Optimization of Associative Knowledge Graph Using TF-IDF Based Ranking Score

Hyun-Jin Kim 1, Ji-Won Baek 2 and Kyungyong Chung 1,*

1 Division of Computer Science and Engineering, Kyonggi University, Suwon 16227, Gyeonggi, Korea; khyunjin1993@gmail.com
2 Department of Computer Science, Kyonggi University, Suwon 16227, Gyeonggi, Korea; hy1233hh@naver.com
* Correspondence: kychung@kgu.ac.kr

Received: 15 May 2020; Accepted: 29 June 2020; Published: 2 July 2020

Abstract: This study proposes the optimization method of the associative knowledge graph using TF-IDF based ranking scores. The proposed method calculates TF-IDF weights in all documents and generates term ranking. Based on the terms with high scores from TF-IDF based ranking, optimized transactions are generated. News data are first collected through crawling and then are converted into a corpus through preprocessing. Unnecessary data are removed through preprocessing including lowercase conversion, removal of punctuation marks and stop words. In the document term matrix, words are extracted and then transactions are generated. In the data cleaning process, the Apriori algorithm is applied to generate association rules and make a knowledge graph. To optimize the generated knowledge graph, the proposed method utilizes TF-IDF based ranking scores to remove terms with low scores and recreate transactions. Based on the result, the association rule algorithm is applied to create an optimized knowledge model. The performance is evaluated in rule generation speed and usefulness of association rules. The association rule generation speed of the proposed method is about 22 seconds faster. And the lift value of the proposed method for usefulness is about 0.43 to 2.51 higher than that of each one of conventional association rule algorithms.

Keywords: TF-IDF; association rule; apriori; FP-tree; associative knowledge graph

1. Introduction

These days, a massive amount of information based on news, social network services, and media is generated in real-time. As a result, structured and unstructured big data, which are hard to be collected and processed in conventional ways or tools, are created. In the era of big data, data mining has been researched for analysis and predictive modeling and has continuously been developed [1,2]. In particular, data mining is actively researched in information search areas, including emerging risk, healthcare, stress management, and real-time traffic information [3,4]. Data is consisted with text, image, number, and category, including continuous and discrete types of information that are semantically associated with each other. Therefore, it is necessary to apply a method of obtaining associated information through exploratory data analysis [5,6]. In short, it is to create a knowledge graph through association rule mining and easily look at significant information in massive data. One of the techniques for data mining, which is the analysis of association rules, is to discover knowledge by extracting new rules or patterns that are useful from large-scale data. It operates based on transaction data format to extract association rules. In order to compose a transaction, one transaction row composed of a transaction ID, and an itemset is generated for each document that is extracted to keywords. It is possible to construct a set of transactions consisting of multiple documents through this process. Since the association rule compares and extracts the itemset between each transaction, this
process is essential to extract the association rule. Rules between transactions are acquired through the association rules algorithm. Rules acquired inevitably increase as the number of transactions and itemsets increases. It is difficult to grasp the meaning of these large rules in the form of texts at a glance. Therefore, visualization is necessary to efficiently understand and provide the meaning and information of the derived rules data. Therefore, visualization is necessary to efficiently understand and provide the meaning and information of the acquired rules data. For visualization, graphs are used. The graph expresses data as nodes and edges so that information can be intuitively understood. As an algorithm to discover keywords from a large text, there is a TF-IDF algorithm which is generally used in text mining. The TF-IDF algorithm is usually applied to information search and text mining-based modeling. In a document or corpus, a weight value can be applied to each word in a DTM (document term matrix) or word2vec [7]. TF-IDF weights are used to extract a similarity level of documents, an importance level of a particular word, and keywords, and to determine the ranking of search results [8]. Accordingly, by removing unnecessary words in a document, it is possible to improve the result and performance of the analysis.

When it comes to association rules in data mining, Apriori is typically used to create meaningful rules. The Apriori algorithm finds associations between items in multiple transactions and generates rules [9]. The generated association rules are visualized on the basis of FP-tree. The algorithm makes data as transactions and applies edge and vertex to association rules so as to visualize the rules in a knowledge graph [10,11]. In addition, it can effectively prune the exponential search space of candidate items sets and can expand it efficiently by considering their frequency patterns. FP-tree calculates the Support Count of each term in all transaction data, sorts them in order of the support count, and makes a Frequent Item Header Table. Based on the Frequent Item Header Table, each factor in transactions resorts in order of the support count of words. A tree is created in order of resorted transaction factor. The factors in the Frequent Item Header Table are linked with each other in the type of Linked List, and therefore an FP-Tree is created. With the use of Linked List, the created FP-Tree is moved from the word with low frequency in a bottom-up way, and then frequent itemsets are generated. With the use of frequent itemsets, the association rules of words can be generated [12].

With the increases in the number of transactions and the number of item sets, association rule algorithm costs high in generating association rules [13]. For this reason, it is necessary to improve the search speed of association rules in order to quickly process massive stream data created in real-time. Data mining utilizes association rules so as to create a graph and expand a knowledge base. In an association rule model, no more than two equal items are included in one transaction. For this reason, if a document is designed with each transaction, the duplication of a word is not counted. To solve the problem, this study proposes the optimized method of the associative knowledge graph using TF-IDF based ranking scores. The proposed method utilizes the keywords extracted from the text data related to traffic accident topics so as to design transactions and optimize a knowledge graph. It applies TF-IDF to calculate the weight of an item in a transaction. In addition, it calculates TF-IDF weights for a document and prioritizes words on the basis of weights. Based on the term ranking, the optimized associated-terms graph is generated. Accordingly, unnecessary words are removed in advance, and consequently it is possible to visualize more cleaned association rules in the graph than in a conventional pruning based graph. In addition, it is possible to obtain knowledge in the graph of association rules with high Lift values. Therefore, unpredicted rules, which are not general, are generated. The method is efficient at processing and extracting latent useful information on unstructured data in real-time.

This study is comprised of as follows: Chapter 2 explains the method of creating an association rule model and the TF-IDF based classification method; Chapter 3 describes the proposed optimization method of the associative knowledge graph using the TF-IDF Rank Table; Chapter 4 describes the performance evaluation in comparison of rule generation speed and usefulness with the uses of Apriori algorithm and FP-Tree algorithm and with the application of TF-IDF Rank Table; Chapter 5 describes a conclusion.
2. Related Work

2.1. Extracting Knowledge Using Associative Rules

An association rule algorithm is used to find the rule as to how an item set frequently occurs [9]. It is applied to the business target marketing to find the probability that a customer who purchased the item A will purchase the item B, on the basis of transaction data [14]. The algorithm is used in various areas including marketing, healthcare, and information search. For example, it is possible to obtain information according to association rules of chronic diseases or association rules of other diseases in the healthcare area [15]. For the association rule algorithm, data sets are designed in a transaction type. Table 1 shows a transaction consists of Transaction ID and Itemset.

