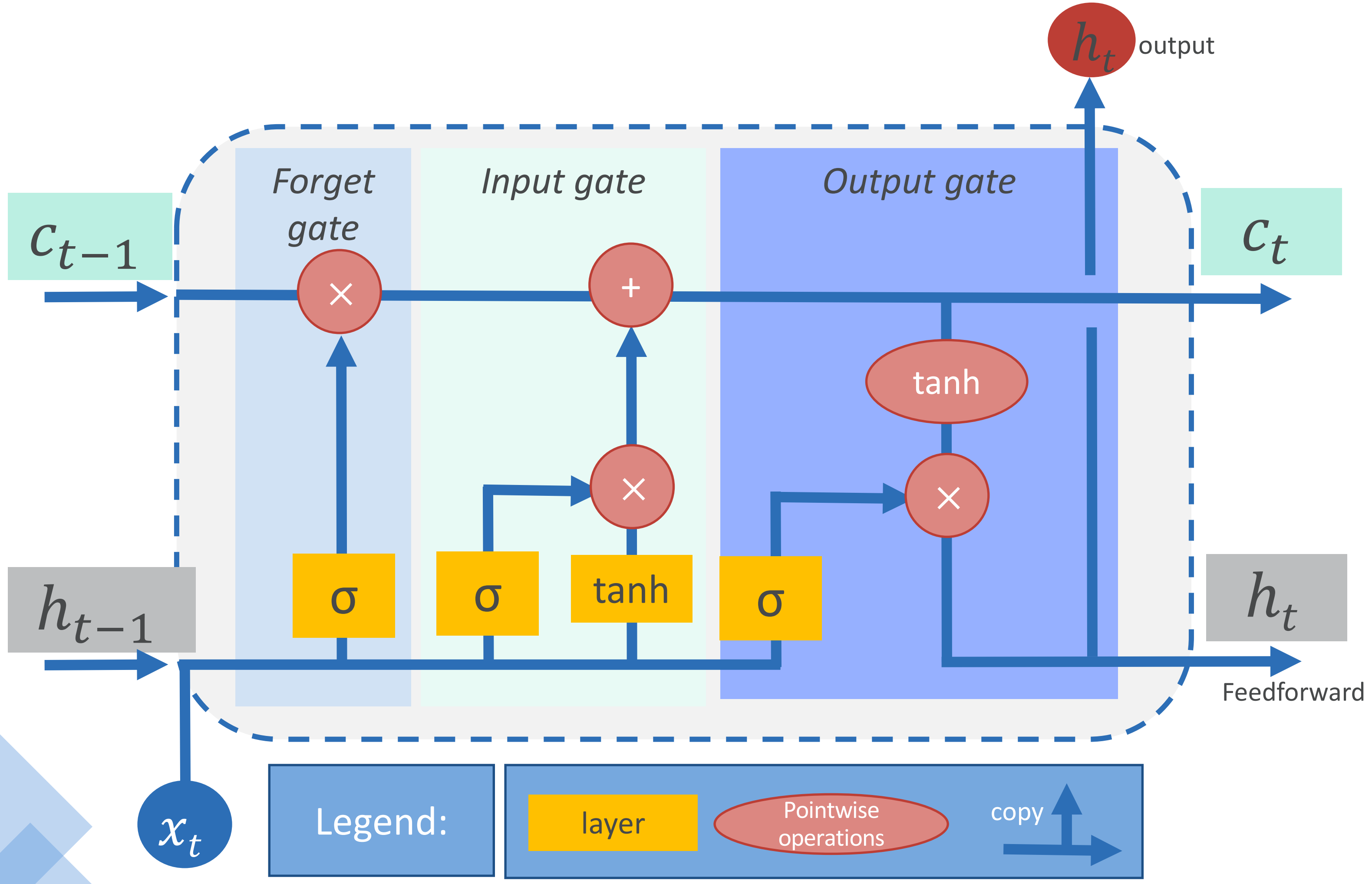
RNNs are good in handling sequential data (use for time series). The order in the sentence could be handled using RNNs.



