Spring 2013

Discovering Pattern Using Automata

Ahmad Marzdasht Yazdankhah
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Yazdankhah, Ahmad Marzdasht, "Discovering Pattern Using Automata" (2013). *Master's Theses*. 4323.
DOI: <https://doi.org/10.31979/etd.gzkw-a9c9>
https://scholarworks.sjsu.edu/etd_theses/4323

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

DISCOVERING PATTERN USING AUTOMATA

A Thesis

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Ahmad M. Yazdankhah

May 2013

© 2013

Ahmad M. Yazdankhah

ALL RIGHTS RESERVED

The Designated Thesis Committee Approves the Thesis Titled
DISCOVERING PATTERN USING AUTOMATA

by

Ahmad M. Yazdankhah

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2013

Dr. Tsau Young Lin	Department of Computer Science
--------------------	--------------------------------

Dr. Soon Tee Teoh	Department of Computer Science
-------------------	--------------------------------

Ronald Mak	Department of Computer Science
------------	--------------------------------

ABSTRACT

DISCOVERING PATTERN USING AUTOMATA

by Ahmad M. Yazdankhah

Most documents written by humans are not just a collection of words, sentences and paragraphs combined at random. It is believed that there is a pattern hidden behind those piles of characters that represents the author's style of writing. In the previous works and in this thesis, we assumed that the aforementioned belief was a true statement and tried to discover and represent the pattern by automata machines. We used the Alergia algorithm to form an automaton from a prefix-tree-accepter. By testing, we verified that the Alergia algorithm was correctly implemented in our software. Our tests showed that we captured only the patterns of the collections of single sentences in a book. Unfortunately, that is not the full content of a book. Therefore, establishing variable chopping units or a less forceful chopping approach would be a promising approach that is beyond the scope of this master thesis.

ACKNOWLEDGEMENTS

First, my warmest thanks to Dr. T. Y. Lin, my advisor, whose great efforts made everything clear in this thesis. His great suggestions about the test cases were brilliant. Second, I would like to thank Dr. S. T. Teoh and R. Mak for accepting to participate in my thesis defense committee and reviewing my thesis documents.

Also, my special thanks to my kind family, Soheila and Milad, my wife and my son, for being so patient to me. Without their support, I couldn't finish my degree. Lastly, additional thanks to Nicole Faby, my editor, who kindly accepted to review and edit my thesis document.

TABLE OF CONTENTS

1. Introduction	1
1.1. Main Questions.....	1
1.2. Previous Works.....	1
1.3. A Big Picture	2
2. Preliminaries	4
2.1. Automata in Brief	4
2.1.1. Introduction	4
2.1.2. Deterministic Finite Automaton (DFA).....	4
2.1.3. Nondeterministic Finite Automaton (NFA).....	6
2.2. Function Words.....	7
2.2.1. Introduction	7
2.2.2. Stop Words	7
2.3. Alergia Algorithm	8
3. New Approaches.....	9
3.1. New Software Implementation	9
3.2. Function Words Categorization.....	10
3.3. Variations of Alergia Algorithm.....	11
3.4. Chopping Units	11
4. Software Implementation.....	12
4.1. Conceptual Design	12
4.1.1. What is the Problem?.....	12
4.1.2. How to Solve the Problem?	12

4.2. Detail Design	14
4.2.1. The Big Picture of the Software Operation.....	14
4.2.2. UML Class Diagram.....	16
4.3. Development Environment	17
5. Experiments	18
5.1. Implementation Verification Tests.....	19
5.1.1. Test Case 01-01: Satisfying the Example in the Alergia Paper.	19
5.1.2. Test Case 01-02: Scramble Test	21
5.1.3. Test Case 01-03: Similar Sentences Test.....	23
5.2. Regular Tests	26
5.2.1. Test Case 02-01: Regular Test.....	27
5.2.2. Test Case 02-02: Regular Test (More Books)	31
5.2.3. Test Case 02-03: Regular Test (100 Books).....	33
5.2.4. Test Case 02-04: Regular Test (Same Author).....	36
5.3. Special Tests	40
5.3.1. Test Case 03-01: Half Book Test.....	40
5.3.2. Test Case 03-02: Half Book Test (More Books)	42
5.3.3. Test Case 03-03: Odd-Even Test	44
5.4. Parameters Effects	48
5.4.1. Test Case 04-01: Sentence versus Paragraph	48
5.4.2. Test Case 04-02: Sentence versus Paragraph (More Books)...	49
5.4.3. Test Case 04-03: DFS versus BFS.....	51
5.4.4. Test Case 04-04: DFS versus BFS (More Books)	52

5.4.5. Test Case 04-05: DFS and Paragraph.....	54
5.4.6. Test Case 04-06: DFS and Paragraph (More Books)	55
6. Conclusion	58
7. Future Direction.....	60
7.1. Function Words Modifications.....	60
7.2. Variable Chopping Units	60
7.3. Adding Intelligence	61
7.4. Software	61
7.4.1. Supporting Other Languages.....	61
7.4.2. Graphical User Interface (GUI)	61
7.4.3. Multiword Support.....	62
7.4.4. Graphical Output for PTA and SDFA	62
Bibliography.....	63
Appendixes.....	65
Appendix A Abbreviations.....	65
Appendix B Function Words	66
Appendix C Stop Words.....	70
Appendix D Test Environment	76
Appendix E List of eBooks	77

LIST OF TABLES

Table 1: The development environment used in this research	17
Table 2: Yu's [3] result for the same books used in test case 02-01.....	30
Table 3: List of abbreviations.....	65
Table 4: List of auxiliary verbs in function words (Code=1)	66
Table 5: List of conjunctions in function words (Code=2)	66
Table 6: List of determiners in function words (Code=3)	67
Table 7: List of prepositions in function words (Code=4).....	67
Table 8: List of pronouns in function words (Code=5)	68
Table 9: List of quantifiers in function words (Code=6).....	69
Table 10: List of stop words.....	70
Table 11: The software specifications in the test environment	76
Table 12: The hardware specifications in the test environment	76
Table 13: The list of eBooks used in this thesis.....	77

LIST OF FIGURES

Fig. 2.1: A deterministic automaton for example 2.1	5
Fig. 2.2: A non-deterministic automaton for example 2.2	6
Fig. 4.1: The block diagram of the software's operation.	15
Fig. 4.2: The UML class diagram of the software.	16
Fig. 5.1: The PTA created by the test case 01-01.	20
Fig. 5.2: The automaton created by the test case 01-01.	20
Fig. 5.3: The output of the test case 01-02	22
Fig. 5.4: Text, coded strings, and PTA of the test case 01-03.	24
Fig. 5.5: The output of the test case 01-03.	25
Fig. 5.6: The first part of the output of the test case 02-01.	28
Fig. 5.7: The second part of the output of the test case 02-01.....	29
Fig. 5.8: The first part of the output of the test case 02-02.	32
Fig. 5.9: The second part of the output of the test case 02-02.....	33
Fig. 5.10: The output of the test case 02-03.	35
Fig. 5.11: The statistical result of the test case 02-03.....	36
Fig. 5.12: The output of the test case 02-04.	38
Fig. 5.13: The statistical result of the test case 02-04.....	39
Fig. 5.14: The output of the test case 03-01	41
Fig. 5.15: The statistical result of the test case 03-01.....	41
Fig. 5.16: The output of the test case 03-02	43
Fig. 5.17: The statistical result of the test case 03-02 for 43 books	44

Fig. 5.18: The output of the test case 03-03	46
Fig. 5.19: The statistical result of the test case 03-03 for 43 books	46
Fig. 5.20: The statistical result of the test case 03-03 for 43 books	47
Fig. 5.21: The output of the test case 04-01	49
Fig. 5.22: The output of the test case 04-02	50
Fig. 5.23: The output of the test case 04-03	52
Fig. 5.24: The output of the test case 04-04	53
Fig. 5.25: The output of the test case 04-05	55
Fig. 5.26: The output of the test case 04-06	56

1. Introduction

1.1. Main Questions

A professional reviewer can recognize a famous writer's style even if his or her name is not listed on the piece. A rough consequence of this observation is: "Every author has his or her own style of writing." But what is the style of a writer? How can we convert the author's style of writing into a format that is understandable by a computer? And how can we recognize the author's style of writing using a computer algorithm? These are the main questions of previous works [1][2][3] and this thesis.

1.2. Previous Works

In 1993, Lin [4] proposed the opposite of randomness as the concept of pattern. In 1999, Lin [5] extended the idea of pattern to numerical data, and in 2005, Baliga and Lin [6] ported the idea to finite automata and used it as a technique for intrusion detection by discovering the patterns in the system-calls. In 2008, Lin and Zhang [1] used automata for analyzing texts. In that research [1], they considered the sequences of the function words in the sentences as the writer's style of writing and tried to find a pattern in them. In 2009, Lu [2] improved the software that was written by Zhang [1] in C. In 2011, Yu [3] re-wrote the software in Java. She made some modifications on the function words and worked on common pattern effect.

This research was continued from the previous research [3]. We made some modifications in the algorithm structure, re-designed and re-wrote the software¹ and also introduced some new parameters that improved the performance and accuracy of the technique. For a detailed explanation of all of the innovations of this thesis, please refer to chapter 3 "New Approaches."

1.3. A Big Picture

In section 1.1 we asked three questions. We assumed that "every author has his or her own style of writing" was a true statement and considered the sequences of the function words² as the author's style of writing. The writer is "repeating" something in his or her writing that human beings can recognize. The repetition of something is equal to existence of a pattern. In previous works [1][2][3] and this research, we used automata machines to find the pattern.

To set up an automata-based pattern discovering system, first we chopped the desired text into units such as sentences or paragraphs. Then we tokenized each unit into words and kept just the function words. Next, we coded the function words based on a categorization system that we employed from natural language-processing research.³ Then we built a prefix-tree acceptor (PTA) from the coded strings. Next, we applied the Alergia algorithm [7] to minimize the PTA to a stochastic deterministic finite automaton (SDFA).

¹ A copy of the software can be downloaded from Dr. Lin's website [14].

² For definition of "function word," please refer to 2.2.

³ For the detail of the functions words' categorization, please refer to 3.2.

Eventually, we converted a text written by an unknown author to the coded string array and fed it to the SDFA to derive the percentage of acceptance that represented the percentage of similarity between the unknown writer's style and the known one.

The structure of this report is as follows: in chapter 2, we discuss the concept of the automata and function words. Chapter 3 deals with the new approaches we employed in this thesis and chapter 4 describes how the software is developed. Chapter 5 consists of the test cases and the results whereas chapter 6 summarizes the conclusions. Finally, chapter 7 has some suggestions for future researchers who are interested in this topic. Five appendices are added to this report as follow: Appendix A lists all abbreviations that are used in this report, Appendix B lists all function words, Appendix C lists all stop words⁴, Appendix D shows the test environment, and Appendix E is the list of all eBooks used in the test cases.

⁴ For the definition of "stop word," please refer to 2.2.2.

2. Preliminaries

2.1. Automata in Brief

2.1.1. Introduction

The theory of computation includes automata theory, formal languages, grammars, computability, and complexity. The automata theory is one of the building blocks of this research. Therefore, the main ideas of the theory are explained in this section.

2.1.2. Deterministic Finite Automaton (DFA)⁵

A deterministic finite automaton (or acceptor) is defined by the following quintuple: $(Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set of internal states,
- Σ is a finite set of symbols called the input alphabet,
- $\delta: Q \times \Sigma \rightarrow Q$ is a total function called the transition function,
- $q_0 \in Q$ is the initial state,
- F (a subset of Q) is a set of final (accepting) states.

⁵ For more information, please refer to the book written by Peter Linz[12].

Notes:

1. The automaton reads one input symbol at a time and transits to the next step according to the transition function δ .
2. At any time, we can be in one and only one state.
3. When the automaton reaches the end of the input string, if the last state belongs to F , then the input string is considered as accepted string.

Example 2.1

The graph shown in the Fig. 2.1 represents the DFA identified by:

$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$$

where δ is given by:

$$\delta(q_0, 0) = q_0$$

$$\delta(q_1, 0) = q_0$$

$$\delta(q_2, 0) = q_2$$

$$\delta(q_0, 1) = q_1$$

$$\delta(q_1, 1) = q_2$$

$$\delta(q_2, 1) = q_1$$

Fig. 2.1: A deterministic automaton for example 2.1

2.1.3. Nondeterministic Finite Automaton (NFA)⁶

Non-determinism means a choice of moves for an automaton. Rather than prescribing a unique move in each situation, we allow a set of possible moves. Formally, we achieve this by defining the transition function so that its range is a set of possible states.