Table 1. Transaction consists of Transaction ID and Itemset.

| Transaction ID | Itemset       |
|----------------|---------------|
| 1              | A, B, C, D    |
| 2              | B, C, E       |
| 3              | A, B, E       |
| 4              | B, C, D, E    |
| 5              | A, B, D, E    |
| ...            | ...           |

For example, as shown in Table 1, a customer whose Transaction ID is ‘1’ can be interpreted to purchase items A, B, C, and D in the market. When association rules are expressed with the use of the transaction data in Table 1, the association rules for the case that a customer who purchased the item A is likely to purchase the item B are generated. Table 2 presents Apriori association data. Lhs means Left-hands-side, and Rhs represents Right-hands-side. Lhs => Rhs means that a left item is associated with the right item.

Table 2. Apriori Association Data.

| Lhs => Rhs | Support | Confidence | Count |
|------------|---------|------------|-------|
| A => B     | 3/5     | 3/3        | 3     |
| B => A     | 3/5     | 3/5        | 3     |
| B => C     | 3/5     | 3/5        | 3     |
| C => B     | 3/5     | 3/3        | 3     |
| ...        | ...     | ...        | ...   |

In the association rules of Table 2, the measures used to find the rules of items are support, confidence, and lift [16,17].

J. Silva et al. [13] proposed the method of classifying customers in detail through the extraction of association rules. It utilizes an association rule algorithm in order to find associations between various types of customers and products and establish a marketing strategy. These association rules can be applied not only in the field of marketing but also in terms of text mining. For example, text data is extracted from a topic related to a traffic accident. Traffic accident-related text is collected through text data in the traffic accident-related section of the press release provided by the Ministry of Land, Infrastructure, and Transport [18]. Words are extracted by preprocessing each press release of the traffic accident domain. Transactions are generated through the extracted word. Text mining can be applied based on the transaction consist of these press releases. Figure 1 shows the transaction and association rule data based on the text mining domain of Ministry of Land, Infrastructure and Transport’s press release. The first news in Figure 1 shows the word, such as accident, traffic, vehicle, and crackdown exists, and association rules generated through a word, such as ‘accident’, appears when the word ‘traffic’ appears.
Accordingly, new information can be gathered through the rules of high weight for rare entities. The proposed method is inferred and weighted based on the initial embedding of entities and relationships. O. Emebo et al. [22] proposed a method of managing and identifying implicit requirements using common sense knowledge, ontology, and text mining. In the proposed method, common sense knowledge automatically identifies and manages implicit requirements. It is an automated tool for identifying sources of implicit requirements and managing them within an organization. As a result, potential requirements can be discovered to reduce the risk of language placement and to detect and correct errors based on the classification results of common sense about objects, the common sense of object relationships, and common sense of interaction. Accordingly, new information can be gathered through the rules of high weight for rare entities.

Knowledge graphs can be generated using the results of the association rules. The knowledge graph means related information and knowledge are linked and expressed. It can also infer potential information and knowledge. Accordingly, when a knowledge graph is generated as a result of an association rule, it is possible to intuitively grasp information and knowledge of related rules, and deduce meaningful information and knowledge. F. Zhao et al. [20] proposed structure-augmented knowledge graph embedding for sparse data with rule learning. The creation of rules of the proposed method is inferred and weighted based on the initial embedding of entities and relationships. Accordingly, new information can be gathered through the rules of high weight for rare entities.

N. Tandon et al. [21] proposed Commonsense Knowledge in Machine Intelligence. The proposed method is a method used to solve the problem of language placement and to detect and correct errors based on the classification results of common sense about objects, the common sense of object relationships, and common sense of interaction. Accordingly, common sense and knowledge can be acquired from the text. In addition, O. Emembo et al. [22] proposed a method of managing and identifying implicit requirements using common sense knowledge, ontology, and text mining. In the proposed method, common sense knowledge automatically identifies and manages implicit requirements. It is an automated tool for identifying sources of implicit requirements and managing them within an organization.
organization. As a result, potential requirements can be discovered to reduce the risk and cost of software development.

2.2. Word Classification Using Term Frequency-Inverse Document Frequency

TF-IDF (Term Frequency-Inverse Document Frequency) is a method of calculating weights of terms in information search and text mining [23,24]. It is used to judge how important a term is in a document. Formula (1) presents TF-IDF:

\[ TF - IDF(t, d, D) = tf(t, d) \times idf(t, D) \] (1)

TF that stands for term frequency represents the frequency of the particular term \( t \) in a document. In short, it means how many times a particular term appears in one document. A TF value has infinite divergence with the rise in the frequency of a particular word. IDF (Inverse Document Frequency) means the inverse value of the word that appears commonly in multiple documents. Formula (2) presents DF (Document Frequency). In Formula (2), \( t \) means the number of documents that include the term \( t \), and \( D \) means the total number of documents. DF represents the frequency of a particular term in documents, so that \( t \) is divided by \( D \):

\[ DF(t, D) = \frac{t(n)}{D(n)} \] (2)

The reason why a DF value is calculated is that if the number of documents that include a particular term increases among all documents, it means the word lacks discrimination. To lower weights of the frequently found words (‘a’, ‘an’, ‘the’, etc.), it is necessary to calculate the value of DF. The higher the value of DF is, the total result value has to be lower, thus, lower the weights of uninformative words. Accordingly, the inverse value of DF is calculated, which is represented as IDF. Nevertheless, since a value of IDF is used, it tends to increase exponentially unless a particular term appears. For example, if the term \( t \) appears nothing in ten documents, the df value is 10/0, which is an infinite value. To solve the problem, it is necessary to apply the logarithm of the inverse value of DF. Formula (3) is used to calculate the logarithm of IDF:

\[ IDF(t, D) = \log \left( \frac{D(n)}{t(n)} \right) \] (3)

