Question Answering using Random Forests

Gao Ziyuan (Student Number: 1605045)

Journals Standard: IEEE Transactions on Knowledge and Data Engineering

Abstract—The bAbi tasks are a set of Question Answering dataset. We introduce the Random Forests, a machine learning method for processing input contexts and questions, building model and training question answering system. The text data are converted to vectors through a function and the suitable matrixes are sent to Random Forest model. The model is trained by the training dataset and returns the accuracies with the testing dataset. The evaluation of the model shows the bad performance of processing question answering data and improvements are required in the future.

Index Terms—Question Answering, Random Forests, Accuracy

1. INTRODUCTION

Question answering is a natural language processing task. It extracts useful facts from the information in the datasets and provides appropriate answers to a natural-language question. An example of question answering is

```
1 John travelled to the hallway.
2 Mary journeyed to the bathroom.
3 Where is John?    hallway 1
4 Daniel went back to the bathroom.
5 John moved to the bedroom.
6 Where is Mary?    bathroom    2
7 John went to the hallway.
8 Sandra journeyed to the kitchen.
9 Where is Sandra?  kitchen 8
10 Sandra travelled to the hallway.
11 John went to the garden.
12 Where is Sandra?     hallway 10
```

shown in Figure 1.

Fig 1. Example stories, questions and answers. The first column is line number and the number behind the answer is "ids" (supporting fact line).

The project aims at building a model for question answering by using a machine learning method called Random Forests.

The datasets come from bAbI tasks, which is a set of QA tasks for Artificial Intelligence. The training data and testing data are both 20 tasks in "en/" directories and each task includes 1000 questions.

Every word in the text data is converted to numeric representation with a word dictionary. The whole numeric matrix is divided into two parts. One is a matrix including all the numeric representation of stories and questions and another is a matrix indicating the answers.

These two matrixes are sent to a new model built by Random Forest Classifier and are utilized to train this model with the training datasets. The testing datasets will help the model process the evaluation and return the accuracy.

The completed model is an unchanged forest after training and testing. When inputting the specified story and question, the probable answer will be displayed.

2. BACKGROUND

2.1 Dynamic Memory Networks (DMN)

There are several researches about DMN models for question answering tasks, especially the article named "Ask me anything: dynamic memory networks for natural language processing" [1].

This research is based on Facebook's bAbI dataset and explores the characteristics of DMN model. The dynamic memory networks in this research consist of input module, question module, episodic memory module and answer module. The first two modules need to encode the text data into vectors. The episodic module collects the inputs and produces memory vectors according to the question vector and previous memory. The answer module generates the answer from the final memory vector. Gated recurrent network (GRU) is used and LSTM is explored. Both work better than the recurrent neural network (RNN). The key point of test processing is text classification and the authors perform sentiment analysis to train their DMN models. After the experiments, this research states the evaluation of DMN model. DMN model is a general architecture for NLP applications [1]. This model can be compared with Random Forest model in this project.

2.2 Random Coattention Forest Network

The scholars from Stanford University perform a research for building a Random Coattention Forest Network for question answering [2].

Their dataset is Stanford Question Answering Dataset

(SQuAD) and is utilized to train and evaluate the model. The text data of questions and contexts are regarded as inputs to encoder and decoder. The parameter "start idx" and "end idx" are used to judge the location of supporting facts and output the answer. The network includes encoder model, LSTM classification decoder and random forest network. The random dropout and the random start make encoder-decoder pairs loosely correlated with each other and this contributes to random forest network. The research observes performance of multiple encoder-decoder pairs in exact match and F1 score. The model solves question answering problem to a satisfactory level [2]. The method of processing text data and the structure of this model have an influence on this project.

2.3 Sentiment Analysis

Sentiment Analysis is a higher-level method compared to judgement from supporting text facts.

This article [3] explores the utility of sentiment analysis and semantic word classes for improving question answering system. The authors use TinySVM to evaluate all the systems and find semantic features are effective [3].

This research introduces another method to deal with question answering tasks and it is helpful to future work of this project [3].

3. METHODOLOGY

The overview of the modules in this project is given and a detailed illustration of the whole structure is shown in Figure.2.