A non-deterministic finite automaton (or acceptor) is defined by the quintuple $(Q, \Sigma, \delta, q_0, F)$, where Q, Σ, q_0 , and F are defined as for deterministic finite accepters, but

$$\delta: Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q.$$

Example 2.2

Consider the transition graph in the Fig. 2.2. It describes a non-deterministic accepter because there are two transitions labeled 'a' out of q_0 .

Fig. 2.2: A non-deterministic automaton for example 2.2

⁶ For more information, please refer to the book written by Peter Linz [12].

2.2. Function Words

2.2.1. Introduction

"Function words (or grammatical words) are words that have little lexical meaning or have ambiguous meaning, but instead serve to express grammatical relationships with other words within a sentence, or specify the attitude or mood of the speaker. They signal the structural relationships that words have to one another and are the glue that holds sentences together. Thus, they serve as important elements to the structures of sentences." [8] Words that are not function words are called content words (aka, open class words or lexical words) including nouns, verbs, adjectives, and most adverbs, although some adverbs are function words. Dictionaries define the specific meanings of content words, but they can only describe the general usages of function words. By contrast, grammar describes the use of function words in detail, but treats content words in general terms only.

2.2.2. Stop Words

A stop word is a commonly used word such as "the" that most search engines ignore when indexing entries or when retrieving them as the result of a search query. The list of the stop words can be found in Appendix C.

2.3. Alergia Algorithm

In this thesis, the Alergia algorithm [7] is used to build a stochastic deterministic finite automaton (SDFA) from a prefix tree acceptor (PTA). In other words, from a finite number of strings, we create a tree and then we use the Alergia algorithm to approximate it with an SDFA. We have not explained this algorithm in detail because the previous research document [3] explained it well.

The only point we need to know is the role of the parameter α in the algorithm. The Alergia algorithm uses Hoeffding bound inequality [9] to compare the two nodes compatibility. The parameter α ranges from 0.0 to 2.0. The larger the α is chosen, the less merges happen. Therefore, if we choose $\alpha = 2.0$, then no merges happen at all and we get the PTA itself.

3. New Approaches

This chapter explains in detail the new achievements that we made in this thesis.

3.1. New Software Implementation

We designed and implemented the software⁷ from scratch. The reasons why we have not used previously implemented software are

1. The software was not designed based on object-oriented principles.
Therefore, if we wanted to add a new feature, then we had to change the entire of the code.
2. A bug was found during the analyzing of the software that caused inaccuracy in the tests' results. The software's tokenizer could not distinguish between the full stop at the end of a sentence and the dot for the abbreviations like "Dr." or "Mr."
3. It could not distinguish the end of the sentence when there was a quotation mark at the end of the sentence.
4. Function words were hard coded.

We used Java as the underlying programming language, object-oriented as the design technique, and agile methodology as the software life cycle implementation.⁸

⁷ A copy of the software can be downloaded from Dr. Lin's website [14].

⁸ For more information about the software, please refer to chapter 4.

3.2. Function Words Categorization

Function words and their classification have a great impact on the tests' results. In this section we explain the classification we used in this thesis. In contrast with the implementation in the previous research [3], in which the function words were hard coded in the program, we designed the software⁹ in such a way that the tester can easily plug-in a new set of function words and he or she can change the classification easily to experiment with the new set of function words or its classifications.

In the previous implementations [3], the function words were classified in five categories as:

1. Adverb
2. Auxiliary Verb
3. Preposition and Conjunction
4. Pronoun
5. Number

In this research, we used the following categorizations: [10]

1. Auxiliary verb
2. Conjunction
3. Determiner
4. Preposition
5. Pronoun

⁹ A copy of the software can be downloaded from Dr. Lin's website [14].

6. Quantifier

We eliminated the "number" category because our investigations showed that it does not impact the author's style. In fact, it does not make sense that the natural numbers like "one" or "two" can show the mood of an author. The author uses the natural numbers whenever he or she needs them to express something appropriately. Instead, we added the categories of "determiner" and "quantifier," and separated preposition and conjunction. Experiments showed that this new categorization improved the tests' results.¹⁰ We also assigned code "0" for empty string.

3.3. Variations of Alergia Algorithm

The Alergia algorithm [7] uses Breadth-First-Search (BFS) in the process of the merging of the nodes. In this research, we used Depth-First Search (DFS) as well.

3.4. Chopping Units

The previous authors [1][2][3] based their experiments on the "sentence" as the basic unit of chopping. In this research, we added "paragraph" as an additional chopping unit.

¹⁰ For more information about the test results, please refer to chapter 5.

4. Software Implementation¹¹

4.1. Conceptual Design

4.1.1. What is the Problem?

In section 1.1 we asked three questions and in section 1.3 we assumed that "every author has his or her own style of writing" is a true statement. We also considered the sequences of function words¹² as the author's style of writing. In the third question we asked "how can we recognize the author's style of writing by computer algorithm?" The previous works [1][2][3] and this research were focused on this question.

4.1.2. How to Solve the Problem?

We can consider the author's style of writing as a formal language¹³, thus we can use an automaton machine to represent it. We need software to chop the whole book into smaller strings and to create a prefix tree acceptor (PTA) from the set of strings. Then we need to use an algorithm to convert the PTA into an automaton machine.

We break the software operation into two distinct phases:

1. Learning phase
2. Testing phase

¹¹ A copy of the software can be downloaded from Dr. Lin's website [14].

¹² For the definition of "function word," please refer to 2.2.

¹³ For the definition of "formal language" in computer science, please refer to the book written by Peter Linz [12].

In the learning phase, we need the software to be able to get a known book as text file and chop it into a given unit, like sentence or paragraph, to create a set of strings called set S. Then we need to create an automaton from the finite set of strings S. We should be able to save this automaton for later use.

In the testing phase, we should be able to feed another set of strings created from another book or any other document, to the saved automaton and the automaton should be able to give us the acceptance percentage.

The following parameters can be set in the software:

1. Function words
2. Function words' classification
3. Chopping unit for the learning books (sentence or paragraph)
4. Chopping unit for the testing books (sentence or paragraph)
5. The set of learning books
6. The set of testing books
7. The parameter alpha's range

4.2. Detail Design

4.2.1. The Big Picture of the Software Operation

As shown in the Fig. 4.1, the whole program operations can be divided into two distinct phases:

1. Learning phase
2. Testing phase

In the learning phase, the learning document is chopped into the desired units (sentences or paragraphs). Then the units are tokenized and the function words are coded in one of the categories we mentioned in section 3.2. At this point, we have a set of strings and each string contains a chain of numeric codes. We can consider this set of strings as language A. Then, a PTA is created using the set of strings or language A. By applying the Alergia algorithm [7], the PTA is converted into an automaton and it can be written into a file for later use.

In the testing phase, the testing document is chopped into the desired units and the same tasks of the learning phase are done to create a set of coded strings that is called language B. Next, we feed this set of strings to the automaton we created before. The automaton that is created to represent language A can provide the percentage of acceptance of the language B. The bigger the percentage is, the more similarity between language A and language B. In other words, in this case, the language B belongs to the author of language A.

Fig. 4.1: The block diagram of the software's operation.

4.2.2. UML Class Diagram

Fig. 4.2 shows the unified modeling language (UML) diagram of the software. The starter class of the program is PRGStarter class. Currently, the program is just for Java developers' use.

Fig. 4.2: The UML class diagram of the software.

4.3. Development Environment

Table 1 summarized the development environment used in this research.

Table 1: The development environment used in this research

Development Environment	
Language	Java 1.7
IDE	NetBeans 7.1.2
UML Tools	StarUML 5.0.2
Design Method	Object-Oriented Design (OOD)
Development Methodology	Agile

5. Experiments

In this chapter, we explain all of the test cases we conducted in detail. All the test cases are divided into four sets as follow:

1. Test case set 01: contains three test cases to verify the reliability of the software's implementation.
2. Test case set 02: contains four test cases to verify the quality of the technique we employed in this research.
3. Test case set 03: contains three special test cases to support our conclusions in the test case set 02.
4. Test case set 04: contains six test cases to test the effect of different parameters.

For the test environment, please refer to Appendix D and for the list of eBooks used in this thesis, please refer to Appendix E. As mentioned earlier, at this moment, the software can be used only by Java developers.

Before explaining the test cases in detail, we need to know how the program shows a node in a PTA or in an automaton. Nodes in this program are shown as the following format:

$$\langle - \{ x - y [z : w] \} - \rangle$$

Where:

x : the node ID

y : the symbol that led to this node

z : the number of strings reached to this node

w : the number of accepted or terminated strings in this node

$->$: this sign shows that there is at least one child for this node. Also note that the children are lined up vertically.

$<-$: this sign shows that this node goes to one of the previous nodes.

Leaf: shows that this node is a leaf and there is no child.

5.1. Implementation Verification Tests

5.1.1. Test Case 01-01: Satisfying the Example in the Alergia Paper

5.1.1.1. Test Case Objective

This test case is designed to verify if the result of the example in the Alergia paper [7] is satisfied by our implementation. Please note that in this particular test case, "-" is equal to empty string.

5.1.1.2. Test Condition

Starting Alpha	0.7
Increment	-
Upper Limit	0.7
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Coding Type	-
Learning Strings	{ "1 1 0", "-", "-", "-", "0", "-", "0 0", "0 0", "-", "-", "-",

	"1 0 1 1 0", "-", "-", "1 0 0"
Testing Strings	The same as the learning string

5.1.1.3. Test Output

Fig. 5.1 and Fig. 5.2 show the PTA and the final SDFA respectively.

```
Total # of nodes = 11
Number of merges = 0

{0-0[15:9]}->
    {4-0[3:1]}->
        {5-0[2:2]}-> Leaf
    {1-1[3:0]}->
        {6-0[2:0]}->
            {10-0[1:1]}-> Leaf
            {7-1[1:0]}->
                {8-1[1:0]}->
                    {9-0[1:1]}-> Leaf
        {2-1[1:0]}->
            {3-0[1:1]}-> Leaf
```

Fig. 5.1: The PTA created by the test case 01-01.

```
Total # of nodes = 11
Number of merges = 9

{0-0[25:15]}->
    <-{0-0[25:15]}
    {1-1[6:0]}->
        <-{0-0[25:15]}
        <-{1-1[6:0]}
```

Fig. 5.2: The automaton created by the test case 01-01.

5.1.1.4. Test Case Conclusion

The outputs of the test are exactly the same as the result of the Alergia paper [7]. Therefore, the test is passed.

5.1.2. Test Case 01-02: Scramble Test

5.1.2.1. Test Case Objective

In this test case, we create an SDFA with a book, and then we scramble the sentences of the book and create the test document from the scrambled sentences and feed it to the SDFA. If the implementation is precise and the Alergia algorithm [7] works fine, then we should get 100% acceptance. To be sure, we choose seven books randomly and repeat this test for all of them.

5.1.2.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.0
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Coding Type	Function Words
Learning Books	Agatha Christie - The Secret Adversary.txt Arthur Conan Doyle -The Adventures of Sherlock Holmes.txt Canaan Doyle - A Study in Scarlet.txt Daniel Defoe - Robinson Crusoe.txt Edgar Rice Burroughs - A Princess of Mars.txt Elliott Whitney - The Pirate Shark.txt Frank Baum - The Wonderful Wizard of Oz.txt
Testing Books	The same as learning books but the sentences are scrambled.