A value of TF-IDF is calculated in the way of multiplying a value of TF which is the frequency of the term \( t \) in a particular document, by a value of IDF, the inverse value of the frequency of the term \( t \) in all documents. As a result, a TF-IDF value is large if the frequency of a particular term in a particular document is high and the number of documents that contain that particular term in all documents is low. In this way, it is possible to decrease the weights of words that contains in most documents and to increase the weights of the term with high importance in a particular document. As a result, important words are selected [25,26]. J. H. Paik [23] proposed the TF-IDF weighing method to determine the effective ranking, which utilizes frequency and normalization to find keywords in a document. The proposed method adaptively applies the combination of weights of components according to the length of a query. The longer the query length is, the more the method is effective, and the more it extracts significant and consistent words. Z. Yun-tao et al. [24] proposed the text classification method using TF-IDF. It utilizes confidence and support in order to improve the recall and precision of document classification. It can solve the problem that a document duplicates in various categories and find a proper category. From the news stream data offered by MBC (Munhwa Broadcasting Corporation, South Korea), 1700 text data related to traffic accident topics [19] were collected and are converted into a corpus. Figure 3 shows the weights of the words calculated with the use of TF-IDF in the news stream data. Based on the calculated TF-IDF weights of the words in each collected documents, the weights of words in all documents are totaled. Terms are sorted in descending order of weight, and therefore a
TF-IDF Rank Table is generated. Words are visualized in the graph in order of score. The x-axis means the words extracted from traffic accident topic text, and the y-axis means the TF-IDF weights.

In Figure 3, the term ‘Traffic’ has the highest TF-IDF weight score, 35 in terms of all documents. The term ‘time’ has the lowest weight score, which is about 10. Based on the TF-IDF Rank Table, transactions are redesigned in order to find efficient association rules. The words with TF-IDF Rank scores, which are not in the top 20% scores of the words extracted from traffic accident topic text, were deleted. A. Rozeva et al. [27] proposed a method and algorithm for evaluating the semantic similarity of text. The proposed method was further analyzed through a vector-based knowledge search approach. This takes into account the mathematical background of the potential meaning analysis by using the weight and similarity calculation of TF-IDF to derive the meaning of words from the text. Accordingly, it is an algorithm that not only calculates similarity but also obtains a vector representation and potential meaning of words in a reduced multidimensional space. Therefore, it can be used to evaluate text at the syntax and semantic level, reduce the dimension of the document vector space, and acquire more meaning of the text.

3. Optimization of Associative Knowledge Graph using the Term Frequency-Inverse Document Frequency based Ranking Score

In order to find significant information in massive data generated in real-time, it is necessary to improve the speed and usefulness of the association rule algorithm. This study proposes the method of optimizing the associated-knowledge graph using TF-IDF based ranking scores. The knowledge graphs made with conventional association rules include information of words with low importance, so that information offering efficiency is low. To solve the problem, the proposed method removes words with low importance and creates a knowledge graph by using TF-IDF based ranking scores. It consists of data collection and preprocessing step, mining-based associated-words knowledge graph extraction step, and TF-IDF based association-knowledge graph optimization step. Figure 4 shows the optimization process of association graph using TF-IDF based ranking scores.
In the first preprocessing step, news about traffic accident topics and traffic safety topics are collected in real-time through crawling and then are converted into a corpus. Unnecessary data are removed from the news corpus, words are extracted in morphological analysis, and transactions are designed. In the second step, an associated-words knowledge graph is extracted with the use of mining. With transactions, a frequent item header table is generated on the basis of support and confidence. In the frequent item header table, association rules are discovered, knowledge is extracted, and a graph is generated. In the last step, the ranking of the words extracted in the first step is determined with the use of TF-IDF. Words with low ranking scores are judged to be less important and thereby are removed, and transactions are redesigned. Through association rules, significant knowledge is extracted, and a graph is generated. With the generated knowledge graph, it is possible to make the knowledge base of the traffic accident and safety and to predict an emerging risk.

3.1. Data Collection and Preprocessing

Information is classified for each topic on the web page that provides news information. Therefore, it is possible to easily access the topic of the necessary information. However, it is difficult to collect a large amount of data. To solve this problem, web crawling is used [28]. To crawl web data, Python’s BeautifulSoup4 [29] package is used. First, news release from the Ministry of Land, Infrastructure and Transport [18] web pages are used to gather information. It is classified as a topic of National City, residential land, construction, transportation logistics, aviation, and road railway. Therefore, the listing page of related articles from the Transportation Logistics topic is retrieved. Traffic logistics topics include content that contains various traffic-related information such as accidents, autonomous driving, and traffic regulations. Accordingly, the pattern on the URL address of the article list page is analyzed, and multiple list pages are accessed. It finds the URL address of all connected articles. Then, the HTML file of the address of the page containing the body of the received news is fetched. A separate parsing process is required to get the main body of the article required from the HTML file. Therefore, the class name of the tag corresponding to the article content of HTML is found and parsed to collect the body text data related to traffic. For data collection, the text in the div tag with the class name of ‘bd_view_cont’ on the relevant page was extracted. From social data which are collected,
news about the traffic accident and safety topics are collected through crawling and are converted into a corpus, in order for a knowledge graph [30]. The news corpus is comprised of news generation date, a category, a news publication company, a title and text, and a uniform resource locator (URL). The collected corpus is preprocessed for the improved quality of the analysis. In the preprocessing step, lowercase conversion and removal of punctuation marks and stop words are performed, in order to apply association rules to the news corpus. Accordingly, the transaction is constructed in the document term matrix. Figure 5 shows the preprocessing of news corpus.

As shown in Figure 5, unnecessary data of news generation date, news Publication Company, and URL are deleted in the preprocessing step [31]. The outcome of the preprocessing step is the corpus consisting of category, news title, and text. With the words extracted from the news corpus, transactions are generated. The words extracted from news titles and texts are analyzed morphologically. In the morphological analysis, punctuation marks, numbers, special characters, and stop words are removed from the news corpus, and only terms are extracted [32]. Since stop words as index words are meaningless, they are removed from the converted vector matrix. Accordingly, with the use of the list of noun words extracted in each news document, the words are properly converted into transactions. For transaction labeling, an ID value is assigned to each transaction row. Table 3 shows the transaction data after preprocessing. The table consists of transaction ID and item, and items are a list of words.

