My Chatbot lightblue: Course Project Report

MA Mingyu Derek, 14110562D, November 30, 2017

Associated code: github.com/derekmma/chatbot-trainer

1 Preparation

Training data lies the solid foundation for lightblue to grow up. For data preparation, **what kind of data is needed** will be discussed in 1.1, **where can I obtain proper data** will be presented in 1.2, **how to preprocess data and ethics issues** will be discussed in 1.3, and **topics selection** will be stated in 1.4.

1.1 Data Needs Analysis by Pretraining

The needs for training data is analyzed by pretraining and literature reviews. I firstly feed in some training data to see the response from the chatbot and I also learned from instructors and papers. The needs for training data may include:

Full Sentence Structure and Consistent Grammar (C1)

Not Too Long Sentences (C2): I found the chatbot often give the right answer for first half part, but wrong for else. So I guess maybe **generator model** is used which calculate probabilities word by word. Then shorter sentences can reduce the risk for wrong answers.

Wide Coverage of Vocabulary (C3)

Commutative Q&A Interactions (C4): I found the chatbot may link the sentences before or after its answers to learn, so it is important to let chatbot not only be an answerer, but also an asker.

1.2 Dataset Choosing

Based on the training data needs, I found several commonly-used datasets in academia and do a comparison. Finally the Eslfast dataset is chosen.

Dataset	C1	C2	C3	C4
NUS SMS Corpus [1]	N	Y	Y	Y
Cornell Movie Dialogs [2]	Y	N	Y	Y
Cornell Court Dialogs [3]	Y	N	Y	N
UCSB Spoken English [4]	N	N	Y	Y
Eslfast [1]	Y	Y	Y	Y

1.3 Data Preprocessing and Ethics

All collected data will be preprocessed to remove the out-of-vocabulary words, shorten the sentence length, enhance commutative elements. This preprocessing is first done by a `Python` program and then double checked manually.

Training data model the personality of lightblue. Clean and polite training data can make sure the ethics issues of the chatbot can be controlled.

1.4 Topics Selection by Vocabulary Analysis

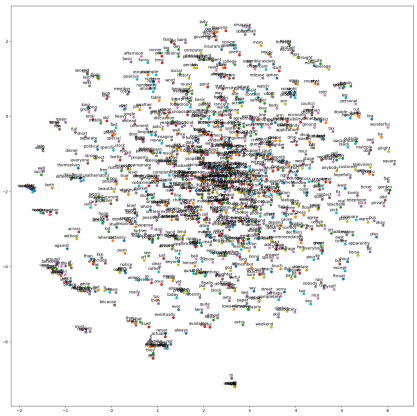
Some most common topics and potential topics for assessments should be first trained. To investigate what kind of topics may be important, a vocabulary analysis is implemented using `Python` with `TensorFlow`. Firstly most commonly used 50000 words[2] will be embedding to word vectors according to their meanings by *word2vec*, a state-of-art computational-efficient predictive neural-network-based model [5]. Then we plot the vocabulary to check the semantic relationships between words. But there is no clear clusters of words, so I just select 11 commonly discussed topics in daily life[3]. For each dialog under a

[1] https://www.eslfast.com/

[2] Obtained from http://mattmahoney.net/dc/textdata.html

[3] All my training data: https://github.com/derekmma/chatbot-trainer/tree/master/chrome_ext/data/topics

topic, there are three mutations of this dialog with slightly different language to increase the robustness.

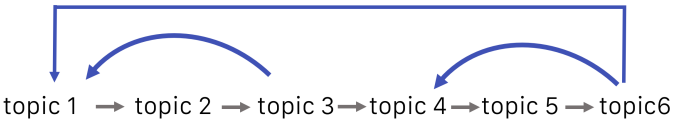


2 Training Strategies and Observations

After preparation, I restarted the chatbot and start to train. **47 dialogs, 270+ unique conversations, 8219 total sentences are fed in.**

2.1 Simulated Annealing and Training Flow

Based on the liteature reviews of common chatbot structure, I found **RNN and LSTM are two most common ones**. But both of them do not have a satisfying memory performance. So inspired by Simulated Annealing algorithm, I plan to train the input data in a cycle flow:



So proper repeating can "cool down" the high-entropy new incoming conversations and let the chatbot settle the structure and knowledge.

2.2 Repeating and Its Effect

The average repeating ratio in my training is 15. In other word, each sentence goes through 15 iterations for feeding. About 15 times repeatings can make the chatbot remember the basic content of the response. **According to the RNN/LSTM structure, the sequence of the inputs can affect the internal memory of neurons**, so inside each repeating, I try to keep the sequence between conversations the same, so that no significant change is needed for adjusting the weight between neurons. In practice, I found if I trained topics in sequence like: $< 1, 2, 3, 1, 2, 3, ..., 3, 2, 1 >$, it will take longer rumination time compared with sequence like $< 1, 2, 3, ..., 1, 2, 3 >$. This finding also verifies that the internal structure may be RNN/LSTM.

2.3 Significant Rumination Effect

Benefited from the training strategy above, the rumination effect is significant. **After several repeating, a rumination can significantly improve the performance**. Before rumination, the response is like "I will it it it...", after it the response is very smooth like "it will rain".

2.4 Long Rumination Time

During my training, **I met several times very long rumination time**. Even when my training data is smaller than 1000, I also used more than one hour to ruminate. A possible explanation is that at that time data has high-entropy where knowledge structure has not been settle down, so the "cool-down" rumination may take longer time.

2.5 Training Tools

To feed in data in a systematic way, **a Chrome extension using JavaScript is developed by me** to automatically feed in data of a topic, "change" the response, "like" the revised response and then open a new session for next topic. A demo video can be found[4].

3 Reflections

After knowing the exact structure of the chatbot, I realized that my repeating process can be replaced by systematic "Ruminate" operations. This can lead to faster training and better results. I should avoid the "big data" strategy which makes the history involute and crowded and hard to ruminate.

[4] https://youtu.be/flt2GLLF8so

REFERENCES

[1] T. Chen and M.-Y. Kan, "Creating a live, public short message service corpus: The nus sms corpus", *Language resources and evaluation*, vol. 47, no. 2, pp. 299–335, 2013.

[2] C. Danescu-Niculescu-Mizil and L. Lee, "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs.", in *Proceedings of the workshop on cognitive modeling and computational linguistics, acl 2011*, 2011.

[3] C. Danescu-Niculescu-Mizil, L. Lee, B. Pang, and J. Kleinberg, "Echoes of power: Language effects and power differences in social interaction", in *Proceedings of the 21st international conference on world wide web*, ACM, 2012, pp. 699–708.

[4] J. W. Du Bois, W. L. Chafe, C. Meyer, S. A. Thompson, R Englebretson, and N Martey, *Santa barbara corpus of spoken american english, parts 1-4. philadelphia: Linguistic data consortium*, 2000.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality", in *Advances in neural information processing systems*, 2013, pp. 3111–3119.