Implementation of rabin-karp algorithm to determine the similarity of synoptic gospels

Anan Bahrul Khoir1*, Husnul Qodim2, Busro Busro3, Aldy Rialdy Atmadja4

1Faculty of Theology and Religious Studies, University of Groningen, Netherlands
2,3Department of Religious Studies, UIN Sunan Gunung Djati Bandung, Indonesia
4Department of Informatics, Sekolah Tinggi Teknologi Garut, Indonesia

*a.b.khoir@student.rug.nl

Abstract. The Synoptic Gospels have been a mystery for several hundred years in the study of the New Testament. One problem that stands out is the similarity of the content inside. Not only the order of the story but also the order of the words. To analyze this issue, this research used Rabin-Karp Algorithm to determine the similarities of the first three Gospels in the New Testament. The algorithm is perfect for searching multiple patterns in two documents. There are several stages in this calculation process: first, verses in each same categories will be split into x letters called substrings. Second, each letter in the substring will have a hashing process and produce a series of numbers. Third, a set of numbers combinations will look for unique numbers called fingerprints. Finally, comparing the fingerprints on document A and B before calculating the similarity using dice similarity coefficient. The results are both Matthew and Luke have a close relationship with Mark. As a conclusion, Matthew is closely related to Mark (59.13%) based on their contents compared to Luke (52.85%) or Mark against Luke (54.41%).

1. Introduction

The Synoptic Gospels consist of three Gospels in the New Testament which are the Gospel according to Matthew, Mark, and Luke. This term refers to a Greek word, “synopsis,” which literally means “seeing together.” The Gospel of John is not categorized into this word because, although it has some same points with the previous three books, it has considerable differences such as stories and sequences [1]–[3].

The similarities of the Synoptic Gospels have been becoming a problem among Christianity particularly the New Testament scholars. They have produced some hypotheses to address this problem. In addition, there are huge studies later that have attempted to explain this problem. However, a lot of them are just an assumption based on their subjective views to the Synoptic Gospels by comparing the texts manually. Some similar researches on the Synoptic Gospels was conducted by Choulakian that analyses the Gospels with statistical methods [4], and by Widdows and Cohen by using Semantic Vector Combinations [5]. However, we applied Rabin-Karp algorithm for some reasons.

Rabin-Karp algorithm is one of string-matching algorithm that can be used to determine the similarities between two documents [6]. Fernando, furthermore, writes in his essay that Rabin-Karp is better than Brute Force, Knuth-Morris-Pratt, and Boyer-Moore algorithm [7]. This algorithm is a perfect algorithm to look for multiple patterns in strings [8]–[10].
2. Theories

2.1 Text Mining

Text mining is a part of data mining which focuses on retrieving information from the text. There are many types of data that can be used for data mining such as relational data (RDBMS), semi-structured data such as XML, JSON, and unstructured data such as video, image, text on social media, and etc [11]. This kind of data has the same problems like high dimension, big amount of data, and changing the structure. In the text mining research, the forms of data become more complex and incomplete text structure, unclear, unstandardized meaning, and different language [12]. Information is usually obtained through forecasting patterns and trends in statistical pattern learning. Forecasting and statistical pattern have related to many topics and algorithms for analyzing text, spanning various communities, including information retrieval [13], natural language processing, data mining, machine learning, and biomedical sciences [14]. To generate the information, typical text mining processes include text clustering, text categorization, concept/entity extraction, granular taxonomy production, sentiment analysis [15], document conclusions [12], and entity relationship modeling, learning relationships between entities [16].