Table 3. Transaction Data after Preprocessing.

| Transaction ID | Itemset |
|---------------|---------|
| 1             | conflict, improve, initiate, discussion, view, open, common, relation, nation, attitude, year, content, history, Olympic, recognition, attention, … |
| 2             | director, hold, game, stadium, gold medal, commemoration, north, south, attainment, battle, victory, national, team, Laos, start, football, India, … |
| 3             | initiation, public, relations, related, content, effort, Tokyo, seeking, problem, missing, reflection, reparation, court, analysis, case, health, … |
| 4             | director, hold, game, stadium, gold medal, commemoration, north, south, attainment, battle, victory, national, team, Laos, music, performance, … |
| 5             | health, studying, performance, Netherlands, ability, broadcasting, growth, start, deterioration, Jansons, orchestra, music, bus, traffic, … |
| 6             | possible, gas, emphasis, attack, science, global, finance, effort, news, achievement, response, city, car, expressway, licence, store, guest, … |
| 7             | price, game, economy, customer, soaring, record, reporter, dollar, help, marketing, store, sales, USA, press, analysis, time, shopping, stress, … |

For example, in Table 3, when a Transaction ID is 1, it means that items such as conflict, improve, initiate, discussion, view, open, common, relation, nation, attitude, year, and content are included in the transaction. The transaction capacity increases as the number of words increases. A data set including the item $k$ can latently generate $2^k-1$ frequency item sets, and the search space becomes large.
exponentially. Therefore, in order to reduce the calculation complexity of frequency item sets, it is necessary to decrease the frequency of comparison or lower the number of candidate itemsets.

3.2. Associative Knowledge-Graph Extraction Using Data Mining

For the extraction of associated-words knowledge, mining is applied to discover association rules [32–34]. Association rules are visualized in the graph [35,36]. In terms of the analysis on the association of words, the association rule algorithm is different from the algorithm in general prediction modeling. A general algorithm in predictive modeling uses the explanatory variables \( x_1 \) and \( x_2 \) in \( y = x_1 + x_2 \) in order to predict the value of the response variable \( y \). But when generating association rules from text data, there is uncertainty about how to set the explanatory and response variable to determine the association between words. For this reason, the association rule algorithm does not set a particular response variable but finds the association of words on the basis of support and confidence.

Based on the generated transactions, the association rule algorithm is applied to make a frequent item header table in which the minimum support is 0.06, and confidence is over 0.1. In the frequent item header table, the minimum support and confidence values are determined in between 0 and 1. The setting basis of the minimum support and confidence is to remove unnecessary data sets from association rule data. With the values enough to generate associated words, the repeated test is conducted. As a result, the optimal values of the minimum support and confidence are determined. A level of association of words is analyzed so that associated words are saved. Table 4 presents a Frequent Item Header Table that consists of Item, Support, and Count. The count means the frequency of words that meet the minimum support.

| Item       | Support  | Count |
|------------|----------|-------|
| news       | 0.5370112| 769   |
| local      | 0.3163408| 453   |
| time       | 0.2688547| 385   |
| relation   | 0.2632682| 377   |
| communication | 0.2108939| 302   |
| police     | 0.2094972| 300   |
| report     | 0.2067039| 296   |
| government | 0.2067039| 296   |
| reporter   | 0.1927374| 276   |
| news, local| 0.1857542| 266   |
| America    | 0.1780726| 255   |
| middle     | 0.1745810| 250   |
| Trump      | 0.1717877| 246   |
| occur      | 0.1710894| 245   |
| Seoul      | 0.1682961| 241   |
| traffic    | 0.1658888| 240   |
| pot-hole   | 0.1658763| 238   |
| bridge     | 0.1648232| 234   |
| express    | 0.1623455| 223   |
| congestion | 0.1593023| 208   |
| fog        | 0.1364892| 186   |
| ...        | ...      | ...   |

For instance, the term ‘news’ in the first row of the Table 4 meets support 0.5370112, and appears 769 times in a document. Table 5 shows the association rules based on the Apriori algorithm. It consists of rules, support, confidence, and count. The count represents the frequency of words used in rules. The support means a probability that Lhs and Rhs appear at the same time so that equal support and count values appear. When the Lhs and Rhs of words used for generating association rules are
switched, the values of support and count are equal, but the value of confidence is different. That is because the confidence is the probability of Rhs is present when Lhs is given in the transaction.

| Lhs => Rhs     | Support   | Confidence | Count |
|----------------|-----------|------------|-------|
| local => news  | 0.18575   | 0.58719    | 266   |
| news => local  | 0.18575   | 0.34590    | 266   |
| time => local  | 0.16829   | 0.62597    | 241   |
| local => time  | 0.16829   | 0.53200    | 241   |
| time => news   | 0.16689   | 0.62077    | 239   |
| news => time   | 0.16689   | 0.31079    | 239   |
| relation => news| 0.14315  | 0.54376    | 205   |
| news => relation| 0.14315 | 0.26658    | 205   |
| seoul => news  | 0.12220   | 0.72614    | 175   |
| news => seoul  | 0.12220   | 0.22756    | 175   |
| police => news | 0.11592   | 0.55333    | 166   |
| news => police | 0.11592   | 0.21586    | 166   |
| communication => news| 0.11522 | 0.54635    | 165   |
| news => communication | 0.11522 | 0.21456    | 165   |
| reporter => news| 0.11451  | 0.58320    | 164   |
| news => reporter| 0.11451 | 0.49093    | 164   |
| traffic => speed| 0.11449  | 0.61239    | 160   |
| speed => traffic| 0.11449 | 0.41233    | 160   |
| distance => time| 0.11389  | 0.59234    | 159   |
| time => distance| 0.11389 | 0.31456    | 159   |
| gas-station => explosion| 0.10998 | 0.56133    | 155   |
| explosion => gas-station| 0.10998 | 0.32331    | 155   |
| expressway => speed| 0.10967 | 0.65333    | 152   |
| speed => expressway| 0.10967 | 0.45467    | 152   |

For example, in the first row of Table 5, when a rule is generated with the Lhs ‘local’ and the Rhs ‘news’, the value of support is 0.18575, the value of confidence is 0.58719, and the count is 266. In the second row, when a rule is generated with the Lhs ‘news’ and the Rhs ‘local’, the value of support is 0.18575, the value of confidence is 0.34590, and the count is 266. At this time, the first row has the same support value and count as the second row, but their confidence values are different because Lhs and Rhs are changed.