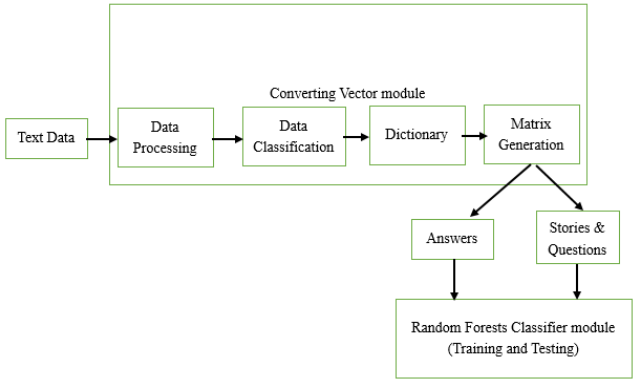

Fig.2 The Structure of the Project

3.1 BAbI Tasks

The tasks are available at http://fb.ai/babi [4]. Source code to generate the tasks is available at http://github.com/facebook/bAbI-tasks [4]. The data file can be downloaded from the course website. The bAbI tasks include four directories and "en/" directory is processed in this project. There are 20 tasks in the directory and the data is divided into training dataset and testing dataset. In this module, the data is loaded at the beginning.

3.2 Converting Vectors module

In this module, the text data from bAbI tasks are loaded and processed properly. The words of tasks are converted to numeric representation and the representation of sentences are fixed length vectors. Finally, the matrixes on the basis of data contexts are generated.

This module is defined as a function in the program and the function can be called by training datasets or testing datasets to create available matrixes.

3.2.1 Text Data Processing

As is shown in the data, the context of each task consists of line number, stories, questions, answers and 'ids' (line numbers which indicate supporting facts).

Obviously, line number and "ids" should not be included in the targeted matrixes. Therefore, when reading each line of text data, loading of words starts from the second string. When reading the lines including question, the program exits the loop after loading answers.

Additionally, the punctuation in the sentences need to be removed from the text data. This is achieved by a conditional statement in the program.

3.2.2 Data Classification and Application of Dictionaries

The data is classified into two parts, which are stories with corresponding questions and answers.

In order to separate the data, the signal for whether the present line includes question is set. The index of story increments when the line number is 1 or the last line includes question. The index of number increments every line and initializes when starting a new story. The index of question and answer are the same and they are determined by the signal.

Dictionaries are widely used in python. It includes keys and values that the keys point to. In this module, words are stored in a dictionary and the corresponding values are the increasing integers beginning from 1. All the words are added into the dictionary and are given a unique value.

According to the indexes and question signal, stories with questions and answers are converted to numeric representation separately.

3.2.3 Generation of Suitable Matrixes

The stories with the corresponding questions are the characteristic values in random forests. The answers are the targeted value in random forests.

Therefore, suitable matrixes should be stories with questions matrix and answers matrix. The first matrix needs to have

two dimensions (index of story, index of word) and the second is one-dimensional matrix (index of answer).

The generation is based on the specified dictionary created by the words in the data and conditional statements are utilized to value matrixes.

3.3 Random Forests Classifier module

The Random Forest model is trained by the training dataset and evaluated by the testing dataset. The accuracy of the model is returned in the end.

3.3.1 Random Forests

A Random Forest consists of many decision trees. Decision trees are generated randomly by the characteristic values. Every time a random forest model is built, the trees are different. Every decision tree will create an alternative targeted value and the value of the most frequency is regarded as the final output of this model [5-6].

3.3.2 Model Building

The RandomForestClassifier function from scikit-learn library contributes to building Random Forest model. The function has many parameters and attributes. The parameter "n_estimators" represents the number of decision trees and the higher "n_estimators" will improve the model. The other parameters can be set as default value.

3.3.3 Model Training and Testing

The stories with questions matrix is set as characteristic value X and the answers matrix is set as targeted value Y. The pairs created from the training dataset are sent to the model for training. The testing dataset is used for evaluating the model and the accuracy is the key index of the evaluation.

4. EXPERIMENTS

The experiment performed in this project is utilizing the random forest model to deal with question answering data. The numeric representation of training dataset including 20 tasks trains the model and the testing dataset evaluates the completed model. The accuracy of each task and the mean accuracy is shown in Table 1.

While getting the value of accuracy from separated task, there are four tasks which fail to return the accuracy. The command bar displays the n_features of training dataset and testing dataset is not the same value and the model cannot return the accuracy.