2.2 Rabin-Karp Algorithm

In 1987, Richard M. Karp and Michael O. Rabin created a string-matching algorithm which is currently called as Rabin-Karp algorithm. The main principle of this algorithm is searching a pattern in an inputed text [17]. Beside that, the technique is matching substring (m) in one document with a substring (n) in another document. Each substring is converted to binary numbers, calculated, and the result is called as hash value. If the hash values are matched, then performed the second matching between the characters, the pattern and the k length characters. If the letters are same, it will add the similarity points. If not, it will be performed a shift of substring as much as one digit to the right. An equation of (n - m) shows the number of shift happened [18], [19]. As a conclusion, something is called similar or same if both the hash value and the characters are same. The algorithm has the worst case with the complexity of time is O((n - m + 1)m) to O(nm + 1) [20], [21].

2.3 Dice Similarity Coefficient

This term is categorized as similarity coefficient, which is used to determine similarity between two documents, two queries, or a document and a query. In the Rabin-Karp, the texts will be splitted into many substrings. After that, every single character in the substring continuously will be changed to binary numbers by using function that is called as hashing. It will generate hash value for each document. Then, the algorithm will search identical hash values in the documents which is next called as fingerprint.

After finding unique hash value in both documents, then looking for hash values found on both fingerprints. The calculation of similarities between two documents based on the formula below.

\[
SC(X, Y) = \frac{|X \cap Y|}{|X| + |Y|} \tag{1}
\]

Where \( SC(X, Y) \) represents the level of similarity ranged from 0 to 1. The \( X \) represents the amount of fingerprints in document \( X \), and \( Y \) itself represents the amount of fingerprints in document \( Y \) [22].

3. Methodology

In this paper, we compared the passages in the particular pericopes which parallel to another Synoptic Gospel(s). Then, the passages are compared each other. After that, determining the similarities of the passages. Before reaching this step, the space and non-alfabethical characters will be removed from the given verses so that the characters will be coincided. The process is called as tokenizing in preprocessing step. Then, the processed texts will be romanized into latin to avoid some mistakes such as errors when transforming a letter into binary numbers. This research used UN/ELOT standard for transliterating Greek manuscript into latin [23]. In the Westcott-Hort Version, there is non-Greek characters, and will be erased while the texts inside will be included into calculation.

After the texts are clean and full of alfabethical characters, the next step is exploding the texts based on the given value of \( k \) in k-grams, that is 3; the basis is 11; and the modulo is 10007. The
merged characters we are called as a substring. For each character in that substring, it will be changed into binary numbers by using hashing function. This is the main technique of the Rabin-Karp. Until this step, the new strings are combination of digits. After that, finding a fingerprint or unique hash value from each string. Dice similarity coefficient will determine the identical hash values between the two strings.

4. Result and Discussion

Due to there are some versions on determining their pericopes, we defined ourself pericopes for research purposes by using some references such as Burton and Goodspeed’s book [24], a Blue Letter Bible’s site [25], and a module in a Biblestudy software, the Word, which is called as The Gospel Compared [26]. Futhermore, we found approximately 93 pericopes that parallel to other Synoptic Gospel(s) or 1904 of 2900 (65.65%) total verses compared in three Books including 713 verses from Matthew, 650 verses from Mark, and 531 from Luke.

Table 1 shows the references for simulation we chose from the pericope number 31, which is Mat 12:43-45 and Luk 11:24-26. There are 6 verses, 3 verses from Mat and 3 verses from Luk. However, in this case, only one verse from each book that will be determined, Mat 12:43 and Luk 11:24.