Conclusion

- RNNs allow the model to take account of the context using feedback loops
- A problem subsist, the sensitivity decays exponentially over time as new inputs overwrite the activation of hidden units and the network forgets earlier inputs. This is the vanishing gradient problem

Using LSTM unit to improve our RNN



How does this work?

- Forget gate helps to decide on what to remove from h_{t-1} in order to keep only relevant things
- Input gate returns the cell state by adding the output of the forget gate to the activated input vector
- Output gate uses the sigmoid activated input vector and the cell state to return the output h_t

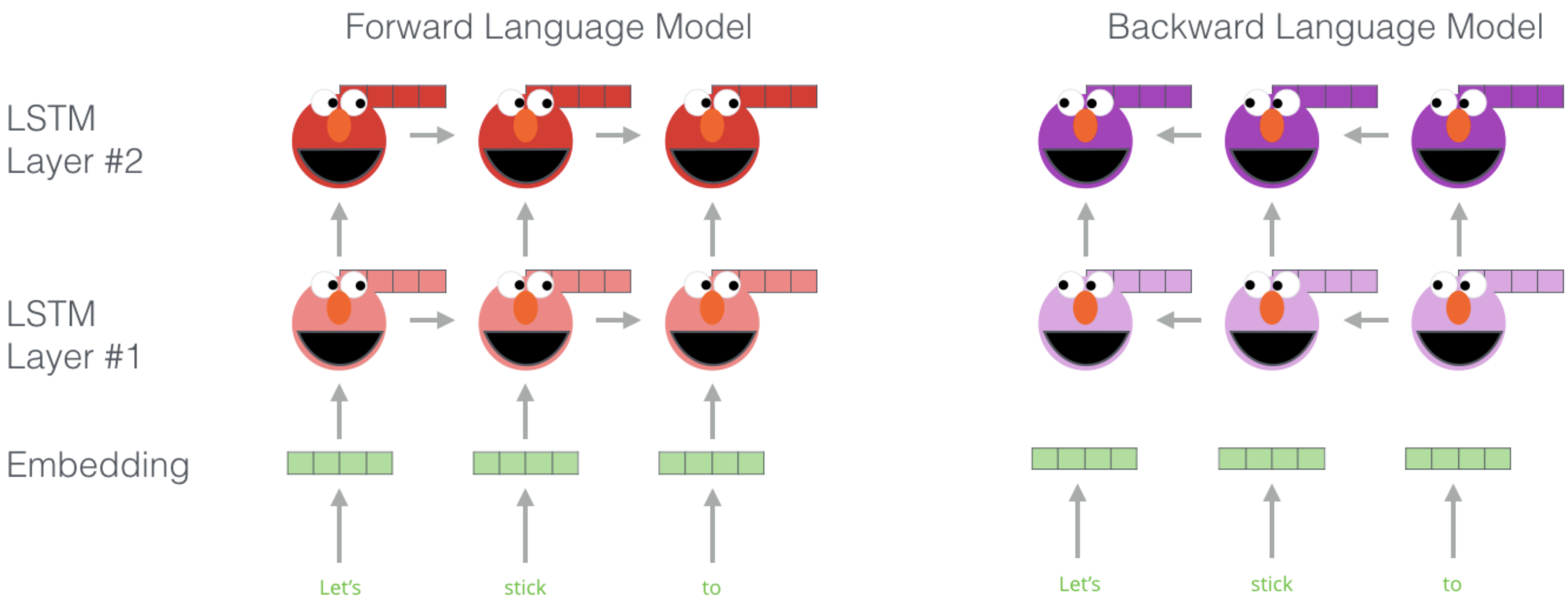
Abbreviations:

h: hidden state vector, aka output vector of the LSTM unit
c: cell state
x: input vector of the LSTM unit

Conclusion

- Forget gates decide which of the previous words are relevant, which solves vanishing gradient problems
- An embedding layer combined with LSTM properly takes account of the meaning and the orders of the word, but the context is only partially tackled by a forward/backword feedback (unidirectional)

