

Assignment 1 Report

Siyamthanda Phakathi

PHKSIY006

Submitted to the University of Cape Town
In accordance to the institutions Academic
Dishonesty Policy



Faculty of Science

CSC 2001F

March 2025

Introduction

Various AI systems use general knowledge to answer questions and use reasoning. These large language models have to be trained and learning from web text may be inaccurate. To better train these models, researchers have curated verified datasets that contain generic statements about the world. One of these datasets is known as GenericsKB, which contains over 3 million verified general statements about the world.

Problem statement

For this assignment I am required to implement a java program to query entries in GenericsKB, add additional entries and update to the knowledge base. These functions will all be performed in memory until the user exits. The dataset contains pre-processed statement with 3 tab-separated fields: the term, the statement and the confidence score (A number between 0 and 1 where 1 is complete confidence in the statement). I am also required to implement two versions of this proof-of-concept program using traditional arrays and a binary search tree.

Object-oriented Design

This section will go through the core functionality required by each version of the program. Additional functionality will be explored later.

The core functionality of this program can be broken down into 3 main operations:

- Reading in the dataset and populating the knowledge base using the specified data structures.
- Allow the user to add new statements and update existing statement if the new statement has a higher confidence score. Both explicitly using the user interface and through the uploading of another file full of entries.
- Display information from the knowledge base. This includes search functionality using both the term and statement.

The first step in this process was to have a common object for which each entry in the dataset is stored. By creating an Entry class I can treat every entry as its own object with the term, statement and confidence score as attributes of that object. This makes for easier manipulation down the line as I can just edit the attributes of the object as opposed to the actual string entry itself.

To handle our first issue of handling reading in a file and populating data I have implemented a KnowledgeBaseArray and KnowledgeBaseBST class. They both do the same thing for their respective data structures. They both read in a Generics file, create Entry objects using the data in the text file, populate their respective data structures and have search functionality to display an entry and its information. Details of each are explored below.

Traditional array implementation

For the KnowledgeBaseArray class we had the following functions and instance variables:

- Entry[] entries : This is the array that will store all Entry objects after they have been created.
- Public KnowledgeBaseArray(String fileName): A constructor that takes in a file name and creates our list of entries using the function addEntries().
- Private Entry[] addEntries(fileName)String file: Reads in filename using buffered reader, creates array with size equal to number of entries using helper function CountLines. After getting the number of entries it loops using a string tokenizer to create a new Entry for each line in the buffered reader and returns an array of all these Entries.

- Private int CountLines(String text): Uses a counter that is incremented while text still has next line.
- Private appendEntries(String newFileName): Allows the addition of new entries by use of another Generics text file. If there are duplicates they are compared and updated accordingly.
- Private findDuplicateIndex: Finds the index of duplicate entries.
- Public void UpdateEntry(String term, String statement, double confidenceScore): Finds the entry we want to update by using term to find the index of the Entry containing the same term. If confidence score is sufficiently high enough we update the statement and confidence score.
- Private Integer FindIndex(String term): Uses a counter to linearly search for the term in entries. If found we return int I which is the entries index. Otherwise we return null so the entry does not exist.
- Public Entry Find(String term): Linear search for entry with its term equal to term we are looking for.
- Public Entry Find(String term, String statement): Linear search for entry with its term and statement equal to those we are looking for. If found returns entry's confidence score.

All these functions are used in the GenericsKbArray class which is the user interface for this program. Through our UI we can complete all our expected functions as per our instruction.

Binary search tree implementation

For the KnowledgeBaseBST we modified BinaryTree.java, BinarySearchTree.java and BinaryTreeNode.java provided to us by the computer science department written by Hussein Suleman. It contains the following functions and instance variables.

- BinarySearchTree<Entry> entries: Our binary search tree containing our Entry objects.
- Public KnowledgeBaseBST(String fileName): Populates our tree using addEntries().
- Private addEntries(String fileName): populates a binary tree with Entry objects using the trees insert function.
- Private appendEntries(String newFileName): Allows the addition of new entries by use of another Generics text file. If there are duplicates they are compared and updated accordingly.
- Public Entry Find(String term): Uses the trees built in find function to search for entry with same term.
- Public Double Find(String term, String statement): returns confidence score if entry with the same term and statement is found.
- Public void UpdateEntry(Entry newEntry): Updates entry if it exists and confidence score is high enough otherwise creates a new entry in our tree.