| Table 1. The process simulation of the references from pericope number 31 |
|---------------------------------|----------------|
| **Matthew**                     | **Luke**       |
| 12:43                           | 11:24          |
| 1. Original text from Westcott-Hort Version | otantoakathonptonpnevmaexelthiapotouanthetaopoudierchetaidianydrontoponzitounanapavsinkaimievriskontotelegeiypostrepsoeistonoikonnouthenexilthon |
| 2. Preprocessing stage by removing the space and non-alphabetical characters; after that, romanization process of Greek text | otantoakathonptonpnevmaexelthiapotouanthetaopoudierchetaidianydrontoponzitounanapavsinkaimievriskontotelegeiypostrepsoeistonoikonnouthenexilthon |
| 3. K-grams formation data | ota tan ant ntonaak oahaka kathitha harr art rito ton onppnppnenevevsmamamaeax exe xeelt lth thhiaiai apo pot ootououuan ant nth thrhropoppou ooud udi die ierercrcrrhe het etai aid ididiaia any nydydrroronontnto top oopononzni zit itotoouun una ana nap apa pav avsvsi sin ink nka aioiououuchchehevrvviririskskei |
| 4. Hashing process stage | 14804 15213 13047 14511 13327 13608 14804 15213 13063 14697 15354 14605 15354 14605 13011 14130 13117 15277 13011 14130 13117 15277 13765 15367 14753 14652 15181 15367 14753 14652 14863 15439 13628 15574 14357 12968 13628 15574 14357 12968 13642 15739 13525 14448 15285 13836 13525 14448 15285 13836 13884 13080 14889 14818 15374 14815 14889 14818 15374 14815 15334 13063 14690 15294 13949 15127 14690 15294 13949 15127 14774 14890 14818 15362 13356 13930 14818 15362 13356 13930 13574 14987 13224 13811 13594 15208 13224 13811 13594 15208 12992 13910 |
5. Fingerprint/Unique hash values from hashing data

| Fingerprint/Unique hash values from hashing data |
|-----------------------------------------------|
| 14804 15213 13047 14511 13327 13608 | 14804 15213 13063 14697 15369 15575 |
| 15354 14605 13011 14130 13117 15277 | 15354 15391 15017 13568 14928 15237 |
| 13765 13107 15181 15367 14753 14652 | 13765 13107 15181 14753 14652 14863 |
| 14863 14539 13628 15574 14357 12968 | 14863 14539 13628 14357 12968 13642 |
| 13642 15739 13525 14448 15285 13836 | 13642 13574 14987 13224 13811 13594 |
| 13884 13080 14889 14818 15374 14815 | 13884 14818 15374 14815 15334 14690 |
| 15334 13063 14690 15294 13949 15127 | 15334 13063 14690 15294 13949 15127 |
| 14774 14890 15362 13356 13930 13574 | 14774 14890 13356 13930 13574 14987 |
| 14987 13224 13811 13594 15208 12992 | 14987 13224 13811 13594 15208 12992 |
| 13910 13352 13882 13068 14741 15855 | 13910 13352 13882 13068 14741 15855 |
| 13465 15125 14757 14697 15369 14883 | 13465 15125 14757 14697 15369 14883 |
| 14763 16033 14092 14828 15464 14487 | 14763 16033 14092 14828 15464 14487 |
| 13044 14489 13066 14737 13150 15648 | 13044 14489 13066 14737 13150 15648 |
| 15180 14022 14584 14636 14693 13993 | 15180 14022 14584 14636 14693 13993 |
| 14817 15350 13813 13633 15637 15064 | 14817 15350 13813 13633 15637 15064 |
| 14077 15193 14163 (93) | 14077 15193 14163 (93) |

6. Fingerprint in both documents

| Fingerprint in both documents |
|--------------------------------|
| 14804 15213 13063 14697 15354 14605 13011 14130 13117 15277 13765 13107 13117 15277 13765 13107 15181 15367 |
| 14753 14652 14863 14539 13628 15574 14357 12968 13642 15739 13525 14448 15285 13836 13884 13080 14889 14818 15374 14815 15334 14690 15294 13949 15127 14774 14890 15362 13356 13930 13574 14987 13224 13811 13594 15208 12992 13910 13352 13882 13068 14741 15855 13465 15125 14757 15369 14883 14763 16033 14092 14828 15464 14487 13044 14489 13066 14737 13150 15648 15180 14022 14584 14119 13003 14043 13150 15648 15180 14022 14584 14119 |