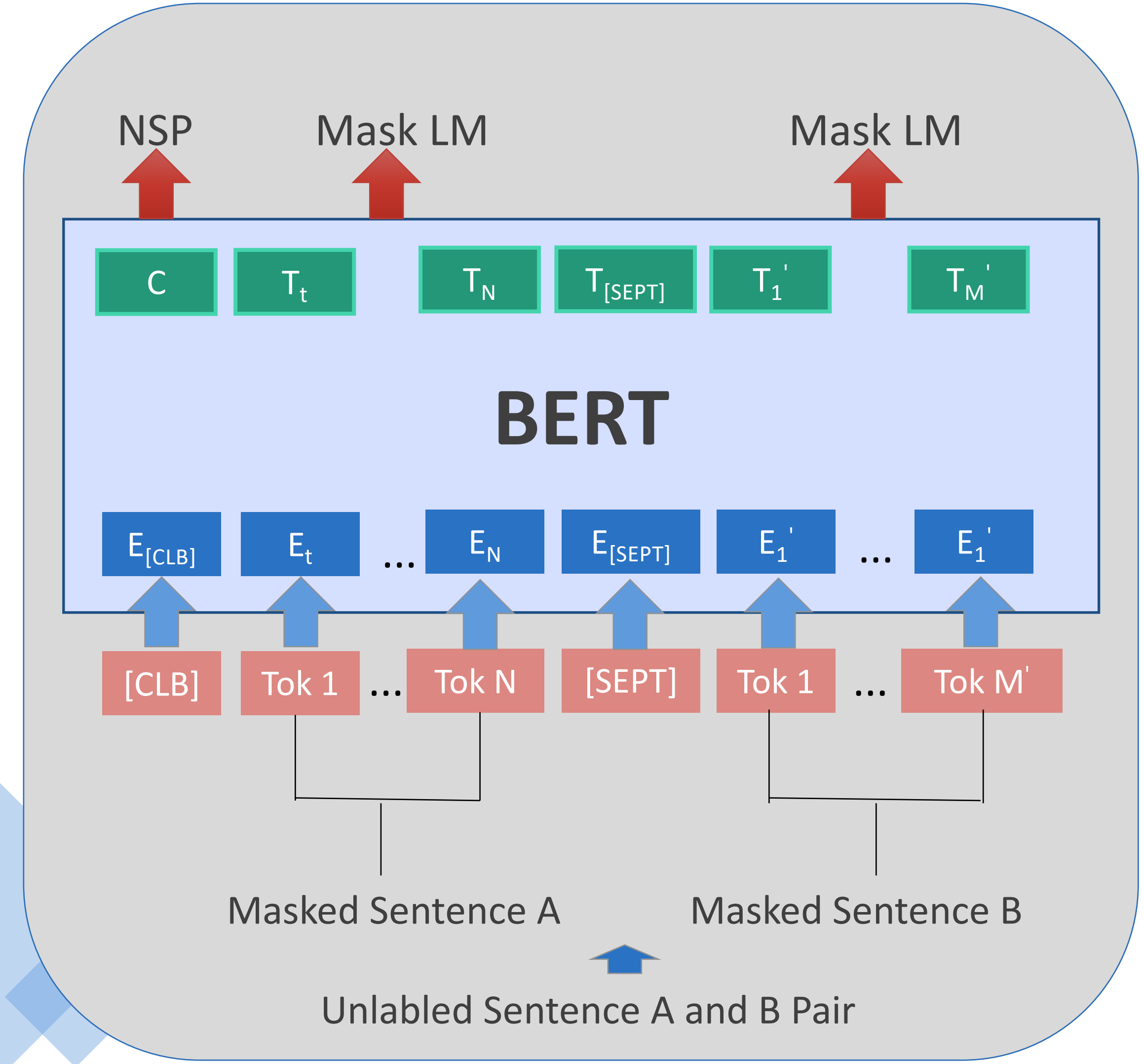
Why bidirectionality is all we need?