All these functions are used in the GenericsKbBST class to fulfil our core functionality.

The main purpose of the KnowledgeBase classes is to create an object that stores all of our entries and have these functions so that we can easily access our entries in these data structures.

Creativity

The creation of Main.java allows one to more easily access both user interfaces without running them separately.

```
Choose which data structure you would like to use
```

```
1: Simple Array
```

```
2: Binary Search Tree
```

```
█
```

This is a much better user experience as it allows you to interact with both programs more easily.

Users are reminded that they cannot do anything until they have loaded a knowledge base. Any time the wrong selection is made or an error occurs, the user is notified of what they did wrong.

I also added the ability to do a partial search. The partial search allows you to search for all entries that begin with a specified prefix. This means that a search like “dog” would return entries with the term “dog trainer” and “dog food” if they exist in the knowledge base. This is useful as you can view all entries relating to the specified prefix.

```
Please enter prefix of term you want that start with it: maple
```

```
maple seed Maple seeds spin in the wind on wing-like attachments. 0.728
```

```
maple sugar Maple sugar is food. 1.0
```

```
maple leaf Maple leafs are emblems. 1.0
```

```
maple syrup Maple syrup is syrup. 1.0
```

```
maple cream Maple cream is located in jars. 1.0
```

```
maple Maple is a pale wood with a fine grain. 0.728
```

```
maple sap Maple sap is a colorless liquid taken directly from the maple tree. 0.805
```

```
maple tree Maple trees are deciduous plants. 1.0
```

Testing

For the testing we will go through every option in our Generics Apps and see if my programs function as intended. For the knowledge base I have use GenericsKB.txt and GenericsKB-additional.txt.

Testing array implementation

Firstly we cannot do any other operations until knowledge base is loaded. If the file from which knowledge base will be loaded does not exist an error will occur.

```
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:2
Knowledge base has not been populated. Please load knowledge base.
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:1
Enter file name: he
Experienced an error while loading knowledge base. Please ensure filename is correct.
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:1
Enter file name: GenericsKB.txt
Knowledge base loaded successfully.
```

Next we will try search for an element by term and then try to update it with a new statement with a lower confidence score. It will not update because it is too low.

```
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:3
Enter the term to search: maple
Statement found: Maple is a pale wood with a fine grain. (Confidence score: 0.728)
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:2
Enter the term: maple
Enter the statement: maple is an animal
Enter the confidence score: 0
Did not update, confidence score too low.
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
```

When we update the term with a statement of a higher confidence score it updates, even when we search for it.


```
Enter your choice:2
Enter the term: maple
Enter the statement: maple has 5 letters in it
Enter the confidence score: 1
Statement for maple has been updated
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:3
Enter the term to search: maple
Statement found: maple has 5 letters in it (Confidence score: 1.0)
```

We will try test out the search by term and statement feature.

```
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:4
Enter the term: maple
Enter the statement: maple is an animal with four legs
Entry not found.
```

The entry does not exist therefore it is not found. Now we will search for a statement that actually exists.

```
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:4
Enter the term: summer day
Enter the statement: Summer days are albums.
The statement was found and has a confidence score of 1.0.
```

We will now try the partial search feature.

```
Enter your choice:5
Please enter prefix of term you want that start with it: maple
maple seed Maple seeds spin in the wind on wing-like attachments. 0.728
maple sugar Maple sugar is food. 1.0
maple leaf Maple leafs are emblems. 1.0
maple syrup Maple syrup is syrup. 1.0
maple cream Maple cream is located in jars. 1.0
maple maple has 5 letters in it 1.0
maple sap Maple sap is a colorless liquid taken directly from the maple tree. 0.805
maple tree Maple trees are deciduous plants. 1.0
Choose an action from the menu:
```

If we try to find a term that isn't there this we get an error.

```
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
Enter your choice:5
Please enter prefix of term you want that start with it: vvhv
No entries found starting with "vvhv"
```

We can also add a new knowledge base whilst one is already loaded. If there are duplicates they will be compared and updated accordingly.

```
Enter your choice:1
Enter file name: GenericsKB-additional.txt
Knowledge base has been appended
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
```

If the new knowledge file contains entries with higher confidence scores than the duplicates in the knowledge base they will be replaced. All other functions will operate as usual.