7. Similarity level by using dice similarity coefficient

\[
S(Mat_{12:43,Luk_{11:24}}) = \frac{2 \times 82}{93 + 128} = \frac{164}{221} \\
= 0.7420814479638 \times 100\% \\
= 74.21\%
\]

Table 2. Security Message Experiment

| No | File Name | Original Size (Byte) | Message Length (Byte) | Result Size (Byte) |
|----|-----------|----------------------|-----------------------|-------------------|
From 93 parallels of the Synoptic Gospels defined, we found that 92 (98.92%) pericopes are found in Matthew, 83 (89.25%) and 74 (79.57%) pericopes are in Mark and Luke respectively (see figure 1). However, there are 63 of 93 pericopes (67.74%) in the all books of the Synoptic Gospels. Interestingly, almost 100% pericopes in Mark are located in Luke.

Figure 2 illustrates the rates of similarity in the Synoptic Gospels based on their parallel pericopes. Generally speaking, Matthew has the closest relationship with Mark (59.13%) compared to other book combinations while Luk-Mat has the lowest similarity average, which is nearly 53%.

As can be seen together on figure 2 that Matthew has considerable similarity to Mark rather than with Luke. It is almost 40 pericopes which are more than 60 per cent average similarity with Mat and/or Mrk, following by Mrk-Luk and Mat-Luk with approximately 18 pericopes that having
similarity level from 60 per cent. Interestingly, all the Synoptic Gospels have a great similarity between 50% and 60%. In other words, almost half of authors’ writings are copied on their works from other books.

As a conclusion, by using Rabin-Karp Algorithm, it was found that the Gospel according Matthew had a close relationship with Mark (59.13%) compared to Luke (52.85%), and then Mark closer to Luke (54.41%) though actually Luke is closer to Matthew if the total similarity is divided by the amount of defined pericopes, which is about 41.49 percent.

5. Conclusion

The Synoptic Gospels cause some problems in the New Testament subject. One of them is the literal problem or literal relationship between the Gospels. In this case, Rabin-Karp is pretty good to analysis this issue and strenghten some of the hypotheses that maybe solve this problem. The research found that almost all the pericopes in the Mark are found in both Matthew and Luke. Thus, it is highly likely that this paper is more inclined to support the hypotesis that put Mark as one of main source for Matthew’ and Luke’s writings or they called this phenomenon as Markan Priority. After that, we found that Matthew has a great literal relationship to Mark rather than Mark to Luke or Luke to Matthew. As a result, it can be ascertained that Matthew used Mark for his writings. This view represents Farrer-Goulder, Two Sources, Three Sources, and Four Sources Hypothesis. Then, Luke are closer based on its content to Matthew rather than to Mark. However, there are some parts in Mark is not contained in Matthew, and found in Luke. So, we state that Luke copied his writings from Matthew and Mark. To conclude this view, we made two options. First, Luke did not use Mark but other sources used literally by Mark that caused some passages in Mark are found in Luke but not in Matthew. This opinion falls to two hypotheses, Two and Four Sources Hypothesis. Second, Luke utilized Matthew and Mark altogether and other sources. Three Sources Hypothesis is in this position. Third, Matthew and Mark are references for Luke without other sources. The opinion is fully supported by Farrer-Goulder Hypothesis.