5.1.2.3. Test Output

```
10/29/2012@21:03 ***** Pattern Discovery Started! *****

10/29/2012@21:03 loading function words file .\FunctionWords\functionWords.txt
10/29/2012@21:03 test engine started.

TEST CASE# 02: Scramble Test
Results of learning with one doc and testing with the scrambled of it

10/29/2012@21:04 processed alpha = 0.1    elapsed:  5 Sec, 484 ms,
10/29/2012@21:04 processed alpha = 0.2    elapsed:  6 Sec, 453 ms,
10/29/2012@21:04 processed alpha = 0.3    elapsed:  6 Sec, 703 ms,
10/29/2012@21:04 processed alpha = 0.4    elapsed:  7 Sec, 735 ms,
10/29/2012@21:04 processed alpha = 0.5    elapsed:  7 Sec, 218 ms,
10/29/2012@21:04 processed alpha = 0.6    elapsed:  7 Sec, 766 ms,
10/29/2012@21:04 processed alpha = 0.7    elapsed:  8 Sec, 813 ms,
10/29/2012@21:04 processed alpha = 0.8    elapsed: 10 Sec, 156 ms,
10/29/2012@21:05 processed alpha = 0.9    elapsed: 13 Sec, 906 ms,
10/29/2012@21:05 processed alpha = 1.0    elapsed: 14 Sec, 828 ms,
10/29/2012@21:05 all alphas are processed. Elapsed:  1 Min, 29 Sec, 62 ms,

Coding: FUNCTION_WORD
Learning Chopping: Sentence
Testing Chopping: Sentence
Merge Algorithm: Alergia BFS

Learning Doc: Each book is learned and is tested against its scrambled version.
Test Folder: .\LearnBooks\

Testing Doc01: Agatha Christie - The Secret Adversary.txt
Testing Doc02: Arthur Conan Doyle -The Adventures of Sherlock Holmes.txt
Testing Doc03: Canaan Doyle - A Study in Scarlet.txt
Testing Doc04: Daniel Defoe - Robinson Crusoe.txt
Testing Doc05: Edgar Rice Burroughs - A Princess of Mars.txt
Testing Doc06: Elliott Whitney - The Pirate Shark.txt
Testing Doc07: Frank Baum - The Wonderful Wizard of Oz.txt

-----
i  Alpha Doc01(%) Doc02(%) Doc03(%) Doc04(%) Doc05(%) Doc06(%) Doc07(%)
--  -----
1  0.10  100.000  100.000  100.000  100.000  100.000  100.000  100.000
2  0.20  100.000  100.000  100.000  100.000  100.000  100.000  100.000
3  0.30  100.000  100.000  100.000  100.000  100.000  100.000  100.000
4  0.40  100.000  100.000  100.000  100.000  100.000  100.000  100.000
5  0.50  100.000  100.000  100.000  100.000  100.000  100.000  100.000
6  0.60  100.000  100.000  100.000  100.000  100.000  100.000  100.000
7  0.70  100.000  100.000  100.000  100.000  100.000  100.000  100.000
8  0.80  100.000  100.000  100.000  100.000  100.000  100.000  100.000
9  0.90  100.000  100.000  100.000  100.000  100.000  100.000  100.000
10 1.00  100.000  100.000  100.000  100.000  100.000  100.000  100.000
-----
```

Fig. 5.3: The output of the test case 01-02

5.1.2.4. Test Case Conclusion

As shown in Fig. 5.3, all SDFA's accepted the whole scrambled documents. It means that the order of the sentences does not matter. Therefore, this test is passed.

5.1.3. Test Case 01-03: Similar Sentences Test

5.1.3.1. Test Case Objective

In this test case, we copy a sentence ten times and create the PTA. The PTA should contain one branch and the number of received symbols in each node should be ten. The number of accepted symbols should be zero in all nodes but ten in the leaf node.

5.1.3.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.0
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Coding Type	Function Words
Learning Books	A manually created text file by copying a sentence 10 times.
Testing Books	A manually created text file by copying the same sentence 20 times.

We repeated this test by changing the chopping unit to paragraph over different values of alpha. In each case, the result was 100% acceptance. Then we changed the merge algorithm to DFS and we received the same result. These results proved that the implementation was accurate and reliable.

5.1.3.3. Test Output

```
10/29/2012@23:35 ***** Pattern Discovery Started! *****

Class: Control.TestEngine
Position: 100
Learn file text:

Class: Control.TestEngine
Position: 102
Test Document Array:

[File Name = .\SimilarSentence\learn.txt
[[1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7,
1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7]]
,
File Name = .\SimilarSentence\test.txt
[[1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7,
1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1
3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7, 1 3 6 4 7]]]

Class: Control.TestEngine
Position: 200
PTA of learning doc:

***** PTA *****
Total # of nodes = 6

{0-0[10:0]}->
    {1-1[10:0]}->
        {2-3[10:0]}->
            {3-6[10:0]}->
                {4-4[10:0]}->
                    {5-7[10:10]}-> Leaf
```

Fig. 5.4: Text, coded strings, and PTA of the test case 01-03.

```

Learning Chopping: Sentence
Testing Chopping: Sentence
Merge Algorithm: Alergia BFS

Learning Doc: .\SimilarSentence\learn.txt
Test Folder: .\SimilarSentence\

Testing Doc01: learn.txt
Testing Doc02: test.txt

-----
i   Alpha Doc01(%) Doc02(%)
--   -
1   0.10  100.000  100.000
2   0.20  100.000  100.000
3   0.30  100.000  100.000
4   0.40  100.000  100.000
5   0.50  100.000  100.000
6   0.60  100.000  100.000
7   0.70  100.000  100.000
8   0.80  100.000  100.000
9   0.90  100.000  100.000
10  1.00  100.000  100.000
-----

```

Fig. 5.5: The output of the test case 01-03.

5.1.3.4. Test Case Conclusion

The output was what we expected. Therefore, the test is passed.

5.2. Regular Tests

We designed four tests cases to verify the reliability of the automata technique using the Alergia algorithm [7]. In fact, we ran hundreds of tests to verify this technique, but we pruned them down and mention only the most important ones in this and the following sections. The general idea behind these test cases was: in the learning phase, we chose at random a book from a known author and called it Doc01. We then created an appropriate SDFA for it. In the testing phase, we used three books as follows:

1. The same book as the learning phase
2. Another book from the same author
3. A book from a different author

We called these three books Doc01, Doc02, and Doc03, respectively. We tested the SDFA against the same book because we needed to make sure that the SDFA was created successfully. We called this process the "Self Test." The SDFA should give the same book 100% acceptance score. In the next step, we investigated how the extent to which the SDFA could recognize the same author's style by testing Doc02; the closer this percentage is to 100%, the better. Ultimately, we tested the SDFA against the Doc03 that was written by the other author to find how much it could distinguish the other author's style; the closer this percentage is to 0%, the better. Consequently, if the acceptance difference was big enough for the most test cases, then we could rely on this technique.

These tests are the most important tests in this research and, as mentioned earlier, we ran those hundreds of times for different authors and with different configurations. That is why we call it the "regular test."

5.2.1. Test Case 02-01: Regular Test

5.2.1.1. Test Case Objective

In this test case, we have chosen the same books that were used by Yu.
[3] We compared her results with ours to show that we improved the acceptance difference.

5.2.1.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	2.0
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Learning Books	HP5.txt
Testing Books	HP5.txt
	HP3.txt
	One Hundred Years of Solitude.txt

5.2.1.3. Test Output

```
10/30/2012@22:18 ***** Pattern Discovery Started! *****
10/30/2012@22:18 loading function words file .\FunctionWords\functionWords.txt
10/30/2012@22:18 test engine started.

TEST CASE# 02-01 - REGULAR TEST
Learning a book from an author; Testing with the same book,
a book from the same author and a book from another author.

10/30/2012@22:18 processed alpha = 0.1, elapsed: 2 sec, 187 ms,
10/30/2012@22:18 processed alpha = 0.2, elapsed: 2 sec, 828 ms,
10/30/2012@22:18 processed alpha = 0.3, elapsed: 3 sec, 219 ms,
10/30/2012@22:18 processed alpha = 0.4, elapsed: 3 sec, 641 ms,
10/30/2012@22:18 processed alpha = 0.5, elapsed: 3 sec, 609 ms,
10/30/2012@22:18 processed alpha = 0.6, elapsed: 4 sec, 578 ms,
10/30/2012@22:18 processed alpha = 0.7, elapsed: 5 sec, 985 ms,
10/30/2012@22:18 processed alpha = 0.8, elapsed: 8 sec, 15 ms,
10/30/2012@22:18 processed alpha = 0.9, elapsed: 12 sec, 157 ms,
10/30/2012@22:19 processed alpha = 1.0, elapsed: 13 sec, 515 ms,
10/30/2012@22:19 processed alpha = 1.1, elapsed: 23 sec, 703 ms,
10/30/2012@22:20 processed alpha = 1.2, elapsed: 25 sec, 328 ms,
10/30/2012@22:27 processed alpha = 1.3, elapsed: 7 min, 19 sec, 485 ms,
10/30/2012@22:35 processed alpha = 1.4, elapsed: 7 min, 47 sec, 829 ms,
10/30/2012@22:43 processed alpha = 1.5, elapsed: 8 min, 33 sec, 46 ms,
10/30/2012@22:55 processed alpha = 1.6, elapsed: 11 min, 28 sec, 532 ms,
10/30/2012@23:09 processed alpha = 1.7, elapsed: 14 min, 21 sec, 31 ms,
10/30/2012@23:24 processed alpha = 1.8, elapsed: 14 min, 59 sec, 437 ms,
10/30/2012@23:39 processed alpha = 1.9, elapsed: 15 min, 21 sec, 907 ms,
10/30/2012@23:58 processed alpha = 2.0, elapsed: 18 min, 28 sec, 703 ms,
10/30/2012@23:58 All alphas are processed.
Elapsed: 1 hour, 40 min, 9 sec, 563 ms
```

Fig. 5.6: The first part of the output of the test case 02-01.

```
Coding: FUNCTION_WORD
Learning Chopping: Sentence
Testing Chopping: Sentence
Merge Algorithm: Alergia BFS

Learning Doc: .\TestCaseSet02\testCase01\HP5.txt
Test Folder : .\TestCaseSet02\testCase01\

Testing Doc01: 0 HP5.txt
Testing Doc02: 1 HP3.txt
Testing Doc03: 2 One Hundred Years of Solitude.txt
```

				SENTENCE BFS	
i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Diff.
1	0.10	100.000	100.000	99.874	0.126
2	0.20	100.000	99.540	98.592	0.948
3	0.30	100.000	98.233	91.423	6.810

4	0.40	100.000	98.458	89.130	9.328
5	0.50	100.000	97.259	79.686	17.573
6	0.60	100.000	94.024	68.220	25.804
7	0.70	100.000	91.401	62.116	29.285
8	0.80	100.000	87.417	52.257	35.160
9	0.90	100.000	82.138	42.651	39.487
10	1.00	100.000	79.525	38.967	40.558
11	1.10	100.000	74.009	33.171	40.838
12	1.20	100.000	71.953	31.365	40.588
13	1.30	100.000	63.129	23.673	39.456
14	1.40	100.000	62.797	23.601	39.196
15	1.50	100.000	62.433	23.167	39.266
16	1.60	100.000	61.491	22.589	38.902
17	1.70	100.000	60.805	22.553	38.252
18	1.80	100.000	60.463	22.030	38.433
19	1.90	100.000	60.270	21.885	38.385
20	2.00	100.000	60.270	21.885	38.385

Fig. 5.7: The second part of the output of the test case 02-01.

5.2.1.4. Important Observations about the Above Results

1. The testing process takes much more time after $\alpha=1.2$. It abruptly jumps from 25 seconds to 7 minutes, almost 16 times greater duration.
2. Before $\alpha = 0.5$, the differences are quite small.
3. After $\alpha = 1.3$, the acceptance percentages and the differences almost remain unchanged.
4. The best result is for $\alpha = 1.1$.

Since the above observations, except number 4, were repeated for all of our experiments, thus we concluded that the best values for α happened between 0.5 and 1.3. This conclusion is also consistent with the Alergia algorithm paper [7].

Therefore, for simplicity and without losing the generality of the research results, we followed this direction for all of our test cases.

5.2.1.5. Comparison with the Previous Research

Table 2: Yu's [3] result for the same books used in test case 02-01

alpha setting	Harry Potter book 5 (<i>self test</i>)	Harry Potter book 3	One Hundred Years Of Solitude	Difference
0.1	100	100	100	0
0.2	100	99.99	99.96	0.03
0.3	100	99.98	99.68	0.3
0.4	100	99.89	99.56	0.33
0.5	100	99.72	98.95	0.77
0.6	100	99.95	98.4	1.55
0.7	100	99.19	94.45	4.74
0.8	100	97.95	88.73	9.22
0.9	100	96.35	81.56	14.79
1.0	100	94.29	75.23	19.06
1.1	100	87.25	52.68	34.57
1.2	100	84.1	47.61	36.49
1.3	100	72.51	28.74	43.77
1.4	100	72.1	28.07	44.03
1.5	100	70.95	27.19	43.76
1.6	100	70.65	26.68	43.97
1.7	100	69.77	25.59	44.18
1.8	100	69.69	25.47	44.22
1.9	100	69.67	25.43	44.24
2.0	100	69.67	25.43	44.24
FPTA	100	69.67	25.43	44.24

The results in the Table 2 show that the acceptance differences before $\alpha = 1.0$ were not significantly different from the results of the Alergia algorithm paper [7]. Also, after $\alpha = 1.3$, she gained almost completely

different results from ours. For the reasons why she gained different results, please refer to section 3.1.