Based on the expression ‘X => Y, which represents the association of the words X and Y on the basis of association rules, edge and vertex are generated. X => Y means the association that the word Y can appear when the word X appears. Based on that, it is possible to create news-based words association graph. Figure 6 represents the knowledge graph of association words. The result shows the associated-words knowledge graph on the basis of the association rule algorithm. A knowledge graph is extracted through a directed graph. A node represents a term. An edge represents the direction of the graph. A direction represents the association of words. The upper graph of Figure 6 shows the number of the generated rules visualized, which is about 2400. In addition, the lower part of the figure shows the related rules by visualizing traffic-related subgraphs out of 2400 rules. The reason for visualizing the subgraph is that there is a problem—rules generated in the entire graph are not easily visible, so a part of the overall association rule graph is enlarged and visualized. The bidirectional arrow in the generated rules means that both words have their association rules. Therefore, although the number of rules is 2400, the generated knowledge could be wider. Since a particular word has associations with multiple words, the rules and unnecessary knowledge caused by word duplication are included. Additionally, there is a limitation in obtaining knowledge due to the low efficiency of visualization.
by word duplication are included. Additionally, there is a limitation in obtaining knowledge due to the low efficiency of visualization.

3.3. Associative Knowledge Graph Optimization using Term Frequency-Inverse Document Frequency

The generated knowledge graph includes a lot of unnecessary and less important words [35]. In particular, the Apriori algorithm fails to consider the term frequency for one term and sets the frequency of a term as '1' even if the term appears multiple times in one document [9,37]. For this
reason, it is necessary to optimize an associated-words graph and calculate a value of TF-IDF for the term extracted in order to create a more cleaned model at high speed. For the application of a TF-IDF value, the document term matrix for the words of 1700 traffic safety news data is generated. The matrix with the calculated TF-IDF values is generated. Since a document is expressed in vector, it is possible to measure the distance or similarity of documents. In the matrix with the TF-IDF calculated values, TF-IDF values of all documents per term are totaled, and TF-IDF based ranking is given [38,39]. Such a way helps to solve the problem that the interaction of words fails to be expressed in a document term matrix. Table 6 shows the result of the TF-IDF ranking scores.

Table 6. Results of the TF-IDF Rank Score.

| TF-IDF Rank | Word     | TF-IDF Rank Score |
|-------------|----------|------------------|
| 1           | traffic  | 35.206371        |
| 2           | depression | 31.395520      |
| 3           | video    | 21.520154        |
| 4           | police   | 21.021509        |
| 5           | stress   | 19.089421        |
| 6           | accident | 18.581047        |
| 7           | injury   | 18.428506        |
| 8           | organization | 18.072522   |
| 9           | bus      | 18.026231        |
| 10          | photo    | 17.912217        |
| 11          | expressway | 17.878551      |
| 12          | president | 16.859654       |
| 13          | psychotherapy | 16.203685  |

Based on the TF-IDF ranking scores, the top 20% of words are used for the comparison with an associative knowledge graph. If the word extracted from news data is not in the top 20%, it is removed. With the use of the transactions generated in the preprocessing step, association rules are generated, and finally an associated-words knowledge graph is made. Figure 7 shows the optimization process of the TF-IDF based associative knowledge graph.

Figure 7. Optimization Process of TF-IDF based Associative Knowledge Graph.

In the first stage, data are collected and preprocessed in order to make a TF-IDF Rank Table. For the calculation of TF-IDF weights, words are extracted from news corpus data. The words with TF-IDF weights are extracted from each news document. After the application of all news documents, the weights of words are totaled. As a result of the addition, a TF-IDF Rank Table is generated. In the second stage, data are processed in order for the creation of association rules. Unnecessary data
columns like URL are deleted from the news corpus. Words data are extracted in morphological analysis. If the extracted word data is in the low rank of the TF-IDF Rank Table, it is removed. In this way, the optimized transactions are generated on the basis of the TF-IDF Rank Table. The Algorithm 1 shows optimized transaction generation algorithm. The input is the news stream data, and the output is the optimized transaction.

**Algorithm 1 Optimized Transaction Generation Algorithm**

Input : News Stream Data → NewsData[k]
Output : Optimized Transaction → OptimizedTransaction[l]

**Step 1: TF-IDF Ranking Table Generation**

RankTable ← NULL // TF-IDF RankTable

for i is number of NewsData[k] do
    TF_Nouns ← extractNoun(NewsData[i])
    TF_Corpus ← Corpus(TF_Nouns) // make TF_Nouns list to Corpus data
    removeUnnecessaryData(TF_Corpus)
    // Create DocumentTermMatrix from TF_Corpus to calculate TF-IDF Weight
    TF_DTM ← DocumentTermMatrix(TF_Corpus)
endfor

TF_DTM ← WeightTFIDF(TF_DTM) // Add TF-IDF Weights per Column to sum up TF-IDF of single News Data

for i is number of column in TF_DTM do
    RankTable[i] ← Sum(TF_DTM[i])
endfor // Sort RankTable by Score

RankTable ← Sort(RankTable, RankTable.TotalColumnScore)

**Step 2: TF-IDF based Optimized Transaction Creation**

for i is number of NewsData[k] do
    Nouns[i] ← extractNoun(NewsData[i]) // extract words from a row of news data
    for j is number of words extracted in Nouns[i] do
        for k is number of words in top 20% of RankTable do
            // remove word if it cannot be found in the top 20% RankTable
            if (Nouns[j] ! = RankTable[k])
                Remove(Nouns[j])
        endfor
    endfor
    NewsCorpus ← Corpus(Nouns[i])
    removeUnnecessaryData(NewsCorpus)
    OptimizedTransaction[i] ← Transaction(NewsCorpus)
endfor

return OptimizedTransaction

In the third stage, the generated transactions are applied to the association rule algorithm. At this time, pruning is performed in order to extract cleaned association rules. In the last stage, based on the association rules, data are visualized. From the simple association rules, latent associations and knowledge are extracted. That is because the model for effectively extracting and observing data meanings is needed. Figure 8 shows the rule graph after visualization.
A graph can present the relation between objects. A graph is classified into the one-way graph (\{Mental\} => \{Urban\}) and bidirected graph (\{Stress\} => \{Urban\} and \{Urban\} => \{Stress\}). In the graph of Figure 8, all edges have their weight. If weight is given to an edge, it is possible to find a level of the relationship between nodes. For this reason, a graph can be used efficiently [35,40]. This study uses a lift value of association rule as the weight of a graph edge. In addition, for the generation of association rules based on keywords, a TF-IDF weight presenting word importance is used for the vertex. For instance, the vertex value of stress is 19.07. The weight of \{stress\} => \{urban\} edge is 5.66. Figure 9 shows the optimized associative knowledge graph based on TF-IDF. It is the modeling of word topics for extracting information on health from the news corpus of traffic accidents. It is possible to expand to the knowledge base for predicting emerging health risks in the relation between a traffic accident and health topics.