References

[1] Drewes B F, 1986 *Satu Injil Tiga Pekabar* Jakarta: Gunung Mulia.
[2] Duyverman M E, 1987 *Pengantar ke dalam Perjanjian Baru* 6th ed. Jakarta: Gunung Mulia.
[3] Keene M, 2010 *Alkitab: Sejarah, Proses Terbentuknya, dan Pengaruhnya* 5th ed. Yogyakarta: Kanisius.
[4] Choulakian V Kasparian S Miyake M Akama H Makoshi N and Nakagawa M, 2006 A Statistical Analysis of the Synoptic Gospels *Journees Int. d’Analyse Stat. des Donnees Textuelles* April 2014.
[5] Widdows D and Cohen T, Semantic Vector Combinations and the Synoptic Gospels.
[6] Nugroho E, 2011, Perancangan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma Rabin-Karp, Malang: Universitas Brawijaya.
[7] Fernando H, 2009 Perbandingan dan Pengujian Beberapa Algoritma Pencocokan String *Makal. IF3051 Strateg. Algoritm*.
[8] Cormen T H Leiserson C E Rivest R L and Stein C, 2003 *Introduction to Algorithms* USA: Massachusetts Institute of Technology.
[9] Siahaan A P U et al., 2018 Combination of Levenshtein Distance and Rabin-Karp to Improve the Combination of Levenshtein Distance and Rabin-Karp to Improve the Accuracy of Document Equivalence Level *Int. J. Eng. Technol.* 7, 2 p. 17–21.
[10] Noprisson H Susilo B and Ernawati, 2014 Implementasi Algoritma Rabin-Karp untuk Menentukan Keterkaitan Antarpublikasi Penelitian Dosen Tahun 2013 *J. Teknol. Inf.* 10, 1 p. 110–127.
[11] Allahyari M Trippe E D and Gutierrez J B, 2017 A Brief Survey of Text Mining: Classification , Clustering and Extraction Techniques.

[12] Slamet C Atmadja A R Maylawati D S Lestari R S Darmalaksana W and Ramdhani M A, 2018 Automated Text Summarization for Indonesian Article Using Vector Space Model in IOP Conference Series: Materials Science and Engineering288, 1.

[13] Gerhana Y A Atmadja A R Zulfikar W B and Ashanti N, 2017 The implementation of K-nearest neighbor algorithm in case-based reasoning model for forming automatic answer identity and searching answer similarity of algorithm case in 2017 5th International Conference on Cyber and IT Service Management (CITSM) p. 1–5.

[14] Krallinger M Rabal O Lourenço A Oyarzabal J and Valencia A, 2017 Information retrieval and text mining technologies for chemistry Chem. Rev.117, 12 p. 7673–7761.

[15] Buslim N Busman B Sinatrya N S and Kania T S, 2018 Analisa Sentimen Menggunakan Data Twitter, Flume, Hive Pada Hadoop dan Java Untuk Deteksi Kemacetan di Jakarta J. Online Inform.3, 1 p. 1.

[16] Agnihotri D Verma K and Tripathi P, 2014 Pattern and cluster mining on text data Proc. - 2014 4th Int. Conf. Commun. Syst. Netw. Technol. CSNT 2014 April p. 428–432.

[17] Rawan A A, 2011 An Algorithm for String Searching Based on Brute-Force Algorithm Int. J. Comput. Sci. Netw. Secur.11, 7 p. 24–27.

[18] Firdaus H B, 2003 Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp J. Ilmu Komput. dan Teknol. Inf.3, 2 p. 1–5.

[19] Jain P A S and Rao A L N, 2012 A Relative Study of Pattern Matching Algorithms Int. J. Comput. Technol.1, 2 p. 14.

[20] Siahaan A P U Mesran Rahim R and Siregar D, 2017 K-Gram As A Determinant Of Plagiarism Level In Rabin-Karp Algorithm Int. J. Sci. Technol. Res.6, 7 p. 350–353.

[21] Gonnet G H and Baeza-Yates R A, 1990 An Analysis of the Karp-Rabin String Matching Algorithm Inf. Process. Lett. p. 271–274.

[22] Thada V and Jaglan V, 2013 Comparison of Jaccard , Dice , Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm Int. J. Innov. Eng. Technol.2, 4 p. 202–205.

[23] Holloway S, 2018, Greek Romanization Table, SCRIPTSOURCE. .

[24] Burton E D W and Goodspeed E J, 1928 A Harmony of The Synoptic Gospels in Greek 5th ed. Chicago: The University of Chicago Press.

[25] Unknown, 2018, Harmony of the Gospels. .

[26] Stergiou C, 2003, theWord. .