5.2.2. Test Case 02-02: Regular Test (More Books)

5.2.2.1. Test Case Objective

This is the same test as test case 02-01 with more books.

5.2.2.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	2.0
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Learning Books	Jane Austen - Emma.txt
Testing Books	Jane Austen - Emma.txt
	Jane Austen - Mansfield Park.txt
	Charles Dickens - Oliver Twist.txt

5.2.2.3. Test Output

```
11/01/2012@07:45 ***** Pattern Discovery Started! *****
11/01/2012@07:45 loading function words file .\FunctionWords\functionWords.txt
11/01/2012@07:45 test engine started.

TEST CASE# 02-02 - REGULAR TEST
The same as 02-01 but different books

11/01/2012@07:45 processed alpha = 0.1, elapsed: 2 sec, 94 ms,
11/01/2012@07:45 processed alpha = 0.2, elapsed: 2 sec, 765 ms,
11/01/2012@07:45 processed alpha = 0.3, elapsed: 3 sec, 360 ms,
11/01/2012@07:45 processed alpha = 0.4, elapsed: 3 sec, 531 ms,
11/01/2012@07:45 processed alpha = 0.5, elapsed: 3 sec, 125 ms,
11/01/2012@07:45 processed alpha = 0.6, elapsed: 3 sec, 437 ms,
11/01/2012@07:45 processed alpha = 0.7, elapsed: 4 sec, 469 ms,
11/01/2012@07:45 processed alpha = 0.8, elapsed: 5 sec, 172 ms,
11/01/2012@07:45 processed alpha = 0.9, elapsed: 7 sec, 843 ms,
11/01/2012@07:46 processed alpha = 1.0, elapsed: 8 sec, 469 ms,
11/01/2012@07:46 processed alpha = 1.1, elapsed: 12 sec, 625 ms,
11/01/2012@07:46 processed alpha = 1.2, elapsed: 14 sec, 828 ms,
11/01/2012@07:54 processed alpha = 1.3, elapsed: 8 min, 21 sec, 875 ms,
11/01/2012@08:04 processed alpha = 1.4, elapsed: 9 min, 5 sec, 953 ms,
11/01/2012@08:13 processed alpha = 1.5, elapsed: 9 min, 47 sec, 719 ms,
11/01/2012@08:28 processed alpha = 1.6, elapsed: 14 min, 40 sec, 937 ms,
11/01/2012@08:47 processed alpha = 1.7, elapsed: 18 min, 44 sec, 937 ms,
11/01/2012@09:06 processed alpha = 1.8, elapsed: 19 min, 3 sec, 907 ms,
11/01/2012@09:23 processed alpha = 1.9, elapsed: 17 min, 31 sec, 78 ms,
11/01/2012@09:47 processed alpha = 2.0, elapsed: 23 min, 52 sec, 422 ms,
11/01/2012@09:47 All alphas are processed.
Elapsed: 2 hour, 2 min, 21 sec
```

Fig. 5.8: The first part of the output of the test case 02-02.

```
Coding: FUNCTION_WORD
Learning Chopping: Sentence
Testing Chopping: Sentence
Merge Algorithm: Alergia BFS

Learning Doc: .\TestCaseSet02\testCase02\1 Jane Austen - Emma.txt
Test Folder : .\TestCaseSet02\testCase02\

Testing Doc01: 1 Jane Austen - Emma.txt
Testing Doc02: 2 Jane Austen - Mansfield Park.txt
Testing Doc03: 3 Charles Dickens - Oliver Twist.txt

-----
i  Alpha Doc01(%) Doc02(%) Doc03(%) Diff.
--  -
1  0.10 100.000 99.987 99.990 -0.003
2  0.20 100.000 96.309 98.499 -2.190
3  0.30 100.000 95.925 97.140 -1.215
4  0.40 100.000 91.838 95.763 -3.925
5  0.50 100.000 86.729 92.704 -5.975
```

6	0.60	100.000	71.044	81.864	-10.820
7	0.70	100.000	71.440	82.966	-11.526
8	0.80	100.000	61.030	74.834	-13.804
9	0.90	100.000	53.136	67.566	-14.430
10	1.00	100.000	51.105	65.770	-14.665
11	1.10	100.000	44.412	60.365	-15.953
12	1.20	100.000	42.675	58.598	-15.923
13	1.30	100.000	35.560	52.423	-16.863
14	1.40	100.000	34.985	52.233	-17.248
15	1.50	100.000	34.615	51.872	-17.257
16	1.60	100.000	33.938	51.302	-17.364
17	1.70	100.000	33.427	51.017	-17.590
18	1.80	100.000	33.261	50.836	-17.575
19	1.90	100.000	33.261	50.817	-17.556
20	2.00	100.000	33.261	50.817	-17.556

Fig. 5.9: The second part of the output of the test case 02-02.

5.2.2.4. Test Case Conclusion

The acceptance of a different author's book was better than the same author's book! We verified this result by other test cases that we designed in the following sections.

5.2.3. Test Case 02-03: Regular Test (100 Books)

5.2.3.1. Test Case Objective

This test case is the same as test 02-02 but run the regular test with 100 books. In other words, we learned with one book and tested with 100 books from other authors. The goals of this test case are to verify statistically how many books are accepted bellows 50% and what would be the best alpha to get this result.

5.2.3.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Learning Books	Walt Whitman - Leaves of Grass.txt
Testing Books	The list of 100 books in Appendix E

5.2.3.3. Test Output

Alpha	0.40	0.50	0.60	0.70	0.80	0.90	1.00	1.10	1.20
Doc01 (%)	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000
Doc02 (%)	68.899	54.511	40.873	27.338	18.124	12.454	11.026	7.925	6.488
Doc03 (%)	96.356	94.013	89.275	82.775	76.753	69.535	66.986	59.100	57.441
Doc04 (%)	97.184	95.174	92.685	87.310	82.239	75.041	72.575	65.050	62.456
Doc05 (%)	89.423	85.206	76.731	66.494	57.905	51.039	47.827	41.463	39.301
Doc06 (%)	91.053	87.437	79.354	70.191	61.924	54.529	51.628	45.032	43.466
Doc07 (%)	92.016	87.300	78.861	68.974	60.632	52.110	50.000	43.423	40.982
Doc08 (%)	90.190	85.001	74.486	63.562	53.027	43.405	39.917	33.386	31.836
Doc09 (%)	91.681	86.832	77.522	65.487	55.044	45.770	43.434	35.681	33.876
Doc10 (%)	89.527	84.414	77.408	67.905	60.734	53.491	51.148	45.314	43.444
Doc11 (%)	89.676	84.382	77.041	67.793	60.847	53.778	51.419	45.879	44.702
Doc12 (%)	88.625	82.732	74.219	63.536	55.734	48.475	45.800	40.159	38.606
Doc13 (%)	91.070	86.424	80.439	73.200	66.606	60.783	58.645	52.964	51.283
Doc14 (%)	88.535	83.870	75.123	65.013	56.099	48.521	46.322	40.386	38.865
Doc15 (%)	48.548	39.059	29.734	23.108	17.546	13.906	13.415	11.084	10.389
Doc16 (%)	84.529	76.906	66.018	52.202	40.867	31.411	29.086	23.184	20.903
Doc17 (%)	76.881	67.371	53.128	41.167	27.895	19.400	17.075	13.229	11.538
Doc18 (%)	94.718	89.687	84.369	76.572	68.559	59.899	56.917	50.269	48.149
Doc19 (%)	93.883	90.040	83.984	74.762	65.755	56.348	54.196	45.804	42.638
Doc20 (%)	90.062	84.073	73.026	59.672	47.338	40.506	37.090	30.701	29.015
Doc21 (%)	70.378	62.290	51.471	42.542	34.769	29.202	28.466	24.212	23.109
Doc22 (%)	78.712	71.250	59.410	45.102	34.921	27.748	25.067	20.228	18.858
Doc23 (%)	82.586	75.697	69.382	58.830	51.093	45.489	43.302	39.940	37.835
Doc24 (%)	90.950	83.288	74.419	64.917	53.575	43.529	40.483	33.303	30.075
Doc25 (%)	86.246	78.904	71.610	61.633	54.059	47.278	45.079	40.149	38.348
Doc26 (%)	96.235	92.830	88.826	81.999	74.932	67.163	64.100	56.468	54.278
Doc27 (%)	86.337	80.473	69.984	58.150	46.530	39.215	35.395	29.048	26.735
Doc28 (%)	96.163	92.826	88.821	81.617	74.746	67.678	65.039	57.637	55.301
Doc29 (%)	95.936	92.386	88.578	81.320	74.519	67.261	64.067	56.752	54.413
Doc30 (%)	96.521	93.567	89.489	83.249	76.485	69.025	66.381	59.210	56.673
Doc31 (%)	95.600	91.846	87.439	80.216	73.386	65.891	63.837	56.716	54.591
Doc32 (%)	94.396	90.418	85.336	77.582	70.687	63.014	60.588	53.650	51.736
Doc33 (%)	94.444	90.670	84.684	76.580	70.021	61.845	60.014	52.348	50.517
Doc34 (%)	95.089	91.144	85.792	77.258	70.335	62.459	60.131	53.037	50.894
Doc35 (%)	95.353	92.267	86.572	78.086	71.014	61.903	59.644	52.443	49.578
Doc36 (%)	92.801	88.404	80.331	70.572	62.048	54.669	51.386	44.699	44.036
Doc37 (%)	88.715	84.788	76.396	67.056	59.527	52.084	49.727	42.868	41.661