For example, in the optimized associative knowledge graph of Figure 9, the weight of \{Traffic\} => \{Congestion\} is 1.877. Having the word ‘congestion’ in a traffic circumstance is a general association rule so that the value of lift is low. On the contrary, the lift value of \{Traffic\} => \{Depression\} is 7.542. The appearance of ‘depression’ in a traffic circumstance is not general. In short, it is an
association rule of providing new information. For this reason, the lift weight is high. Accordingly, the TF-IDF based knowledge graph generates more optimized knowledge than a conventional knowledge graph. Figure 10 presents non-optimized associative knowledge graph. The graph made in the way of setting the number of nodes only with the use of pruning in a conventional association rule graph.

As shown in Figure 10, most nodes are empty root nodes, such as {} => {tunnel}, presenting association rules. A relation is not displayed with the use of the association rule of two words so that it is hard to find new information. Additionally, as presented in {Press, traffic} => {expressway}, rules without meanings and specialty are generated. In the circumstance, it is difficult to find new and significant information. Since unnecessary words are removed in the optimized graph made with TF-IDF, it is possible to find association rules of keywords and information easily.

4. Experimental Results

The hardware and operating system for implementing the optimized associative knowledge graph proposed in this study are MacOS Catalina 10.15.3 (O/S), i7-7820HQ 2.9GHz(3.9GHz) CPU, and 16GB LPDDR3 RAM. Performance is evaluated in two ways. In the optimized transactions, the generation speed of association rules is evaluated. The rule generation speed and objective usefulness of the generated association rules are evaluated according to association rule algorithms.

In the first performance evaluation, the proposed method is compared with a conventional method in terms of the generation speed of association rules in the transactions related to traffic accident topics. From news stream data, 1700 traffic accident data are collected through crawling and are converted into a corpus. The collected corpus is cleansed for the improved quality of the analysis. In the data cleaning process, missing values and outliers are processed, and unnecessary data are removed, so that data dimensionality is reduced. In the morphological analysis, traffic accident topics are extracted, and transactions are designed with the use of TF-IDF weights. For the comparison of the performance of the association rule algorithm, the data as the results of FP-Tree and Apriori algorithms are generated. In the first evaluation, the generation speed of association rules is compared according to the independent changes in the minimum support and confidence. In order to evaluate
the performance of the associated-words graph generation using TF-IDF weight-based ranking and
the performance of the associated-words graph generation with no use of TF-IDF, this study compared
the generation speed of association rules [41]. In other words, the generation speed and count are
compared according to the top 5%, 10%, 15%, and 20% of TF-IDF. In consideration of the characteristics
of the stream data created in real-time, the support and confidence that fit knowledge generation is
judged. When it comes to the comparison of generation speed according to a change in support, the
value of confidence that best expresses generation speed is 0.1. Accordingly, in the condition that
the confidence value is set to 0.1, the rule generation speed is compared according to a change in the
minimum support. Figure 11 presents word count and generation speed according to TF-IDF and min
support. The results from the comparison of the association rule generation speed according to the
changes in TF-IDF ranking ratio and the minimum support.

Figure 11. Word Count and Generation Speed according to TF-IDF and Min Support.

As shown in Figure 11, in the condition that the confidence value is set to 0.1 and the support
value is 0.005, the model with the use of TF-IDF 15%, 20% are compared to the model with no use of
TF-IDF, spends about 16.7 seconds and 9 seconds more in generating association rules. In the condition
that the support value is over 0.01, the difference was 22 seconds or more and thus the association rule
generation speed value is greatly reduced. Additionally, in the condition when support value is over
0.01, the model in which words in top TF-IDF ranking are applied shortens the time of generating
association rules. In the condition that the support value is fixed to 0.01 and a confidence value varies,
the association rule generation speed is compared. Figure 12 shows word count and generation speed
according to TF-IDF and confidence. The results from the comparison of the association rule generation
speed according to the changes in TF-IDF and confidence.

As shown in Figure 12, the difference in the association rule generation speed depending on
whether there is TF-IDF ranking applied is 12 seconds (about 44 times or more) in all measured
confidence values. In particular, in the model with the use of TF-IDF, its rule generation speed in all
confidence values does not exceed one second. That is because the number of unnecessary words (in
transactions) is reduced. Therefore, the proposed algorithm generates association rules faster than a
conventional association rule algorithm.
As shown in Figure 12, the proposed Apriori-TF-IDF algorithm generates rules 0.4~0.8 seconds faster than other algorithms. For the objective evaluation, support, confidence, and lift are applied. Table 7 shows average support, confidence value of association algorithms.

As shown in Figure 13, the proposed Apriori-TF-IDF algorithm generates rules 0.4~0.8 seconds faster than other algorithms. For the objective evaluation, support, confidence, and lift are applied. Table 7 shows average support, confidence value of association algorithms.
As shown in Table 7, FP-Tree and Apriori algorithms that do not use TF-IDF Ranking Scores have higher average values of support than the algorithm that uses the ranking. Regarding a value of confidence, Apriori with no use of ranking has the highest score, followed by Apriori with the use of top 20% TF-IDF Ranking, FP-Tree with no use of ranking, and FP-Tree with the use of top 20% TF-IDF Ranking in order.

Nevertheless, judging the consistency of rules with the uses of Support and Confidence is limited. For example, if the association rule \{Beverage\} -\> \{Coke\} is extracted on the basis of the terms Beverage and Coke, there is a possibility that the rule is a common-sense rule, rather than unpredicted new information. From the perspective of usefulness, there is no positive result. Therefore, the usefulness of rules is evaluated with the use of the lift. Table 8 shows average lift value of association algorithms. Figure 14 shows the average lift score of association algorithms by number of rules.

