A Chatbot Based on Stackoverflow

Wen Zhong, Yiqing Liu

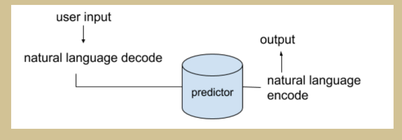
Stanford
CS221

motivation

when users are trying to find answers to a specific question online, they usually would get lots of relevant information, which is overwhelming. Sometimes it is challenging to find the best answer, especially for new beginners. Our goal is to build a chatbot, which could find the best matched question with the user query and return the top-rated answer of that question as answer to user query. This will save users' efforts for information filtering.
The underlying idea is to compute text similarity between user input and existing questions in our database
Keywords: NLP, Neural Networks, Autoencoder, text similarity, text match

problem definition

input: the sentence from user input
For example: How to import package in Python?
output: the top-rated answer of the best matched question according to user input
underlying problem: implement an approach to compute the similarity between input text and questions stored in DB efficiently



dataset: StackOverflow Data from kaggle
https://www.kaggle.com/stackoverflow/stackoverflow
StackOverflow is one of the largest online community for programmers to learn, share their knowledge.
After data cleaning, we would mainly use two tables:
questions: question_id, question_title, top_rated_answer_id
answers: answer_id, answer_content

questions

Schema | Details | Preview

Field name	Type	Mode	Description
qid	INTEGER	NULLABLE	
title	STRING	NULLABLE	
accepted_answer_id	INTEGER	NULLABLE	

answers

Schema | Details | Preview

Field name	Type	Mode	Description
aid	INTEGER	NULLABLE	
body	STRING	NULLABLE	

Stanford University

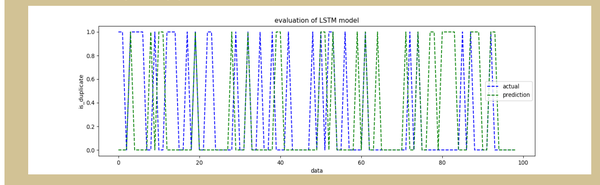
challenges

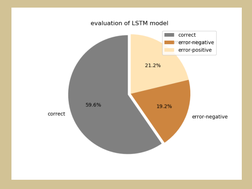
- **unsupervised learning**
  According to our dataset, it is difficult and expensive to label the data. In other words, the dataset couldn't tell us which pair of questions are most similar. Therefore, **supervised learning algorithms** are **inapplicable**.
  For example, with a labeled dataset we could perform LSTM algorithm.
  dataset: **Quora question pair dataset** https://www.kaggle.com/c/quora-question-pairs/data

	# id	# qid1	# qid2	A question1	A questio...	# is_dupli...
1	0	1	2	What is the step by step guide to invest in share market in india?	What is the step by step guide to invest in share market?	0

  - <qid1,qid2> represent the question pair
  - label field(is_duplicate) represents whether these two questions are duplicated(similar)

  We applied word2vector algorithm and implemented a basic LSTM model, the experimental accuracy is approximately **60%**



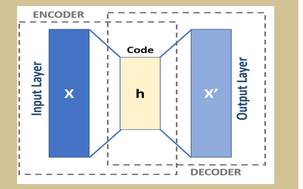


  According to the experiment, we could see the labeled data can be applied for a lot of supervised algorithms
  Here, we are given unlabeled dataset
  - challenging to develop an efficient model to automatically detect the similarities of texts

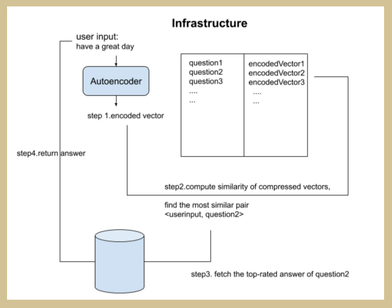
- **large dataset**
  - After data cleaning, the question table includes 7625030 records
  - the running time for computing the similarity of user input and questions in our database make sure user won't wait for too long to get the result

Approaches

autoencoder:
- a type of artificial neural network
- used to learn data in an unsupervised manner
- help find out a good representation of input big data



infrastructure:
our **goal** : to find the intermediate result - h, to represent the data from input layer.

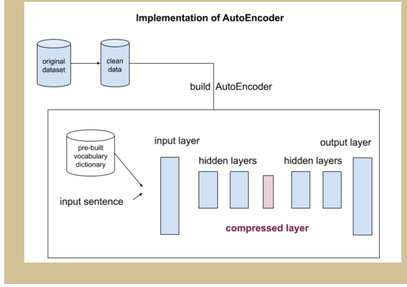


- implemented an Autoencoder for compressing the user input sentence to a short vector
- use that representation to compute the similarity between sentences
- find the most similar question in the database
- return the top-rated answer of that question.

the **similarity of two vectors** are measured in the following way:

$$\text{similarityScore}(\text{vector1}, \text{vector2}) = \cos(x) = \frac{\overrightarrow{vector1} * \overrightarrow{vector2}}{|\overrightarrow{vector1}||\overrightarrow{vector2}|}$$

implementation of AutoEncoder



dependencies:
language: Python3
packages: tensorflow, keras
data storage: BigQuery from Google Cloud

- **measurement of the performance**
  - difficult to test on the user input and outcome in the quantitative way
  - adopt another way to analyze the performance of Autoencoder

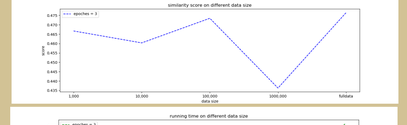
$$\text{similarityScore}(\text{vector1}, \text{vector2}) = \cos(x) = \frac{\overrightarrow{vector1} * \overrightarrow{vector2}}{|\overrightarrow{vector1}||\overrightarrow{vector2}|}$$

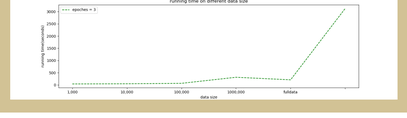
Notice: If the original vector is same with the vector after encoding and decoding, it means the intermediate vector could work as a good representative of the original vector.

results & analysis

- **testing results**

Testing on 20% of full dataset, the average similarityScore is 0.476496





- **running on different data size**
  - the similarity score is stable on different data size
  - running time increases with the larger dataset

Acknowledgments
This work was supported by Stanford University. The authors gratefully acknowledge the help from Prof. Percy Liang, Prof. Dorsa Sadigh and CA Haoshen Hong

5min presentation: https://youtu.be/rFjsGmN0XEU

Demo: https://youtu.be/jvU5nPcFhgA