Doc38 (%)	81.413	74.164	62.082	50.372	42.007	35.316	34.015	29.089	27.881
Doc39 (%)	80.087	72.629	61.912	50.253	42.071	35.047	34.106	28.096	28.023
Doc40 (%)	83.012	77.162	67.352	56.380	48.640	41.896	39.443	32.533	31.217
Doc41 (%)	85.236	79.706	69.305	57.788	49.975	42.745	40.386	34.145	33.283
Doc42 (%)	80.635	75.000	62.801	52.735	43.846	36.461	34.847	28.228	26.833
Doc43 (%)	87.955	82.066	72.228	61.462	52.382	44.722	42.502	35.727	34.758
Doc44 (%)	82.919	77.534	66.913	55.940	47.081	40.386	37.466	32.114	30.822
Doc45 (%)	63.614	54.081	44.112	33.084	25.047	20.810	20.000	16.075	15.140
Doc46 (%)	65.186	55.189	43.492	30.545	22.527	15.828	14.023	11.454	10.691
Doc47 (%)	92.724	87.663	79.794	71.174	62.950	52.392	49.268	42.467	41.716
Doc48 (%)	92.037	87.249	79.310	68.866	61.126	53.758	50.905	44.232	41.727
Doc49 (%)	89.919	82.451	74.497	61.457	50.682	39.006	34.543	27.075	23.514
Doc50 (%)	96.860	94.169	90.792	83.245	77.309	68.021	64.908	56.623	54.195
Doc51 (%)	90.613	85.990	78.550	68.117	60.048	52.365	49.199	42.054	39.808
Doc52 (%)	90.067	84.884	77.404	67.584	59.127	50.794	48.162	41.455	39.434
Doc53 (%)	89.153	83.973	77.575	69.226	63.315	57.160	54.113	46.496	44.729
Doc54 (%)	93.977	88.820	83.243	73.159	66.256	56.687	53.357	44.951	42.962
Doc55 (%)	85.615	79.712	70.500	59.627	50.331	41.951	39.796	33.384	31.128
Doc56 (%)	88.079	81.555	72.988	62.409	54.909	45.244	43.598	37.439	35.244
Doc57 (%)	82.153	78.470	68.272	50.425	39.660	34.561	33.144	25.779	23.229
Doc58 (%)	86.263	79.960	70.273	60.232	50.913	42.742	39.839	32.163	31.015
Doc59 (%)	84.683	77.374	69.011	57.105	48.400	39.905	37.139	30.884	28.921
Doc60 (%)	86.322	81.722	73.298	62.094	53.466	46.126	43.611	36.496	35.678
Doc61 (%)	90.055	79.669	74.254	61.989	51.381	45.525	40.442	33.702	31.934
Doc62 (%)	86.689	81.617	72.775	62.359	53.673	47.338	44.151	38.496	36.183
Doc63 (%)	87.938	81.966	75.334	64.456	57.334	47.589	45.728	38.978	36.305
Doc64 (%)	92.390	88.272	82.640	75.656	67.905	61.344	57.126	51.231	49.071
Doc65 (%)	90.577	85.756	79.845	71.076	63.348	55.911	53.222	45.954	44.380
Doc66 (%)	88.039	80.764	72.010	61.406	53.514	47.472	45.869	38.718	37.238
Doc67 (%)	90.981	86.076	80.142	69.778	64.003	58.307	55.380	49.051	46.994
Doc68 (%)	87.304	82.193	72.960	63.644	57.708	47.403	45.342	39.406	37.923
Doc69 (%)	84.430	76.660	63.471	50.074	39.357	30.128	26.853	21.107	19.202
Doc70 (%)	69.597	61.791	48.316	36.237	28.184	21.693	19.885	16.598	14.955
Doc71 (%)	65.688	56.981	47.192	35.033	28.800	24.781	22.257	18.959	18.032
Doc72 (%)	95.158	91.900	86.968	79.427	72.127	62.579	60.090	50.362	49.140
Doc73 (%)	78.175	69.809	58.711	44.631	38.196	32.844	30.575	26.420	25.325
Doc74 (%)	89.309	81.750	74.346	63.940	55.242	46.648	44.473	37.613	36.474
Doc75 (%)	90.819	85.745	78.805	69.281	62.216	54.497	52.163	46.374	43.697
Doc76 (%)	94.768	90.205	83.391	73.778	66.457	57.128	55.628	48.489	46.279
Doc77 (%)	94.116	90.264	85.021	75.749	68.331	60.342	58.131	50.571	48.966
Doc78 (%)	93.737	89.601	83.456	75.447	68.040	60.340	58.008	50.572	49.047
Doc79 (%)	93.302	89.684	82.211	74.419	67.839	58.987	56.749	49.798	48.267
Doc80 (%)	93.686	90.494	84.366	76.375	68.872	61.417	59.273	52.004	50.116
Doc81 (%)	89.712	84.462	76.080	65.919	56.118	49.069	46.698	40.305	37.997
Doc82 (%)	81.198	71.527	60.195	49.368	41.213	33.995	31.361	27.138	25.334
Doc83 (%)	79.277	70.186	55.621	42.620	34.457	25.122	22.385	19.013	17.107
Doc84 (%)	93.119	88.637	82.555	73.535	64.906	56.593	54.138	46.532	44.858
Doc85 (%)	93.624	89.455	82.645	73.533	66.459	58.234	54.650	47.218	45.897
Doc86 (%)	93.424	88.493	81.610	72.611	64.196	56.339	52.717	46.447	44.023
Doc87 (%)	90.215	86.219	79.206	67.872	59.446	51.509	49.660	41.560	40.011
Doc88 (%)	93.506	89.261	81.023	72.927	63.565	54.878	52.123	45.488	43.182
Doc89 (%)	94.123	90.798	84.711	77.101	69.656	62.983	59.705	52.751	50.503
Doc90 (%)	93.215	87.040	79.607	69.223	61.042	51.485	49.990	41.534	39.548
Doc91 (%)	94.451	90.132	83.701	74.937	68.080	59.694	57.677	49.953	48.077
Doc92 (%)	95.834	93.797	88.298	81.022	74.449	68.136	65.266	57.878	55.767
Doc93 (%)	92.918	88.214	80.951	70.742	63.867	55.906	53.244	46.265	43.887
Doc94 (%)	92.744	88.071	81.695	71.848	63.479	55.303	52.550	45.036	43.374
Doc95 (%)	85.368	78.237	67.684	56.184	47.711	39.868	37.237	31.316	30.447
Doc96 (%)	92.458	87.738	80.556	70.834	62.753	55.311	52.330	45.949	44.229
Doc97 (%)	92.601	89.144	82.798	72.401	65.676	57.656	54.523	47.259	45.666
Doc98 (%)	91.231	87.034	78.918	68.315	59.733	51.150	48.601	41.480	39.708
Doc99 (%)	87.569	80.045	70.304	57.673	47.359	39.836	37.668	30.319	29.547
Doc100 (%)	89.033	82.005	75.022	62.265	54.208	46.061	44.091	37.690	36.213

Fig. 5.10: The output of the test case 02-03.

To analyze the result of this test case, we counted the acceptance percentages in the three desired ranges as shown in Fig. 5.11.

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.4	1	6	93
2	0.5	1	15	84
3	0.6	6	39	55
4	0.7	16	66	18
5	0.8	29	66	5
6	0.9	48	51	1
7	1.0	57	42	1
8	1.1	82	17	1
9	1.2	88	11	1

Fig. 5.11: The statistical result of the test case 02-03.

5.2.3.4. Test Case Conclusion

The statistical result showed that the best results were for $\alpha > 1.0$. Next, we need to verify if we get the same result for the same author's books?

5.2.4. Test Case 02-04: Regular Test (Same Author)

5.2.4.1. Test Case Objective

This test case focused on all books of the same author. In other words, we learned with a book of an author and tested with all of his or her books. The objective of this test case is to verify our conclusion from test case 02-03. Strictly speaking, we expect to have the best result for $\alpha > 1.0$. But please note that the testing books are from the same author. Therefore, the desired result would be the acceptance percentages in the range of $76 < \alpha < 100$.

5.2.4.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Learning Books	T S Arthur - Grappling with the Monster.txt
Testing Books	T S Arthur - Grappling with the Monster.txt
	T S Arthur - After the Strom.txt
	T S Arthur - Alls for the Best.txt
	T S Arthur - Cast Adrift.txt
	T S Arthur - Danger or Wounded.txt
	T S Arthur - Finger Posts .txt
	T S Arthur - Friends and Neighbors.txt
	T S Arthur - Hair Breadth Escapes.txt
	T S Arthur - Heart Histories and Life Pictures.txt
	T S Arthur - Home Lights and Shadows.txt
	T S Arthur - Lessons in Life.txt
	T S Arthur - Lizzy Glenn.txt
	T S Arthur - Married Life.txt
	T S Arthur - Ten Nights in a Bar Room.txt
	T S Arthur - The Allen House.txt
	T S Arthur - The Good Time Coming.txt
	T S Arthur - The Hand But Not the Heart.txt
	T S Arthur - The Home Mission.txt
	T S Arthur - The Lights and Shadows of Real Life.txt
	T S Arthur - The Wedding Guest.txt
	T S Arthur - Trials and Confessions of a Housekeeper.txt
	T S Arthur - Woman's Trials.txt

T S Arthur - Words for the Wise.txt

T S Arthur - Words of Cheer for the Tempted.txt

T S Arthur - Wreaths of Friendship.txt

5.2.4.3. Test Output

Alpha	0.40	0.50	0.60	0.70	0.80	0.90	1.00	1.10	1.20
Doc01 (%)	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000
Doc02 (%)	90.448	87.001	81.647	73.677	65.768	54.005	51.876	42.791	40.904
Doc03 (%)	91.655	87.910	82.810	77.389	68.367	57.026	53.852	45.043	44.437
Doc04 (%)	90.598	88.076	82.546	75.843	67.542	56.263	53.564	45.380	43.840
Doc05 (%)	90.003	86.537	81.134	74.184	67.031	55.486	53.164	44.581	43.622
Doc06 (%)	90.913	87.046	82.223	75.722	68.336	57.456	54.613	47.297	45.480
Doc07 (%)	85.605	79.678	72.883	63.738	56.054	45.512	42.231	35.097	33.785
Doc08 (%)	75.660	69.990	54.252	44.575	32.796	22.141	21.408	15.934	15.591
Doc09 (%)	89.548	85.233	80.975	72.364	64.441	53.338	49.823	41.566	40.320
Doc10 (%)	89.794	85.701	80.589	74.043	66.138	53.462	51.896	42.822	41.803
Doc11 (%)	89.384	85.344	80.385	70.911	64.113	51.714	49.819	40.596	39.621
Doc12 (%)	87.306	84.045	77.467	68.443	59.228	47.105	44.550	36.695	35.472
Doc13 (%)	90.076	84.369	79.280	71.774	62.047	51.617	48.018	39.865	38.263
Doc14 (%)	90.892	87.193	83.283	76.539	67.736	57.176	54.601	47.272	45.657
Doc15 (%)	89.617	85.565	77.915	70.305	61.180	48.555	45.801	36.873	35.398
Doc16 (%)	91.189	86.885	82.897	75.441	68.112	56.274	54.524	46.280	44.846
Doc17 (%)	93.205	89.798	87.169	79.744	73.820	63.192	60.285	52.046	51.139
Doc18 (%)	88.757	85.268	78.832	71.879	62.988	51.098	49.522	40.812	39.364
Doc19 (%)	88.757	85.367	80.000	71.356	62.712	51.122	48.918	40.460	38.878
Doc20 (%)	80.842	75.421	65.737	56.842	47.737	36.921	35.421	28.053	27.842
Doc21 (%)	88.258	84.757	79.356	70.734	62.272	52.090	48.350	41.568	39.648
Doc22 (%)	89.846	86.254	80.610	72.023	65.406	54.577	50.824	42.938	41.534
Doc23 (%)	87.065	82.805	76.150	67.382	59.359	47.512	43.874	36.847	35.541
Doc24 (%)	83.084	77.578	67.638	57.524	48.430	37.045	33.607	26.981	25.585
Doc25 (%)	83.572	78.648	72.695	62.399	54.566	42.883	39.839	33.393	31.647

Fig. 5.12: The output of the test case 02-04.

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.4	0	1	24
2	0.5	0	2	23
3	0.6	0	5	20
4	0.7	1	20	4
5	0.8	3	21	1
6	0.9	8	16	1
7	1.0	15	9	1
8	1.1	23	1	1
9	1.2	23	1	1

Fig. 5.13: The statistical result of the test case 02-04.

5.2.4.4. Test Case Conclusion

The above results clearly show that we get the worst result for $\text{Alpha} > 1.0$ and this is in contradiction with the conclusion of test case 02-03.

5.3. Special Tests

The following tests are designed to support our previous conclusions.

5.3.1. Test Case 03-01: Half Book Test

5.3.1.1. Test Case Objective

The objective of this test case is to show how reliable this technique is. We divided a book into two halves. Then we learned by one half and tested by another half. We know that some books are written by two or more authors, but the books we chose were written by one author¹⁴.

5.3.1.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Learning Books	The first half of the following books: Agatha Christie - The Secret Adversary.txt Harrison Williams - Legends of Loudoun.txt John Milton - Paradise Lost.txt Julia Ward Howe - From the Oak to the Olive.txt Mary Wollstonecraft - Frankenstein.txt Oscar Wilde - The Picture of Dorian Gray.txt T S Arthur - Cast Adrift.txt

¹⁴ In fact, we did not find any evidence to prove that those books were written by two or more writers but we supported the result of this test case with odd-even test in the next sub-section.

5.3.1.3. Test Output

i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Doc04 (%)	Doc05 (%)	Doc06 (%)	Doc07 (%)
1	0.10	99.885	99.951	85.060	100.000	90.618	98.787	99.104
2	0.20	99.611	95.868	98.916	93.257	91.033	98.755	98.912
3	0.30	98.648	87.069	87.711	91.663	74.347	97.542	95.808
4	0.40	97.045	74.380	58.072	89.658	85.036	89.946	83.424
5	0.50	94.937	70.637	46.265	82.232	74.287	87.935	79.488
6	0.60	89.851	68.255	40.482	65.695	54.394	83.817	78.432
7	0.70	89.118	59.893	26.988	55.626	44.715	76.189	74.336
8	0.80	85.040	52.990	22.410	47.289	38.658	75.870	66.272
9	0.90	81.375	51.191	16.867	40.957	26.960	68.848	59.136
10	1.00	76.472	48.274	18.675	38.223	22.506	64.698	57.280
11	1.10	73.356	42.343	14.337	27.107	17.577	61.443	51.776
12	1.20	71.913	41.857	12.651	25.467	16.211	58.506	49.120

Fig. 5.14: The output of the test case 03-01

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.1	0	0	7
2	0.2	0	0	7
3	0.3	0	1	6
4	0.4	0	2	5
5	0.5	1	2	4
6	0.6	1	3	3
7	0.7	2	3	2
8	0.8	3	3	1
9	0.9	3	3	1
10	1.0	4	2	1
11	1.1	4	3	0
12	1.2	5	2	0

Fig. 5.15: The statistical result of the test case 03-01

5.3.1.4. Test Case Conclusion

The table in Fig. 5.15 shows that we get better response for alphas between 0.1 and 0.5. This is in contrast with the previous results; in regular tests, we gained better results between 0.4 and 1.3. Thus, we need more test cases to verify the results.