Table 7. Average Support, Confidence Value of Association Algorithms.

|       | 500 Rules | 1000 Rules | 1500 Rules | 2000 Rules | 2500 Rules |
|-------|-----------|------------|------------|------------|------------|
| Support |           |            |            |            |            |
| TF-IDF top 0 FP-Tree | 0.09979 | 0.08379 | 0.07304 | 0.06389 | 0.06087 |
| TF-IDF top 20 FP-Tree | 0.05093 | 0.04188 | 0.03635 | 0.03278 | 0.03036 |
| TF-IDF top 0 Apriori | 0.1404 | 0.11343 | 0.09818 | 0.09097 | 0.08495 |
| TF-IDF top 20 Apriori | 0.06794 | 0.05186 | 0.04522 | 0.04133 | 0.03903 |
| Confidence |           |            |            |            |            |
| TF-IDF top 0 FP-Tree | 0.17034 | 0.17132 | 0.17251 | 0.17189 | 0.17189 |
| TF-IDF top 20 FP-Tree | 0.16144 | 0.16273 | 0.16123 | 0.16078 | 0.16114 |
| TF-IDF top 0 Apriori | 0.4745 | 0.5077 | 0.4941 | 0.4877 | 0.4902 |
| TF-IDF top 20 Apriori | 0.3793 | 0.4138 | 0.4398 | 0.4559 | 0.4653 |

Table 8. Average Lift Value of Association Algorithms.

| Lift | 500 Rules | 1000 Rules | 1500 Rules | 2000 Rules | 2500 Rules |
|------|-----------|------------|------------|------------|------------|
| TF-IDF top 0 FP-Tree | 1.15356 | 1.2108 | 1.2697 | 1.31759 | 1.34029 |
| TF-IDF top 20 FP-Tree | 1.42306 | 1.50636 | 1.52958 | 1.62002 | 1.69837 |
| TF-IDF top 0 Apriori | 1.5555 | 1.6721 | 1.6558 | 1.6719 | 1.794 |
| TF-IDF top 20 Apriori | 1.991 | 2.7055 | 3.1918 | 3.3879 | 3.8551 |

Figure 14. Average Lift Score of Association Algorithms by Number of Rules.

5. Discussion and Conclusions

In the lift-based evaluation, the proposed algorithm shows two times better performance than other association rule algorithms. In particular, with the increase in the number of rules, the algorithm with the use of TF-IDF Ranking had far better performance. Therefore, no matter how much the
number of association rules increases in the stream news corpus collected in real-time, the proposed method improves the rule generation speed and usefulness. In addition, through the optimization of the knowledge graph, it is possible to extract significant information in real-time. This study proposed the method of optimizing the associative knowledge graph using TF-IDF based ranking scores. The proposed method calculates the TF-IDF weights for words in the news corpus related to traffic accident topics to make the ranking scores of all words. Word ranking is applied to remove the words which are not in the top 20% scores of all words extracted from the news corpus. Word data of the news corpus are optimized and are converted into transactions. A TID is set by news and item sets are generated. With the generated transactions, association rules of words are generated. According to the association rules, the edge based on confidence weight and the vertex based on word importance are generated and visualized in the knowledge graph. In the evaluation of performance (a degree of optimization), the graph was compared with the associated-words knowledge graph with no use of TF-IDF ranking. An association rule algorithm with the use of TF-IDF was compared with an association rule algorithm with no use of TF-IDF in terms of rule generation speed. As a result, in the condition that the support value is 0.01 or more and in all values of confidence, the association rule algorithm with the use of TF-IDF generated association rules about 22 seconds (44 times or more) faster than the association rule algorithm with no use of TF-IDF. In addition, the average lift value of the proposed TF-IDF based association rule algorithm was two times (up to 2.51) higher than those of Apriori and FP-Tree algorithms, so that the proposed one generated more useful association rules. Therefore, when an association rule knowledge graph is generated with the use of TF-IDF, it is possible to quickly make association rules for massive data collected in real-time. Given the two-times increase in the lift value, the usefulness of association rules is better. The contributions of the methods proposed in this paper are as follows:

(1) The method proposed in this paper has a problem in that it does not count the number of word duplicates when it is composed of transactions. Therefore, in order to solve the problem of the existing association rule algorithm, the transaction was optimized by using the ranking based on the TF-IDF weight.

(2) By removing unnecessary keywords and considering the characteristics of stream data generated in real-time, the speed of generation is improved.

(3) The effectiveness and usefulness of providing knowledge was improved. Accordingly, it is possible to extract new information that is hard to be predicted and provide information to the user intuitively.

The future plan is to apply the classification model based on the top TF-IDF importance application to the corpus in various domains for category classification. In addition, it is planned to process data in order for efficient data analysis in the classification model, and then conduct modeling to estimate a causal relation.

Author Contributions: H.-J.K. implemented the optimization of associative knowledge graph using TF-IDF based ranking scores. H.-J.K. and J.-W.B. performed the experiments and analyzed the results. K.C. and J.-W.B. conceived and designed the framework. All authors have contributed in writing and proofreading the paper. All authors have read and agree to the published version of the manuscript.

Funding: This work is supported by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 20CTAP-C157011-01).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, S.R.; Kang, M.M.; Park, S.M. The Future of Big Data. Commun. Korean Inst. Inf. Sci. Eng. 2018, 30, 18–24.
2. Song, G.Y.; Cheon, Y.; Lee, K.; Lim, H.; Chung, K.Y.; Rim, H.C. Multiple categorizations of products: Cognitive modeling of customers through social media data mining. Pers. Ubiquitous Comput. 2014, 18, 1387–1403. [CrossRef]
3. Kang, J.S.; Shin, D.H.; Baek, J.W.; Chung, K. Activity Recommendation Model using Rank Correlation for Chronic Stress Management. *Appl. Sci.* **2019**, *9*, 4284. [CrossRef]

4. Baek, J.W.; Kim, J.C.; Chun, J.; Chung, K. Hybrid Clustering based Health Decision-making for improving Dietary Habits. *Technol. Health Care* **2019**, *27*, 459–472. [CrossRef]

5. Chung, K.; Park, R.C. P2P based open health cloud for medicines management. *Peer Peer Netw. Appl.* **2020**, *13*, 610–622. [CrossRef]

6. Kim, J.C.; Chung, K. Sequential-index Pattern Mining for Lifecare Telecommunication Platform. *Clust. Comput.* **2019**, *22*, 1039–1048. [CrossRef]

7. Kang, J.S.; Baek, J.W.; Chung, K. Multimodal Media Content Classification using Keyword Weighting for Recommendation. *J. Converg. Inf. Technol.* **2019**, *9*, 1–6.