Testing binary search tree implementation

The binary search tree version can do all the tasks shown above. I have repeated them in the same order.

```
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
Enter your choice:4
Knowledge base has not been populated. Please load knowledge base.
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
Enter your choice:1
Enter file name: heb
Experienced an error while loading knowledge base. Please ensure filename is correct.
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
```

We loaded a knowledge base after failing to do anything else before. Now I will update an existing knowledge base and demonstrate the search capabilities.

```
Enter the term to search: maple
Statement found: Maple is a pale wood with a fine grain. (Confidence score: 0.728)
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
Enter your choice:3
Enter the term: maple
Enter the new statement: maple is a name
Enter the new confidence score: 0.2
Entry has not been updated,confidence score is too low.
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
Enter your choice:3
Enter the term: maple
Enter the new statement: maple has just been updated
Enter the new confidence score: 1
Statement for the term maple has been updated.
Choose an action from the menu:
```

I will now search by term and statement

```
Enter your choice:5
Enter the term: maple
Enter the statement: maple has just been updated
The statement was found and has a confidence score of 1.0.
```

We will also test out the partial search feature for BST.


```

Enter your choice:6
Please enter prefix of term you want that start with it.
maple
maple seed Maple seeds spin in the wind on wing-like attachments. 0.728
maple leaf Maple leafs are emblems. 1.0
maple cream Maple cream is located in jars. 1.0
maple maple has just been updated 1.0
maple sap Maple sap is a colorless liquid taken directly from the maple tree. 0.805
maple sugar Maple sugar is food. 1.0
maple syrup Maple syrup is syrup. 1.0
maple tree Maple trees are deciduous plants. 1.0
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit

```

The BST exclusive feature in this project is the ability to add new entries. We will search for a non existent entry, add it and finally search for it again to prove it exists.

```

Enter your choice:4
Enter the term to search: Siya
Entry not found
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
Enter your choice:2
Enter the new term: Siya
Enter the new statement: Siya is the best, trust
Enter the confidence new score: 1
Entry Siya has been added to the knowledge base.
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
Enter your choice:4
Enter the term to search: Siya
Statement found: Siya is the best, trust (Confidence score: 1.0)
Choose an action from the menu:
1. Load a knowledge base from a file

```

We were able to add a new entry to our database, a binary search tree exclusive feature. Finally we will populate our knowledge base with additional entries through means of loading a new file.

```
7. Quit
Enter your choice:1
Enter file name: GenericsKB-additional.txt
Knowledge base has been appended
Choose an action from the menu:
1. Load a knowledge base from a file
2. Add a new statement to the knowledge base
3. Update entry in the knowledge base
4. Search for a statement in the knowledge base by term
5. Search for a statement in the knowledge base by term and sentence
6. Search multiple entries with a partial term
7. Quit
Enter your choice:█
```

The program works perfectly. Now I will test Main.java which allows for easy access to both implementations.

```
Choose which data structure you would like to use
1: Simple Array
2: Binary Search Tree
1
Choose an action from the menu:
1. Load a knowledge base from a file
2. Update entry in the knowledge base
3. Search for a statement in the knowledge base by term
4. Search for a statement in the knowledge base by term and sentence
5. Search multiple entries with a partial term
6. Quit
```

Everything is done in memory so user history does is not tracked once you quit the program.

Git log

0: commit c1b49973ad74932c8b3f2ee388857f4981a1610a

1: Author: Siyamthanda Phakathi <phksiy006@myuct.ac.za>

2: Date: Sun Mar 16 21:41:57 2025 +0000

3:

4: Finalised all the features including the addition of multiple files, error handling, updating through the use of text files and proper javadoc comments. Also created a simple working Makefile.

5:

6: commit 761fce7e78469e317785edfaed7a95f0f1c861e2

7: Author: Siyamthanda Phakathi <phksiy006@myuct.ac.za>

8: Date: Mon Mar 10 21:37:05 2025 +0200

9:

...

35: Author: Siyamthanda Phakathi <phksiy006@myuct.ac.za>

36: Date: Tue Mar 4 22:43:22 2025 +0200

37:

38: Basic classes established

39:

40: commit 18709a8d22b01d2ba5f13d04e3b7cc48995a274b

41: Author: Siyamthanda Phakathi <phksiy006@myuct.ac.za>

42: Date: Tue Mar 4 21:19:43 2025 +0200

43:

44: First commit