5.3.2. Test Case 03-02: Half Book Test (More Books)

5.3.2.1. Test Case Objective

This test case continues the above experience with more books. To be sure, we also ran the same test case for 43 additional books and the results are shown in Fig. 5.17.

5.3.2.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Learning Books	The first half of the following books: Charles Dickens - A tale of Two Cities.txt Charles Dickens - David Copperfield.txt J K Rollins - Harry Potter 4.txt J K Rollins - Harry Potter 7.txt Jane Austen - Persuasion.txt

Mark Twain - The Adventures of Huckleberry Finn.txt
 T S Arthur - The Lights and Shadows of Real Life.txt
 Testing Books The second half of the above books

5.3.2.3. Test Output

i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Doc04 (%)	Doc05 (%)	Doc06 (%)	Doc07 (%)
1	0.10	99.884	99.615	93.253	99.937	99.855	100.000	99.302
2	0.20	97.929	98.688	97.534	98.747	100.000	97.738	93.141
3	0.30	96.556	97.595	98.019	99.253	81.998	74.835	98.517
4	0.40	85.851	92.164	94.390	89.567	70.849	80.961	86.859
5	0.50	82.593	86.356	91.209	91.124	67.954	74.835	85.969
6	0.60	77.193	79.071	87.145	83.730	56.708	67.595	81.274
7	0.70	72.074	78.625	85.395	82.856	54.923	62.583	73.019
8	0.80	64.068	69.783	81.855	74.626	44.739	58.336	65.812
9	0.90	56.295	62.148	74.431	71.195	39.865	49.008	58.778
10	1.00	53.386	58.501	71.403	68.093	34.749	48.695	55.654
11	1.10	49.151	53.463	67.391	63.687	30.212	44.100	48.743
12	1.20	47.475	52.283	64.618	60.889	30.019	40.933	46.841

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.1	0	0	7
2	0.2	0	0	7
3	0.3	0	1	6
4	0.4	0	1	6
5	0.5	0	2	5
6	0.6	0	2	5
7	0.7	0	4	3
8	0.8	1	5	1
9	0.9	2	5	0
10	1.0	2	5	0
11	1.1	4	3	0
12	1.2	4	3	0

Fig. 5.16: The output of the test case 03-02

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.1	0	1	42
2	0.2	1	1	41
3	0.3	2	5	36
4	0.4	2	8	33
5	0.5	5	18	20
6	0.6	8	21	14
7	0.7	12	20	11
8	0.8	16	26	1
9	0.9	22	20	1
10	1.0	26	16	1
11	1.1	32	11	0
12	1.2	34	9	0

Fig. 5.17: The statistical result of the test case 03-02 for 43 books

5.3.2.4. Test Case Conclusion

The result of the test case 03-01 is confirmed. It means that for half book test, the best results are gained for alpha between 0.1 and 0.5. To be sure, we repeated the above test for 43 additional books and we confirmed the same result as Fig. 5.17 shows.

5.3.3. Test Case 03-03: Odd-Even Test

5.3.3.1. Test Case Objective

In this test case, we divided a book into two halves as follows: we created the learning book from the odd sentences and we created the testing book from the even sentences. We repeated the same test for seven books. We purposely chose the same books as in the test case 03-01 to be able to compare the test results.

5.3.3.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia BFS
Learning Books	The odd sentences of the following books: The first half of the following books: Agatha Christie - The Secret Adversary.txt Harrison Williams - Legends of Loudoun.txt John Milton - Paradise Lost.txt Julia Ward Howe - From the Oak to the Olive.txt Mary Wollstonecraft - Frankenstein.txt Oscar Wilde - The Picture of Dorian Gray.txt T S Arthur - Cast Adrift.txt
Testing Books	The even sentences of the above books

5.3.3.3. Test Output

i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Doc04 (%)	Doc05 (%)	Doc06 (%)	Doc07 (%)
1	0.10	100.000	99.891	100.000	99.861	97.022	99.035	99.296
2	0.20	96.939	99.563	100.000	93.016	93.151	99.035	96.539
3	0.30	97.266	88.354	89.401	93.663	85.289	93.695	73.599
4	0.40	93.737	79.661	61.097	83.811	71.590	92.187	79.525
5	0.50	95.116	77.911	51.621	80.342	74.687	84.706	76.591
6	0.60	90.559	62.274	40.773	70.490	53.425	81.418	73.218
7	0.70	87.684	51.558	32.793	56.198	53.365	78.824	72.309
8	0.80	84.062	54.183	23.441	50.694	37.761	74.268	70.930
9	0.90	79.878	45.161	16.334	38.529	27.099	68.778	63.508
10	1.00	76.817	43.193	14.838	34.644	23.347	64.947	61.103
11	1.10	72.634	39.038	12.469	25.856	19.833	59.970	55.265
12	1.20	70.507	36.796	10.848	25.347	16.677	58.160	52.068

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.1	0	0	7
2	0.2	0	0	7
3	0.3	0	1	6
4	0.4	0	2	5
5	0.5	0	2	5
6	0.6	1	4	2
7	0.7	1	4	2
8	0.8	3	3	1
9	0.9	4	2	1
10	1.0	4	2	1
11	1.1	4	3	0
12	1.2	4	3	0

Fig. 5.18: The output of the test case 03-03

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.1	0	1	42
2	0.2	1	1	41
3	0.3	2	5	36
4	0.4	2	8	33
5	0.5	5	18	20
6	0.6	8	21	14
7	0.7	12	20	11
8	0.8	16	26	1
9	0.9	22	20	1
10	1.0	26	16	1
11	1.1	32	11	0
12	1.2	34	9	0

Fig. 5.19: The statistical result of the test case 03-03 for 43 books

5.3.3.4. Test Case Conclusion

Fig. 5.19 shows that the best alpha is between 0.1 and 0.5 and Fig. 5.20 shows the statistical result of the same test case for 43 books. Please note that the same conclusion of the test case 5.3.2 is confirmed.

STATISTICAL REPORT				
i	Alpha	0 <= n < 50	51 <= n < 75	76 <= n < 100
1	0.1	0	0	43
2	0.2	0	1	42
3	0.3	0	4	39
4	0.4	2	9	32
5	0.5	4	14	25
6	0.6	7	22	14
7	0.7	9	27	7
8	0.8	17	25	1
9	0.9	21	21	1
10	1.0	26	16	1
11	1.1	31	12	0
12	1.2	34	9	0

Fig. 5.20: The statistical result of the test case 03-03 for 43 books

5.4. Parameters Effects

During the following sub-sections, we measure the importance of two parameters: chopping unit (paragraph versus sentence) and merging algorithm (DFS versus BFS). The key criterion of the importance of a parameter is its contribution in making a bigger acceptance difference when we feed the same author's book and the other author's book into a SDFA.

5.4.1. Test Case 04-01: Sentence versus Paragraph

5.4.1.1. Test Case Objective

In this test case, we investigated the chopping unit effect. We repeated the test case 02-01 but changed the chopping unit to paragraph.

5.4.1.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Paragraph
Test Doc Chopping Unit	Paragraph
Merge Algorithm	Alergia BFS
Learning Books	HP5.txt
Testing Books	HP5.txt
	HP3.txt
	One Hundred Years of Solitude.txt

5.4.1.3. Test Output

					PARAGRAPH BFS
i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Diff.
1	0.10	100.000	97.817	99.339	-1.522
2	0.20	100.000	72.052	85.997	-13.945
3	0.30	100.000	0.000	2.114	-2.114
4	0.40	100.000	0.000	1.189	-1.189
5	0.50	100.000	0.000	0.793	-0.793
6	0.60	100.000	0.000	0.793	-0.793
7	0.70	100.000	0.000	0.793	-0.793
8	0.80	100.000	0.000	0.793	-0.793
9	0.90	100.000	0.000	0.793	-0.793
10	1.00	100.000	0.000	0.793	-0.793
11	1.10	100.000	0.000	0.793	-0.793
12	1.20	100.000	0.000	0.793	-0.793

Fig. 5.21: The output of the test case 04-01

5.4.1.4. Test Case Conclusion

The self test¹⁵ result in Fig. 5.21 ensured that the SDFA was created correctly and the results of the same author and different author showed that the paragraph as the chopping unit did not add any value to the desired result.

5.4.2. Test Case 04-02: Sentence versus Paragraph (More Books)

5.4.2.1. Test Case Objective

This is the same test as test case 04-01 but with more books to support our previous conclusion.

¹⁵ For the definition of "self test," please refer to section 5.2.

5.4.2.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Paragraph
Test Doc Chopping Unit	Paragraph
Merge Algorithm	Alergia BFS
Learning Books	Jane Austen - Emma.txt
Testing Books	Jane Austen - Emma.txt Jane Austen - Mansfield Park.txt Charles Dickens - Oliver Twist.txt

5.4.2.3. Test Output

i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Diff.
1	0.10	100.000	99.488	99.787	-0.299
2	0.20	100.000	93.235	97.179	-3.944
3	0.30	100.000	60.944	80.575	-19.631
4	0.40	100.000	26.322	51.277	-24.955
5	0.50	100.000	35.020	61.096	-26.076
6	0.60	100.000	20.694	45.130	-24.436
7	0.70	100.000	17.908	39.862	-21.954
8	0.80	100.000	13.530	27.861	-14.331
9	0.90	100.000	8.812	22.166	-13.354
10	1.00	100.000	8.698	19.878	-11.180
11	1.10	100.000	4.832	13.624	-8.792
12	1.20	100.000	4.491	12.853	-8.362

Fig. 5.22: The output of the test case 04-02

5.4.2.4. Test Case Conclusion

With the results shown in the Fig. 5.22, we concluded that the paragraph as chopping unit did not add any value to the desired result.

5.4.3. Test Case 04-03: DFS versus BFS

5.4.3.1. Test Case Objective

In this test case, we investigated the effect of depth first search (DFS) and breadth first search (BFS) in the Alergia algorithm [7]. We repeated test case 02-01 but changed the merging algorithm to DFS.

5.4.3.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.3
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia DFS
Learning Books	HP5.txt
Testing Books	HP5.txt
	HP3.txt
	One Hundred Years of Solitude.txt

5.4.3.3. Test Output

i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	SENTENCE	
					DFS	BFS
					Diff.	Diff.
1	0.10	100.000	97.944	90.430	7.514	0.126
2	0.20	100.000	87.995	53.593	34.402	0.948
3	0.30	100.000	85.040	49.350	35.690	6.810
4	0.40	100.000	78.871	40.701	38.170	9.328
5	0.50	100.000	79.075	38.082	40.993	17.573
6	0.60	100.000	78.604	36.656	41.948	25.804
7	0.70	100.000	76.044	34.670	41.374	29.285
8	0.80	100.000	73.313	32.629	40.684	35.160
9	0.90	100.000	70.036	29.289	40.747	39.487
10	1.00	100.000	69.415	28.169	41.246	40.558
11	1.10	100.000	66.631	26.165	40.466	40.838
12	1.20	100.000	66.074	25.551	40.523	40.588

Fig. 5.23: The output of the test case 04-03.

5.4.3.4. Test Case Conclusion

Fig. 5.23 shows that the acceptance difference using DFS as merging algorithm in Alergia algorithm [7] is more stable and converges quickly to maximum value. We repeated the same test with additional books to verify our conclusion.

5.4.4. Test Case 04-04: DFS versus BFS (More Books)

5.4.4.1. Test Case Objective

This test case is the same as test case 04-03 but with the books used in the test case 02-02.

5.4.4.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Sentence
Test Doc Chopping Unit	Sentence
Merge Algorithm	Alergia DFS
Learning Books	Jane Austen - Emma.txt
Testing Books	Jane Austen - Emma.txt Jane Austen - Mansfield Park.txt Charles Dickens - Oliver Twist.txt

5.4.4.3. Test Output

					DFS	BFS
i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Diff.	Diff.
1	0.10	100.000	96.998	97.900	-0.902	-0.003
2	0.20	100.000	75.757	84.344	-8.587	-2.190
3	0.30	100.000	73.036	82.671	-9.635	-1.215
4	0.40	100.000	59.727	72.487	-12.760	-3.925
5	0.50	100.000	51.744	66.255	-14.511	-5.975
6	0.60	100.000	48.921	64.868	-15.947	-10.820
7	0.70	100.000	46.366	61.894	-15.528	-11.526
8	0.80	100.000	44.501	60.602	-16.101	-13.804
9	0.90	100.000	42.189	57.971	-15.782	-14.430
10	1.00	100.000	41.448	56.897	-15.449	-14.665
11	1.10	100.000	38.472	55.434	-16.962	-15.953
12	1.20	100.000	37.859	54.427	-16.568	-15.923

Fig. 5.24: The output of the test case 04-04.