TABLE 1 Test accuracies on bAbI tasks with RandomForests (N/A means features of train and test datasets not equal, cannot calculate accuracy)

Task	RandomForests
1_single-supporting-fact	47.20%
2_two-supporting-facts	N/A
3_three-supporting-facts	N/A
4_two-arg-relations	60.40%
5_three-arg-relations	N/A
6_yes-no-questions	N/A
7_counting	68.20%
8_lists-sets	N/A
9_simple-negation	65.30%
10_indefinite-knowledge	54.30%
11_basic-coreference	59.10%
12_conjunction	61.00%
13_compound-coreference	78.60%
14_time-reasoning	20.00%
15_basic-deduction	23.30%
16_basic-induction	37.70%
17_positional-reasoning	49.60%
18_size-reasoning	85.90%
19_path-finding	10.90%
20_agents-motivations	92.20%
Mean accuracy (%)	54.25%

5. DISCUSSION

The dictionaries in python is a simple method for converting the words to numeric representation. This is easy to achieve but the relationship between the words is ignored. There are some available ways for vectors converting. Word2Vec creates word pairs with a specified window size and uses them to train the neural network. The output vectors are the cosine similarity between the central word and other words. Additionally, GloVe is the improvement of Word2Vec and is recommended to use.

From the results of experiments, the accuracies of the Random Forest model can be used to compared with MemNN model and DMN model. The comparison is displayed in Table 2.

TABLE 2 Comparison of Test Accuracies [1]

Task	RandomForests	MemNN	DMN
1_single-supporting-fact	47.20%	100.00%	100.00%
2_two-supporting-facts	N/A	100.00%	98.20%
3_three-supporting-facts	N/A	100.00%	95.20%
4_two-arg-relations	60.40%	100.00%	100.00%
5_three-arg-relations	N/A	98.00%	99.30%
6_yes-no-questions	N/A	100.00%	100.00%
7_counting	68.20%	85.00%	96.90%
8_lists-sets	N/A	91.00%	96.50%
9_simple-negation	65.30%	100.00%	100.00%
10_indefinite-knowledge	54.30%	98.00%	97.50%
11_basic-coreference	59.10%	100.00%	99.90%
12_conjunction	61.00%	100.00%	100.00%
13_compound-coreference	78.60%	100.00%	99.80%
14_time-reasoning	20.00%	99.00%	100.00%
15_basic-deduction	23.30%	100.00%	100.00%
16_basic-induction	37.70%	100.00%	99.40%
17_positional-reasoning	49.60%	65.00%	59.60%
18_size-reasoning	85.90%	95.00%	95.30%
19_path-finding	10.90%	36.00%	34.50%
20_agents-motivations	92.20%	100.00%	100.00%
Mean accuracy (%)	54.25%	93.30%	93.60%

It is obvious that MemNN model and DMN model have a better performance than Random Forest model when processing the same data. The Random Forest model is not suitable for question answering data. This model cannot link words well with sentences and similarly cannot combine sentences well with the large context.

On the contrary, DMN model is stable and the accuracy of the model is quite high. It is a good choice for natural language processing.

6. CONCLUSION

Through the project, the bAbI tasks data are converted to numeric representation and used in the Random Forest model. This model is an alternative method for question answering data but from the evaluation, it is not very suitable. In the future, by adjusting the parameters of random forests, this model is expected for improvement. Also, some other method such as DMN and MemNN will be tried.

In addition, the method of converting data to vectors is not limited with numeric conversion. Word2Vec plays an important role in this area and the improvements of this technique need to be explored. From the lessons about Deep Learning in Stanford University, mixed application of Singular Value Decomposition (SVD) and Word2Vec is a better method. SVD can reduce the dimension of the data and complex large data can also be solved in nowadays.

Future work will focus on the study of Word2Vec and the improvement of Random Forests model and other multi-task models.

REFERENCES

[1] A. Kumar, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani and V. Zhong, "Ask Me Anything: Dynamic Memory Networks for Natural Language Processing", arXiv preprint arXiv:1506.07285v5, 2016.

[2] J. Chen, T. Lee and Y. Chen, "*Random Coattention Forest for Question Answering*", [Online]. Available: https://web.stanford.edu/class/cs224n/reports/2760668.pdf.

[3] J. Oh, K. Torisawa, C. Hashimoto, T. Kawada, S. Saeger, J. Kazama and Y. Wang, "Why Question Answering using Sentiment Analysis and Word Classes", *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012.

[4] J. Weston, A. Bordes, S. Chopra, A. Rush, B. Merrienboer, A. Joulin and T. Mikolov, "Towards Ai-complete Question Answering: A Set of Prerequisite Toy Tasks", arXiv preprint arXiv:1502.05698, 2015.

[5] B. Leo and C. Adele, "*Random Forests*". Internet: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#missing1.

[6] H. Ned, "*Introduction to Decision Trees and Random Forests"*, American Museum of Natural History's Center for Biodiversity and Conservation.