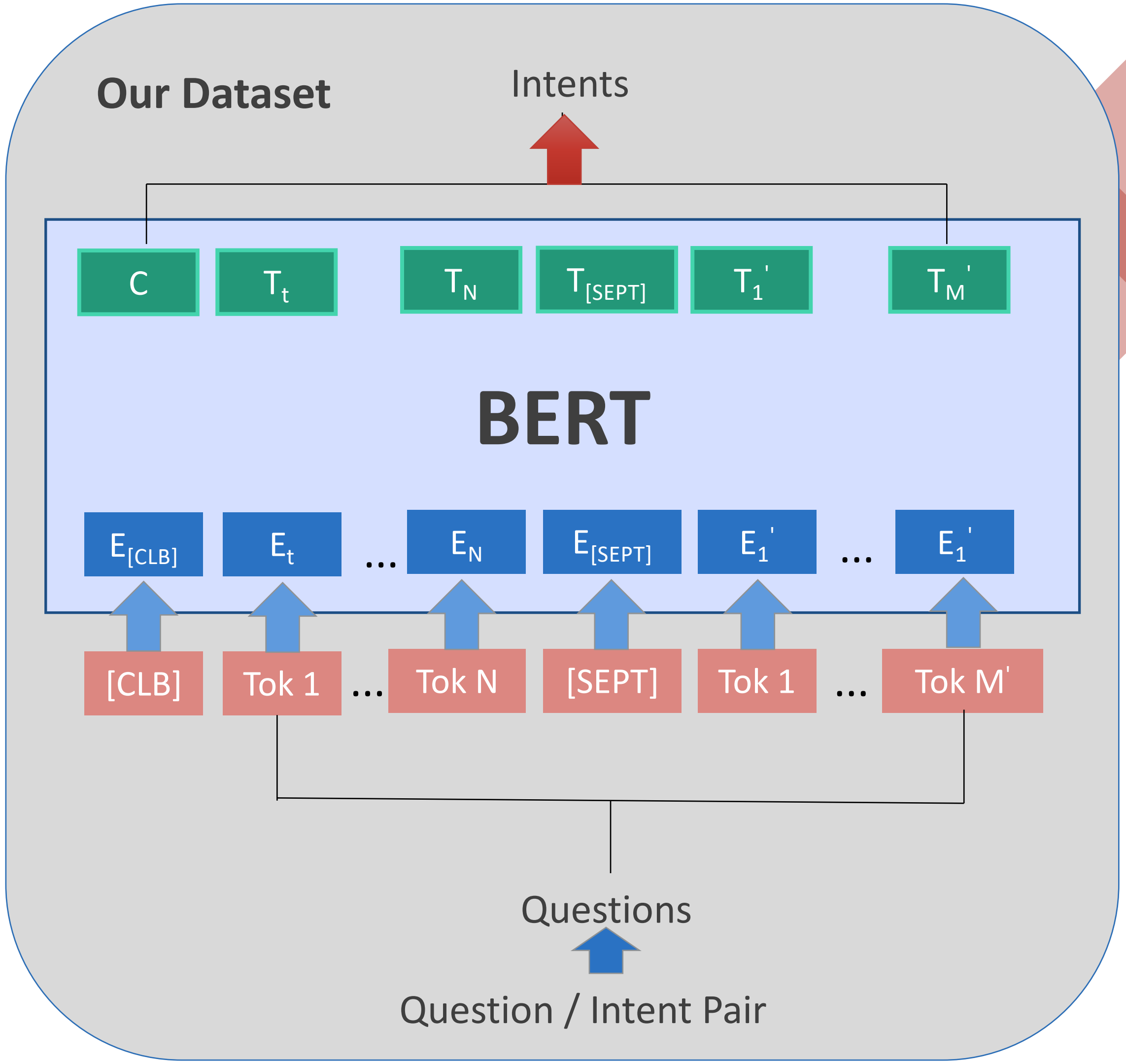
Conclusion

- A LSTM-RNN needs to choose to feed information forward or backward. In either cases, it advantages the beginning or the end of the sentence
- To simulate the human mind during reading, a model needs to look at the whole sentence

What is Bidirectional Encoder Representations from Transformers (BERT)? How did we fine-tuned it on our dataset?



