Project Proposal

Submitted as part of the requirements for:

CE888 Data Science and Decision Making

Name: Zhao Li

Prid number: 1606786

Supervisor: Spyros Samothrakis

Date: 22 February 2017

Abstract

This project is aimed to create random forests to process the dataset from bABi tasks to simulate a system of question answering for artificial intelligence. By using methods similar with word2vec, input questions are converted into operative vectors, and then through random forests, an accurate answer will be provided. The result shows that this project is able to respond correctly to a testing question.

Keyword: random forests, bABi, vectors

Introduction

This paper focuses on the creation of random forests applied to the bABi tasks which include data about numbers of questions answering tasks for Artificial Intelligence. In this project, tasks sentences are transformed into fixed length vector representations. Then these vectors are fed into random forests, and finally the expected answer will be printed out. Word2vec is an efficient method to convert word into vectors. The core concept of word2vec can be implemented in this project. With inputting target sentences into the program, the corresponding representations of vectors will be produced to be ready for the random forests.

Random forest is a machine learning algorithm which is commonly used for large datasets. According to the features of target dataset, many random decision trees are created. And random forests are the combination of these decision trees. [1] For this project, because there are 20 different types of tasks in the bABi dataset, 20 random forests are created respectively. In each forest, all the classes in the corresponding task should be included by creating uncorrelated decision trees. After the classification of different features, each random forest are supposed to respond correctly to the target vectors.

Background

There are many projects which use random forests as an algorithm to research targeted dataset. Tanaka's random forests-based early warning system is a system to analysis whether banks are in danger of failing according to the current bank-level financial statements. The project uses dataset from BankScope. The researcher classified banks into "active banks" and "inactive banks". Through random forests, three most important variables are provided which distinguish active and inactive bank. By operating these three variables, the system is able to predict bank failure. Comparing with other methods of early warning system, this system is more accurate. Moreover, it concludes that there are 730 banks in danger with assets equivalent to about 95.3 million US dollars in total. [2]

Methodology

The dataset of this project is from bABi tasks, which was a set of prerequisite toy tasks generated by an algorithm invented by Weston. These tasks are consisted of many contexts which behave like a classic text adventure game. [3] To begin with, all the sentences in the

dataset need to be converted into fixed length representations of vectors through a similar method to word2vec. Most importantly, the key challenge of this project is the creation of random forests. There are 20 distinctive types of tasks in bABi tasks, so it is essential to create a special random forest for each task. Then according to the types of input vectors, the program will choose the corresponding forest, and answer rightly. For each task, there are N random decision trees in the forest; in each decision tree, M random classes are chosen from the task file; every selected class includes all the values of a kind of feature. When an input vector is inputted, the corresponding forest is selected from the 20 forests according to the features of the vector. Then the vector is fed to that forest, all the decision trees will give it a classification according to the features of the input vector. The forest will calculate the occurrences of each kind of classification given by the decision trees. At the end, the forest will choose the classification which is "voted" most. [1]

Experiments

In bABi tasks, for each task, there are 1000 questions for training, and 1000 for testing. For this project, training data are used to create random forests. Firstly, N random decision trees are created base on a specific number of questions (the number must be smaller than 1000). As long as N is big enough, all the dataset will be included, which attributes to the accuracy of the whole forest. Due to the truth that there are many features in every task, every decision tree should operate all the features. For example, for the first task, which is a basic factoid QA with single supporting fact, the features are "name", "place", "action", and the specific feature for the question is "where" related "place". The decision trees are able to classify these features into different classes. Class "name" is composed of all the names, and class "action" is corresponding to all the actions. In addition, the class "place" as the aimed class is composed of all different places. Every random decision tree in the forest will give a value from "place" class as an answer to the input question with context. And the forest aims to choose the answer which has the highest frequency of occurrence. The following structure diagram1 gives a brief understanding of this project.