8. Kim, D.; Seo, D.; Cho, S.; Kang, P. Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec. *Inf. Sci.* **2019**, *477*, 15–29. [CrossRef]

9. Luna, J.M.; Padilla, F.; Pechenizkiy, M.; Ventura, S. Apriori versions based on mapreduce for mining frequent patterns on big data. *IEEE Trans. Cybern.* **2017**, *48*, 2851–2865. [CrossRef] [PubMed]

10. Nair, J.J.; Thomas, S. Improvised Apriori with frequent subgraph tree for extracting frequent subgraphs. *J. Intell. Fuzzy Syst.* **2017**, *32*, 3209–3219. [CrossRef]

11. Sheng, G.; Hou, H.; Jiang, X.; Chen, Y. A novel association rule mining method of big data for power transformers state parameters based on probabilistic graph model. *IEEE Trans. Smart Grid* **2016**, *9*, 695–702. [CrossRef]

12. Regar, E.; Silfianti, W. Visit Patterns Analysis of Foreign Tourist in Indonesian Territory Using Frequent Pattern Growth (FP-Growth) Algorithm. *Int. J. Comput. Sci. Softw. Eng.* **2018**, *7*, 212–217.

13. Silva, J.; Varela, N.; López, L.; Millán, R. Association Rules Extraction for Customer Segmentation in the SMEs Sector Using the Apriori Algorithm. *Procedia Comput. Sci.* **2019**, *151*, 1207–1212. [CrossRef]

14. Widiartha, K.K.; Dewi, D. Shopping Cart Analysis System in Product Layout Management with Apriori Algorithm. *Int. J. Appl. Comput. Sci. Inform. Eng. (ACSIE)* **2019**, *7*, 151–157.

15. Ministry of Land, Infrastructure and Transport. Available online: [https://www.molit.go.kr](https://www.molit.go.kr) (accessed on 15 April 2020).

16. MBC News (Munhwa Broadcasting Corporation News). Available online: [https://imnews.imbc.com](https://imnews.imbc.com) (accessed on 15 April 2020).

17. Zhao, F.; Sun, H.; Jin, L.; Jin, H. Structure-augmented knowledge graph embedding for sparse data with rule learning. *Comput. Commun.* **2020**, *159*, 271–278. [CrossRef]

18. Emebo, O.; Varde, A.S.; Melo, G. Commonsense Knowledge in Machine Intelligence. *SIGMOD Rec.* **2017**, *46*, 49–52. [CrossRef]

19. Paik, J.H. A novel TF-IDF weighting scheme for effective ranking. In Proceedings of the WorldComp DMIN 2016 International Conference on Data Mining, Las Vegas, NV, USA, 25–28 July 2016; pp. 146–152.

20. Yun-tao, Z.; Ling, G.; Yong-cheng, W. An improved TF-IDF approach for text classification. *J. Zhejiang Univ. Sci. A* **2005**, *6A*, 49–55. [CrossRef]

21. Qaiser, S.; Ali, R. Text mining: Use of TF-IDF to examine the relevance of words to documents. *Int. J. Comput. Appl.* **2018**, *181*, 25–29. [CrossRef]

22. Mohsen, A.M.; Hassan, H.A.; Idrees, A.M. Documents emotions classification model based on tf-idf weighting measure. *Int. J. Comput. Inf. Eng.* **2016**, *10*, 252–258.
27. Rozeva, A.; Zerkova, S. Assessing semantic similarity of texts—Methods and algorithms. In Proceedings of the 43rd International Conference Applications of Mathematics in Engineering and Economics, American of Conference, Sozopol, Bulgaria, 8–13 June 2017; Volume 1910.

28. D’Haen, J.; Poel, D.; Thorleuchter, D.; Benoit, D.F. Integrating expert knowledge and multilingual web crawling data in a lead qualification system. Decis. Support Syst. 2016, 82, 69–78. [CrossRef]

29. Beautiful Soup Package. Available online: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (accessed on 15 April 2020).

30. Prajapati, D.J.; Garg, S.; Chauhan, N.C. Interesting association rule mining with consistent and inconsistent rule detection from big sales data in distributed environment. Future Comput. Inform. J. 2017, 2, 19–30. [CrossRef]

31. Yoon, B.; Park, I.; Coh, B.Y. Exploring technological opportunities by linking data and products: Application of morphology analysis and text mining. Technol. Forecast. Soc. Chang. 2014, 86, 287–303. [CrossRef]

32. Kim, J.C.; Chung, K. Associative feature information extraction using text mining from health big data. Wirel. Pers. Commun. 2019, 105, 691–707. [CrossRef]

33. Kim, J.C.; Chung, K. Discovery of knowledge of associative relations using opinion mining based on a health platform. Pers. Ubiquitous Comput. 2019. [CrossRef]

34. Chung, K.; Jung, H. Knowledge-based dynamic cluster model for healthcare management using a convolutional neural network. Inf. Technol. Manag. 2020, 21, 41–50. [CrossRef]

35. Duan, Y.; Shao, L.; Hu, G.; Zhou, A.; Zou, Q.; Lin, A. Specifying architecture of knowledge graph with data graph, information graph, knowledge graph and wisdom graph. In Proceedings of the IEEE 15th International Conference on Software Engineering Research, London, UK, 7–9 June 2017; pp. 327–332.

36. Wu-Yan, E.; Betzel, R.F.; Tang, E.; Gu, S.; Pasqualetti, F.; Bassett, D.S. Benchmarking measures of network controllability on canonical graph models. J. Nonlinear Sci. 2018. [CrossRef]

37. Liu, L.; Yu, S.; Wei, X.; Ning, Z. An improved Apriori-based algorithm for friends recommendation in microblog. Int. J. Commun. Syst. 2018, 31, e3453. [CrossRef]

38. Shouzhong, T.; Minlie, H. Mining microblog user interests based on TextRank with TF-IDF factor. J. China Univ. Posts Telecommun. 2016, 23, 40–46. [CrossRef]

39. Lee, J.H.; Lee, M.; Kim, J.W. A study on Korean language processing using TF-IDF. J. Inf. Syst. 2019, 28, 105–121.

40. Ko, J.W.; Chung, K.; Han, J.S. Model transformation verification using similarity and graph comparison algorithm. Multimed. Tools Appl. 2015, 74, 8907–8920. [CrossRef]

41. Xue, Y.; Li, T.; Zhang, H.; Wu, X.; Li, M.; Hu, X. An apriori-based algorithm for mining semi-order-preserving submatrix. Int. J. Comput. Sci. Eng. 2016, 13, 66–79. [CrossRef]

42. Park, S.S.; Chung, K. MMCNet: Deep learning-based multimodal classification model using dynamic knowledge. Pers. Ubiquitous Comput. 2019. [CrossRef]

43. Choi, S.Y.; Chung, K. Knowledge process of health big data using mapreduce-based associative mining. Pers. Ubiquitous Comput. 2019. [CrossRef]

44. Jung, H.; Chung, K. Social mining based clustering process for big-data integration. J. Ambient Intell. Humaniz. Comput. 2020. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).