Pre-training



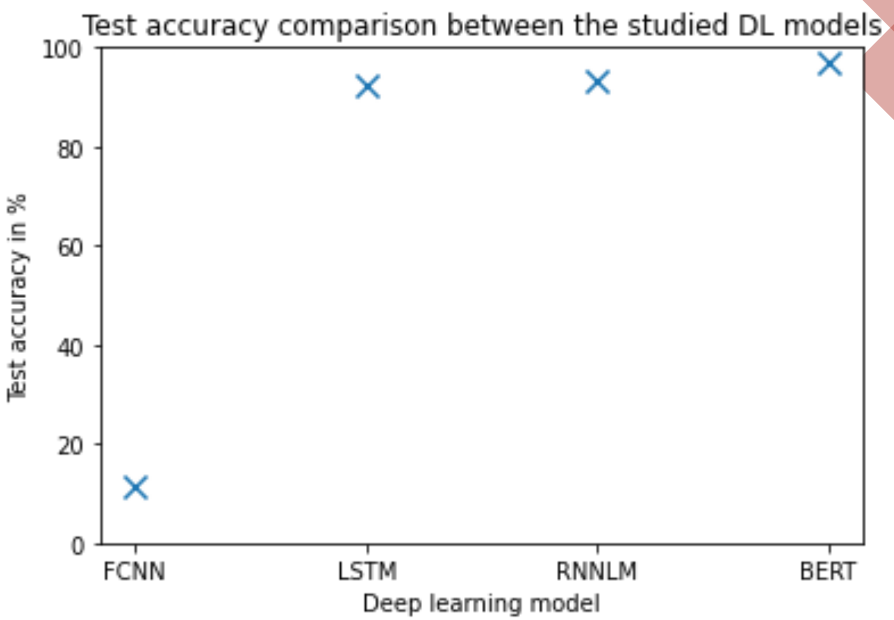
Fine-Tuning

Let’s test these models on real tasks: Intent classification for chatbot design

Comparative complexity of the model used

Model	Pre-trained	parameters	meaning	order	context
FCNN		430k			
LSTM		200k	X	X	/
RNNLM	X	48M	X	X	/
BERT	X	110M	X	X	X