5.4.4.4. Test Case Conclusion

As Fig. 5.24 shows, using DFS as merging algorithm did not add any value to the desired result.

5.4.5. Test Case 04-05: DFS and Paragraph

5.4.5.1. Test Case Objective

In this test case, we combined the effect of DFS as merging algorithm and paragraph as chopping unit and compared the results with BFS as merging algorithm and sentence as chopping unit.

5.4.5.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Paragraph
Test Doc Chopping Unit	Paragraph
Merge Algorithm	Alergia DFS
Learning Books	HP5.txt
Testing Books	HP5.txt
	HP3.txt
	One Hundred Years of Solitude.txt

5.4.5.3. Test Output

					PARAGRAPH		SENTENCE	
					DFS	BFS	DFS	BFS
i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	Diff.	Diff.	Diff.	Diff.
1	0.10	100.000	0.437	14.927	-14.490	-1.522	7.514	0.126
2	0.20	100.000	0.000	10.832	-10.832	-13.945	34.402	0.948
3	0.30	100.000	0.000	0.793	-0.793	-2.114	35.690	6.810
4	0.40	100.000	0.000	0.793	-0.793	-1.189	38.170	9.328
5	0.50	100.000	0.000	0.793	-0.793	-0.793	40.993	17.573
6	0.60	100.000	0.000	0.793	-0.793	-0.793	41.948	25.804
7	0.70	100.000	0.000	0.793	-0.793	-0.793	41.374	29.285
8	0.80	100.000	0.000	0.793	-0.793	-0.793	40.684	35.160
9	0.90	100.000	0.000	0.793	-0.793	-0.793	40.747	39.487
10	1.00	100.000	0.000	0.793	-0.793	-0.793	41.246	40.558
11	1.10	100.000	0.000	0.793	-0.793	-0.793	40.466	40.838
12	1.20	100.000	0.000	0.793	-0.793	-0.793	40.523	40.588

Fig. 5.25: The output of the test case 04-05.

5.4.5.4. Test Case Conclusion

Fig. 5.25 shows that we gained the best result from combination of DFS as merging algorithm and sentence as chopping unit. To investigate this conclusion, we tested the same configuration with more books.

5.4.6. Test Case 04-06: DFS and Paragraph (More Books)

5.4.6.1. Test Case Objective

This test case is the same as test case 04-05 configuration but with different books.

5.4.6.2. Test Condition

Starting Alpha	0.1
Increment	0.1
Upper Limit	1.2
Learn Doc Chopping Unit	Paragraph
Test Doc Chopping Unit	Paragraph
Merge Algorithm	Alergia DFS
Learning Books	Jane Austen - Emma.txt
Testing Books	Jane Austen - Emma.txt
	Jane Austen - Mansfield Park.txt
	Charles Dickens - Oliver Twist.txt

5.4.6.3. Test Output

i	Alpha	Doc01 (%)	Doc02 (%)	Doc03 (%)	PARAGRAPH		SENTENCE	
					DFS	BFS	DFS	BFS
					Diff.	Diff.	Diff.	Diff.
1	0.10	100.000	23.536	42.629	-19.093	-0.299	-0.902	-0.003
2	0.20	100.000	42.524	68.121	-25.597	-3.944	-8.587	-2.190
3	0.30	100.000	26.265	52.155	-25.890	-19.631	-9.635	-1.215
4	0.40	100.000	15.406	34.726	-19.320	-24.955	-12.760	-3.925
5	0.50	100.000	9.608	24.188	-14.580	-26.076	-14.511	-5.975
6	0.60	100.000	7.732	20.037	-12.305	-24.436	-15.947	-10.820
7	0.70	100.000	6.595	17.190	-10.595	-21.954	-15.528	-11.526
8	0.80	100.000	5.856	14.795	-8.939	-14.331	-16.101	-13.804
9	0.90	100.000	4.434	13.225	-8.791	-13.354	-15.782	-14.430
10	1.00	100.000	5.003	12.799	-7.796	-11.180	-15.449	-14.665
11	1.10	100.000	3.638	11.442	-7.804	-8.792	-16.962	-15.953
12	1.20	100.000	3.695	10.857	-7.162	-8.362	-16.568	-15.923

Fig. 5.26: The output of the test case 04-06.

5.4.6.4. Test Case Conclusion

Fig. 5.26 shows the result of the effect of paragraph as chopping unit and DFS as merging in the Alergia algorithm [7]. The results of the test cases 02-02 and 04-04 are added to the Fig. 5.26 for comparison. The results are negative and not acceptable. Therefore, we conclude that the automata technique works for some books but does not work for some other books. We ran this test hundreds of times for different books and the above conclusion was confirmed. Therefore, the technique is not reliable with the configurations that we employed in this research.

6. Conclusion

In this research, we asked three questions that previous authors [1][2][3] and this thesis tried to answer. We assumed that every author has his or her own style of writing. We considered the sequences of function words¹⁶ as the author's style of writing to respond to the second question. To respond to the third question, we used automata machines to represent the author's style of writing.

We implemented reliable and expandable software¹⁷ from scratch using object-oriented design and implemented it in Java. We designed four sets of test cases to verify if the automata machines technique can represent the author's style of writing. The first set of test cases verified the accuracy of the software's implementation. The second set of test cases showed that the automata machine technique represented the styles of some authors but not others. To explain these results, one could assume that the writer developed new styles over the course of his or her life. To verify this conclusion, we designed the third set of test cases. The half-book test rejected the conclusion because we could not assume that the writer developed a new style while writing the book. But one still could assume that the books were written by two or more writers. For this reason, we designed the odd-even test case. The results of the odd-even tests were almost the same as the half-book tests. In that case, we could not assume that the odd and even sentences of the book were written by different authors.

¹⁶ For definition of "function word," please refer to 2.2.

¹⁷ For a copy of the software, please download it from Dr. Lin's website[14].

We designed the fourth set of test cases to verify the effect of the several parameters on the tests' results. Chopping the books into paragraphs and using DFS as the merging method in the Alergia algorithm [7] did not add any value to our previous results.

Overall, our tests showed that we have captured only the patterns of the collections of the single sentences in a book which unfortunately is not its full contents. Establishing variable chopping units or a less forceful chopping approach would be a promising approach that is beyond the scope of this MS thesis.

7. Future Direction

This chapter contains our recommendations and suggestions for future research.

7.1. Function Words Modifications

In section 1.3, we defined the author's style of writing as the sequences of the function words.¹⁸ Therefore, changes in the choice of function words can tremendously affect the results of this research. In the software we developed, we considered just one-word function words and used the main word of the multiple-word function phrases. For example, for the function phrase "the lack of," we just used the word "lack" which is not as precise as using the entire phrase.

7.2. Variable Chopping Units

In this thesis, we focused on constant chopping units such as sentences or paragraphs. Another approach could be variable chopping units such as: one sentence, two sentences and so on.

¹⁸ For definition of "function word," please refer to section 2.2.

7.3. Adding Intelligence

Currently, the software uses a fix algorithm to create the automaton and the automaton remains unchanged during the test phase. A new approach would be adding intelligence to the technique to let it learn from the new experiments. In this way, it can modify the automaton in such a way that it better recognizes the same author while not mixing up his or her with that of the other authors.

7.4. Software

This section contains our recommendations about how to improve the quality of the software.¹⁹

7.4.1. Supporting Other Languages

This research focused on English language documents but the software has the potential to be used for other languages even those written right-to-left. The software's tokenizer is able to tokenize the contents of any languages but the rest of the software needs some modifications to be able to test documents written in other languages.

7.4.2. Graphical User Interface (GUI)

Currently, the software can be used by Java developers. By adding a graphical user interface (GUI), the software can be used by any researcher.

¹⁹ For downloading a copy of the software, please refer to Dr. Lin's website[14].

7.4.3. Multiword Support

The software's tokenizer can distinguish one word at a time but in natural languages like English, there are many phrasal words that include two or more words. Therefore, the software can be improved to be able to distinguish the phrasal words.

7.4.4. Graphical Output for PTA and SDFA

The program might be able to draw the PTA's and SDFA's graphically. There are some open source programs that can be adapted and customized for this software.

Bibliography

- [1] S. Zhang, "An Automata Based Authorship Identification System," M.S. Thesis, San Jose State University, San Jose, 2008.
- [2] Y. Lu, "An Automata Based Text Analysis System," M. S. Thesis, San Jose State University, San Jose, 2009.
- [3] Q. Yu, "Learning Author's Writing Pattern System," M. S. thesis, San Jose State University, San Jose, 2011.
- [4] T. Y. Lin, "Rough Patterns in Data-Rough Sets and Foundation of Intrusion Detection Systems," *Journal of Foundation of Computer Science and Decision Support*, vol. 18, pp. 225-241, 1993.
- [5] T. Y. Lin, "Patterns in Numerical Data: Practical Approximations to Kolmogorov Complexity," *SFDGrC*, vol. 1, pp. 509-513, 1999.
- [6] P. a. T.Y.Lin, "Kolmogorov Complexity Based Automata," in *IEEE International Conference on Granular Computing*, Beijing, China, 2005.
- [7] R. a. J.Oncina, "Learning stochastic regular grammars by means of a state merging method," in *Proceedings of the 2nd International Colloquium on Grammatical Inference (ICGI)*, Alicante, Spain, 1994.
- [8] T. K. a. e. all, *Analyzing English Grammar*, 6 ed., : Longman, 2009.
- [9] t. F. E. Wikipedia, "Hoeffding's inequality," [Online]. Available: http://en.wikipedia.org/wiki/Hoeffding's_inequality.
- [10] L. G. a. F. Morales, "Function Words," 2005. [Online]. Available: <http://www.sequencepublishing.com/academic.html>.
- [11] "Free eBooks by Project Gutenberg," 2013. [Online]. Available: <http://www.gutenberg.org/>.
- [12] P. Linz, *An Introduction to Formal Languages and Automata*, 4 ed., Sudbury: Jones and Bartlett Publishers, 2006, pp. 38-39.

[13] F. M. Leah Gilner, "Function Words," 2005. [Online]. Available: <http://www.sequencepublishing.com>.

[14] D. T. Y. Lin, "Dr. Tsau Young Lin Website," [Online]. Available: <http://xanadu.cs.sjsu.edu/~drtylin/>.

Appendixes

Appendix A Abbreviations

Table 3 lists the abbreviations used in this thesis.

Table 3: List of abbreviations

Abbreviation	Full Text
BFS	Breadth-First-Search
BFS	Breadth First Search
DFA	Deterministic Finite Automaton (or Acceptor)
DFS	Depth First Search
GUI	Graphical User Interface
IDE	Integrated Development Environment
NFA	Non-deterministic Finite Automaton (or Acceptor)
NLP	Natural Language Processing
PTA	Prefix Tree Acceptor
SDFA	Stochastic Deterministic Finite Automaton
UML	Unified Modeling Language

Appendix B Function Words

Appendix B shows the complete list of the function words sorted by the categorizations used in this thesis.