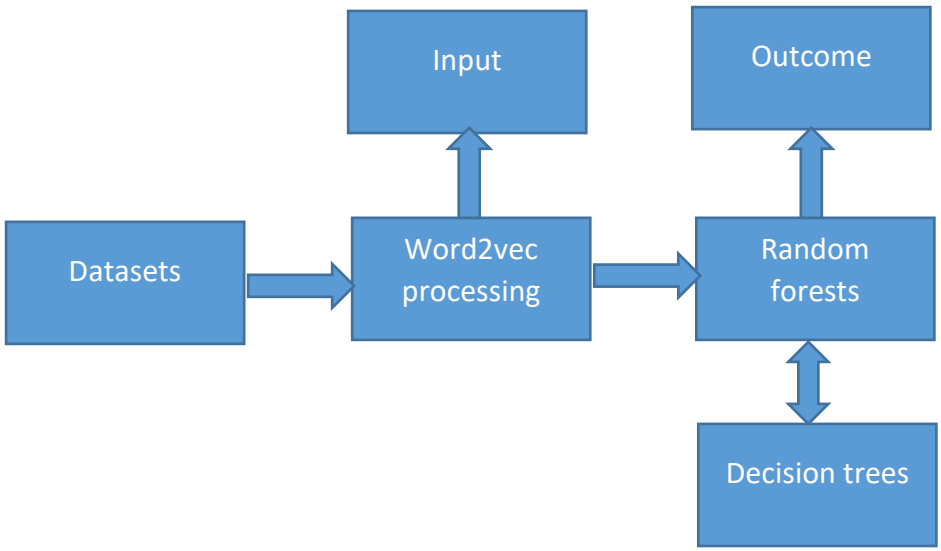


Diagram 1 Brief structure of the project

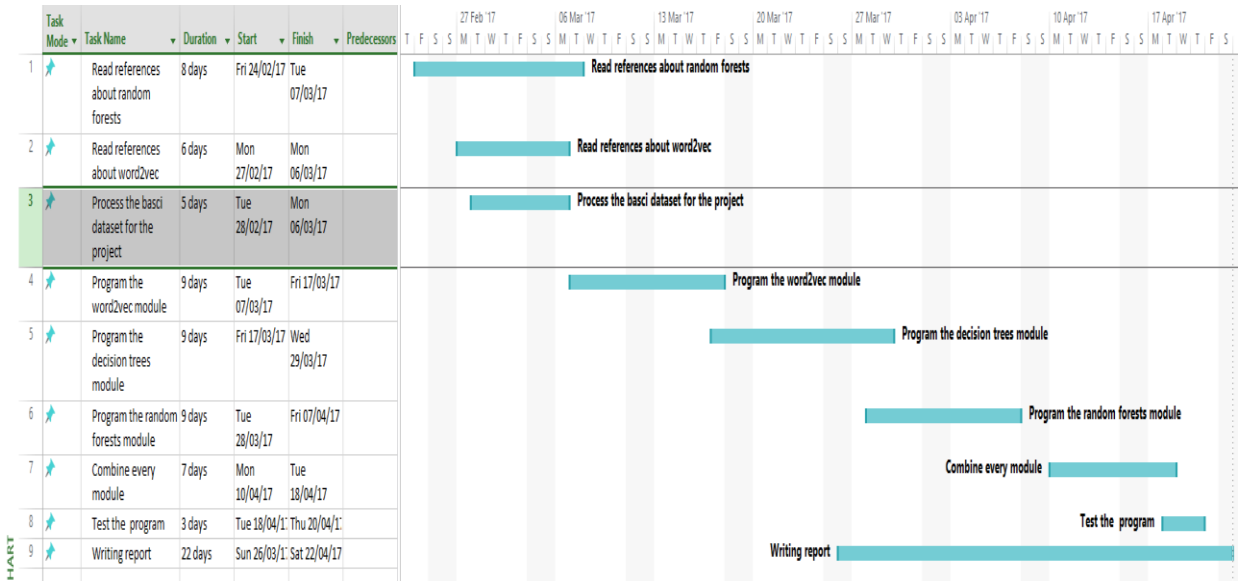
Discussion

As was mentioned before, the dataset also includes 1000 testing questions. Therefore, these testing questions can be used to test the accuracy of the project. They have the same structure with the training questions, so simply input the testing questions into the program, and then compare the results with the correct answers. In addition, there is an assumption that make the project learn to add the test data into the original dataset. But the implementation will still be a tough task.

Conclusion

This project is expected to implement random forests to operate a large dataset from bABi tasks. After inputting testing dataset into the project, the random forests finally print out the correct answer relating to the context of the input sentences. Because of the complication of some tasks, as there are too many features, the accuracy of the output needs to be improve. Furthermore, the project should have been supposed to have the ability to learn through the process of input, but the implementation is complex.

Project Plan



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		Read references about random forests	8 days	Fri 24/02/17	Tue 07/03/17	
2		Read references about word2vec	6 days	Mon 27/02/17	Mon 06/03/17	
3		Process the basci dataset for the project	5 days	Tue 28/02/17	Mon 06/03/17	
4		Program the word2vec module	9 days	Tue 07/03/17	Fri 17/03/17	
5		Program the decision trees module	9 days	Fri 17/03/17	Wed 29/03/17	
6		Program the random forests module	9 days	Tue 28/03/17	Fri 07/04/17	
7		Combine every module	7 days	Mon 10/04/17	Tue 18/04/17	
8		Test the program	3 days	Tue 18/04/17	Thu 20/04/17	
9		Writing report	22 days	Sun 26/03/17	Sat 22/04/17	

References

[1] L.Breiman; A.Cutler. *Random Forests* [Online]. Available:

https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[2] Katsuyuki Tanaka, Takuji Kinkyo, Shigeyuki Hamori, "Random Forests-based Early Warning System for Bank Failures," *Economics Letters* vol. 148, pp. 118-121, Nov. 2016

[3] J.Weston, *et al*, "Towards ai-complete question answering: A set of prerequisite toy tasks." New York, USA, arXiv preprint arXiv: 1502.05698. Dec. 2015