Test accuracy comparison

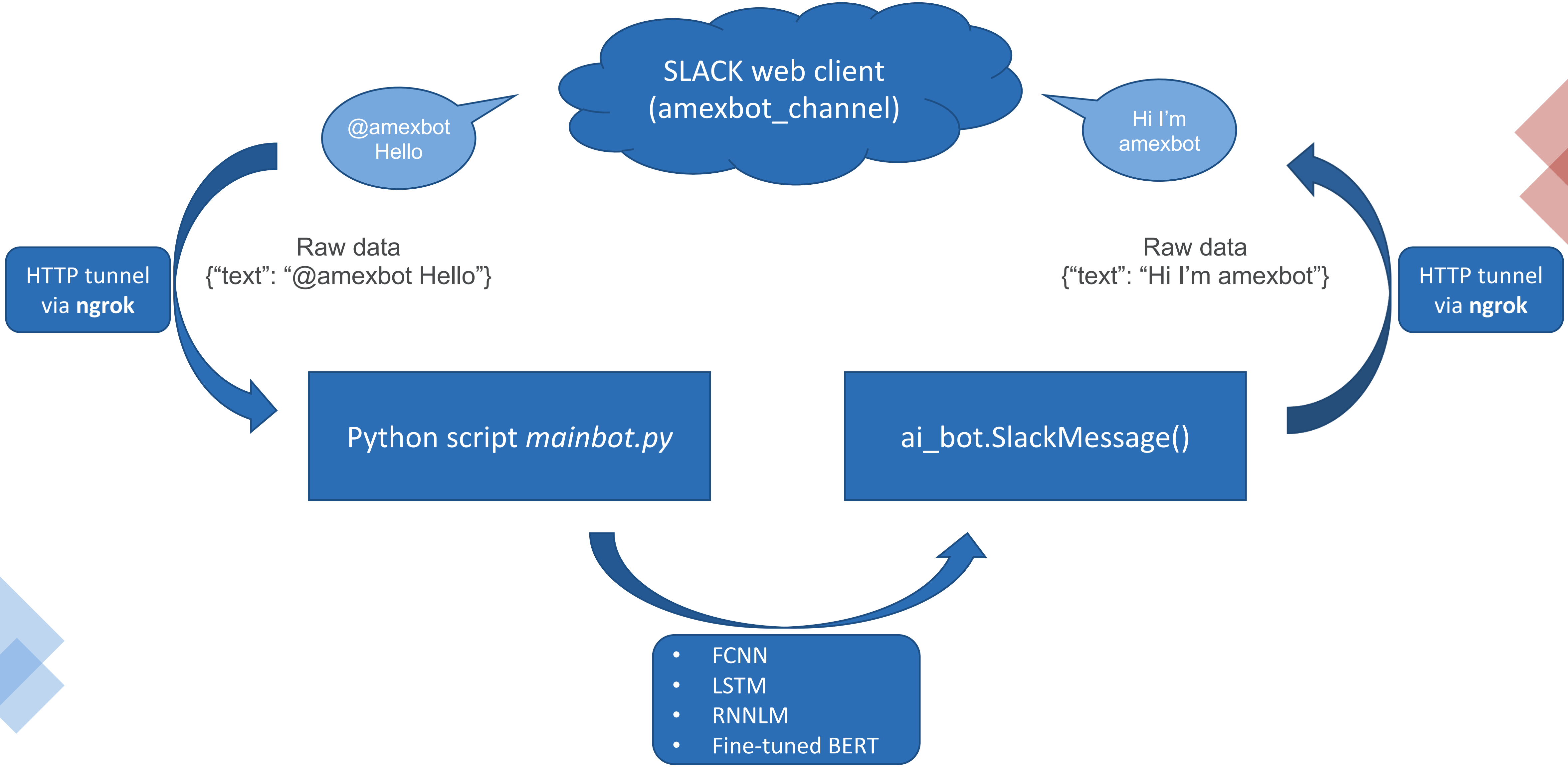


Conclusion

- The pre-trained models usually leverages on their training on a large dataset to give better performances
- The addition of complex features improves the performance of the models. BERT has a test accuracy of 96.9% by using only 4 epochs to train

For more details on the training procedure of the models, please look at the notebook of the github

A chatbot exposed on a Slack channel



The main components of the amexbot Software

Python class library: *ai_bot.py*

class Model*():

- The wildcard character (*) replaces:
  - Fcnn (Fully Connected Neural Network)
  - Rnnlm (Reccurent NN Language Model)
  - Lstm (Long Short-Term Memory RNN)
  - Bert (Bidirectional Encoder Representations from Transformers)
- The classes all contains these 3 methods:
  - _get_input_array() #preprocessing
  - _build() #build the NN architecture
  - _train() #train the model and save the best performing iteration

class LoadingData():

- Loads data from the "intents.json" data file
- Contains preprocessing methods

class SlackMessage():

- Methods to loop over Model class objects and predict answers from sentences (used by *main.py* and *mainbot.py*)

Python executables: *main*.py*

Python script: *main.py*

- 4 DL models for intent prediction
- A while loop for question-answering
- Automatically generated answers /predictions

Python script: *mainbot.py*

- Answer to event triggers (Slack Event API)
- Post answers / predictions to Slack
- Possible interactions with more people

Further developments

- Add more complexity in the intent labeling (consider more than one country, the preference of the client, the price, …)
- Expose the chatbot on a proper website (using wordpress and javascript)
- Add diversity of sentences in the dataset (we could use conversation data from Slack)
- Add more noise in the dataset using data augmentation

Conclusion

- By using state-of-the-art NLP methods, we managed to solve an 8-label classification problem with 893 data points. Our best performing model, fine-tuned Bert gives extremely high accuracy on the test set
- Further developments could further increase the performance of our deep learning models
- To interact with our chatbot and test its robustness, please feel free to join our team on Slack and leave our bot some message on a travel in Beijing for visiting **historical** places, seeing **animals**, **relaxing**, participating to **night tours** or **shows/concerts**, going on a **cruise** or eating good **food**.