Auxiliary Verbs

Table 4: List of auxiliary verbs in function words (Code=1)

able	be	am	are	is
was	were	can	could	dare
have	has	had	may	might
must	need	ought	shall	should
will	would			

Conjunctions

Table 5: List of conjunctions in function words (Code=2)

accordingly	albeit	although	and	as
because	but	consequently	hence	however
if	nevertheless	nor	once	or
so	than	that	then	therefore
though	thus	unless	whatever	when
whenever	where	whereas	wherever	whether
whichever	while	whilst	whoever	whomever
yet				

Determiners

Table 6: List of determiners in function words (Code=3)

a	an	another	any	both
each	either	every	my	neither
no	other	our	the	their
these	this	those	your	

Prepositions

Table 7: List of prepositions in function words (Code=4)

aboard	about	above	absent	according
across	after	against	ahead	along
alongside	amid	amidst	among	amongst
anti	around	aside	astraddle	astride
at	away	bar	barring	before
behind	below	beneath	beside	besides
between	beyond	by	circa	close
concerning	considering	despite	down	due
during	except	excepting	excluding	failing
following	for	from	front	in
including	inside	instead	into	like
minus	near	next	notwithstanding	of
off	on	onto	opposite	out
outside	over	past	pending	per
pertaining	place	plus	regarding	respecting
round	save	saving	similar	since

aboard	about	above	absent	according
spite	through	throughout	thru	till
to	top	toward	towards	under
underneath	unlike	until	unto	up
upon	versus	via	wanting	with
within	without			

Pronouns

Table 8: List of pronouns in function words (Code=5)

anybody	anyone	anything	everybody	everyone
everything	he	her	hers	herself
him	himself	his	i	it
its	itself	me	mine	myself
no_one	nobody	none	nothing	one
ours	ourselves	she	somebody	someone
something	such	theirs	them	themselves
they	us	we	what	which
who	whom	whose	you	yours
yourself	yourselves			

Quantifiers

Table 9: List of quantifiers in function words (Code=6)

all	amount	bit	certain	couple
enough	few	fewer	heaps	lack
less	little	loads	lots	majority
many	masses	minority	more	most
much	number	numbers	part	plenty
plethora	quantities	quantity	remainder	rest
several	some	tons	various	whole

Appendix C Stop Words

Table 10 contains all stop words²⁰ used in this thesis.

Table 10: List of stop words

a	able	about	above	abroad
according	accordingly	across	actually	adj
after	afterwards	again	against	ago
ahead	ain't	all	allow	allows
almost	alone	along	alongside	already
also	although	always	am	amid
amidst	among	amongst	an	and
another	any	anybody	anyhow	anyone
anything	anyway	anyways	anywhere	apart
appear	appreciate	appropriate	are	aren't
around	as	a's	aside	ask
asking	associated	at	available	away
awfully	back	backward	backwards	be
became	because	become	becomes	becoming
been	before	beforehand	begin	behind
being	believe	below	beside	besides
best	better	between	beyond	both
brief	but	by	came	can
cannot	cant	can't	caption	cause
causes	certain	certainly	changes	clearly
c'mon	co	co.	com	come

²⁰ For the definition of stop words, please refer to section 2.2.2.

a	able	about	above	abroad
comes	concerning	consequently	consider	considering
contain	containing	contains	corresponding	could
couldn't	course	currently	dare	daren't
definitely	described	despite	did	didn't
different	directly	do	does	doesn't
doing	done	don't	down	downwards
during	each	edu	eg	eight
eighty	either	else	elsewhere	end
ending	enough	entirely	especially	et
etc	even	ever	evermore	every
everybody	everyone	everything	everywhere	ex
exactly	example	except	fairly	far
farther	few	fewer	fifth	first
five	followed	following	follows	for
forever	former	formerly	forth	forward
found	four	from	further	furthermore
get	gets	getting	given	gives
go	goes	going	gone	got
gotten	greetings	had	hadn't	half
happens	hardly	has	hasn't	have
haven't	having	he	he'd	he'll
hello	help	hence	her	here
hereafter	hereby	herein	here's	hereupon
hers	herself	he's	hi	him
himself	his	hither	hopefully	how

a	able	about	above	abroad
howbeit	however	hundred	i'd	ie
if	ignored	i'll	i'm	immediate
in	inasmuch	inc	inc.	indeed
indicate	indicated	indicates	inner	inside
insofar	instead	into	inward	is
isn't	it	it'd	it'll	its
it's	itself	i've	just	k
keep	keeps	kept	know	known
knows	last	lately	later	latter
latterly	least	less	lest	let
let's	like	liked	likely	likewise
little	look	looking	looks	low
lower	ltd	made	mainly	make
makes	many	may	maybe	mayn't
me	mean	meantime	meanwhile	merely
might	mighn't	mine	minus	miss
more	moreover	most	mostly	mr
mrs	much	must	mustn't	my
myself	name	namely	near	nearly
necessary	need	needn't	needs	neither
never	nevertheless	new	next	nine
ninety	no	nobody	non	none
nonetheless	no-one	nor	normally	not
nothing	notwithstanding	novel	now	nowhere
obviously	of	off	often	oh

a	able	about	above	abroad
ok	okay	old	on	once
one	ones	one's	only	onto
opposite	or	other	others	otherwise
ought	oughtn't	our	ours	ourselves
out	outside	over	overall	own
particular	particularly	past	per	perhaps
placed	please	plus	possible	presumably
probably	provided	provides	quite	rather
rd	re	really	reasonably	recent
recently	regarding	regardless	regards	relatively
respectively	right	round	said	same
saw	say	saying	says	second
secondly	see	seeing	seem	seemed
seeming	seems	seen	self	selves
sensible	sent	serious	seriously	seven
several	shall	shan't	she	she'd
she'll	she's	should	shouldn't	since
six	so	some	somebody	someday
somehow	someone	something	sometime	sometimes
somewhat	somewhere	soon	sorry	specified
specify	specifying	still	sub	such
sup	sure	take	taken	taking
tell	tends	than	thank	thanks
that	that'll	That is	the	their
theirs	them	themselves	then	thence

a	able	about	above	abroad
there	thereafter	thereby	there'd	therefore
therein	there'll	there're	there's	thereupon
there've	these	they	they'd	they'll
they're	they've	thing	things	think
third	thirty	this	thorough	thoroughly
those	though	three	through	throughout
thru	thus	till	to	together
too	took	toward	towards	tried
tries	truly	try	trying	twice
two	un	under	underneath	undoing
unfortunately	unless	unlike	unlikely	until
unto	up	upon	upwards	us
use	used	useful	uses	using
usually	value	various	versus	very
via	viz	vs	want	wants
was	wasn't	way	we	we'd
welcome	well	we'll	went	were
we're	weren't	we've	what	whatever
what'll	what's	what've	when	whence
whenever	where	where-after	whereas	whereby
wherein	where's	whereupon	wherever	whether
which	whichever	while	whilst	whither
who	who'd	whoever	whole	who'll
whom	whomever	who's	whose	why
will	willing	wish	with	within

a	able	about	above	abroad
without	wonder	won't	would	wouldn't
yes	yet	you	you'd	you'll
your	you're	yours	yourself	yourselves
you've	zero			

Appendix D Test Environment

Table 11 and Table 12 contain the test environment used in this thesis.

Table 11: The software specifications in the test environment

Software Environment	
Language	Java 1.7
IDE	NetBeans 7.1.2
OS	Windows XP Professional, SP3

Table 12: The hardware specifications in the test environment

Hardware Environment	
Make and Model	Lenovo T400
CPU	Intel(R) Core(TM)2 Duo CPU
Speed	P8400 @ 2.26GHz
RAM	3 GB

Appendix E List of eBooks

Table 13 lists all eBooks²¹ used in the thesis. The eBooks are downloaded from "Free eBooks by Project Gutenberg" website[11]. Please refer to the website for the copy right statements.

Table 13: The list of eBooks used in this thesis

Num.	Book's Name
1	Adam Smith - An Inquiry into the nature.txt
2	Agatha Christie - The Mysterious Affair at Styles.txt
3	Agatha Christie - The Secret Adversary.txt
4	Alexander Dumas - The Count of Monte Cristo.txt
5	Alexander Dumas - The Three Musketeers.txt
6	Arthur Conan Doyle -The Adventures of Sherlock Holmes.txt
7	Bram Stoker - Dracula.txt
8	Canaan Doyle - A Study in Scarlet.txt
9	Charles Dickens - A tale of Two Cities.txt
10	Charles Dickens - David Copperfield.txt
11	Charles Dickens - Great Expectations.txt
12	Charles Dickens - Oliver Twist.txt
13	Charlotte Bronte - Jane Eyre.txt
14	Daniel Defoe - Robinson Crusoe.txt
15	Dante Alighieri - The Divine Comedy.txt
16	Edgar Rice Burroughs - A Princess of Mars.txt
17	Elliott Whitney - The Pirate Shark.txt
18	Elliott Whitney - The Rogue Elephant.txt
19	Frank Baum - The Wonderful Wizard of Oz.txt
20	Friedrich Nietzsche - Beyond Good and Evil.txt
21	Gabriel Garcia Marquez - 100 Years of Solitude.txt
22	Harrison Williams - Legends of Loudoun.txt

²¹ These eBooks are included in the software and can be downloaded from Dr. Lin's website [14].

Num.	Book's Name
23	Herbert George Wells - The War of the Worlds.txt
24	Herman Melville - Moby Dick.txt
25	Herman Hesse - A Son of the City.txt
26	Hermann Hesse - Siddhartha.txt
27	J K Rollins - Harry Potter 1.txt
28	J K Rollins - Harry Potter 2.txt
29	J K Rollins - Harry Potter 3.txt
30	J K Rollins - Harry Potter 4.txt
31	J K Rollins - Harry Potter 5.txt
32	J K Rollins - Harry Potter 6.txt
33	J K Rollins - Harry Potter 7.txt
34	James Joyce - Dubliners.txt
35	James Matthew Barrie - Peter Pan.txt
36	Jane Austen - Emma.txt
37	Jane Austen - Lady Susan.txt
38	Jane Austen - Love and Freindship.txt
39	Jane Austen - Mansfield Park.txt
40	Jane Austen - Northanger Abbey.txt
41	Jane Austen - Persuasion.txt
42	Jane Austen - Pride and Prejudice.txt
43	Jane Austen - Sense and Sensibility.txt
44	John Milton - Paradise Lost.txt
45	Jonathan Swift - Gulliver's Traveles.txt
46	Joseph Conrad - Heart of Darkness.txt
47	Jules Verne - Around the World in 80 Days.txt
48	Julia Ward Howe - From the Oak to the Olive.txt
49	Laura Dent Crane - The Automobile Girsl .txt
50	Leo Tolstoy - Anna Karenina.txt
51	Leo Tolstoy - War and Peace.txt
52	Lewis Carroll - Alice's adventures in Wonderland.txt
53	Lucy Maud Montgomery - Anne of Green Gables.txt

Num.	Book's Name
54	Mark Twain - Adventures of Huckleberry Finn.txt
55	Mark Twain - Christian Science.txt
56	Mark Twain - Eve's Diary.txt
57	Mark Twain - Life on the Mississippi.txt
58	Mark Twain - Lords of the Houseltops.txt
59	Mark Twain - Roughing It.txt
60	Mark Twain - Sketches New and Old.txt
61	Mark Twain - Some Rambling Notes of an Idle Excursion.txt
62	Mark Twain - The 30,000 Dollar Bequest and Other Stories.txt
63	Mark Twain - The Adventures of Tom Sawyer.txt
64	Mark Twain - The American Claimant.txt
65	Mark Twain - The Curious Republic of Condor, and Other Whimsical Sketches.txt
66	Mark Twain - The Man That Corrupted Hadleyburg.txt
67	Mark Twain - Tom Sawyer, Detective.txt
68	Mary Wollstonecraft - Frankenstein.txt
69	Matthew Arnold - Celtic Literature.txt
70	Matthew Arnold - Culture and Anarchy.txt
71	Oscar Wilde - The Picture of Dorian Gray.txt
72	Plato - The Republic.txt
73	Robert Stevenson - Treasure Island.txt
74	Rudyard Kipling - The Jungle Book.txt
75	T S Arthur - After the Strom.txt
76	T S Arthur - Alls for the Best.txt
77	T S Arthur - Cast Adrift.txt
78	T S Arthur - Danger or Wounded.txt
79	T S Arthur - Finger Posts .txt
80	T S Arthur - Friends and Neighbors.txt
81	T S Arthur - Grappling with the Monster.txt
82	T S Arthur - Hair Breadth Escapes.txt
83	T S Arthur - Heart Histories and Life Pictures.txt

Num.	Book's Name
84	T S Arthur - Home Lights and Shadows.txt
85	T S Arthur - Lessons in Life.txt
86	T S Arthur - Lezzy Glenn.txt
87	T S Arthur - Married Life.txt
88	T S Arthur - Ten Nights in a Bar Room.txt
89	T S Arthur - The Allen House.txt
90	T S Arthur - The Good Time Coming.txt
91	T S Arthur - The Hand But Not the Heart.txt
92	T S Arthur - The Home Mission.txt
93	T S Arthur - The Lights and Shadows of Real Life.txt
94	T S Arthur - The Wedding Guest.txt
95	T S Arthur - Trials and Confessions of a Housekeeper.txt
96	T S Arthur - Woman's Trials.txt
97	T S Arthur - Words for the Wise.txt
98	T S Arthur - Words of Cheer for the Tempted.txt
99	T S Arthur - Wreaths of Friendship.txt
100	Walt Whitman - Leaves of Grass.txt