sustainability

MDPI

Article

ACS: Construction Data Auto-Correction System—Taiwan Public Construction Data Example

Meng-Lin Yu and Meng-Han Tsai *

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan; m10705510@ntust.edu.tw
* Correspondence: menghan@mail.ntust.edu.tw; Tel.: +8862-2737-6356

Abstract: This study aims to develop an automatic data correction system for correcting the public construction data. The unstructured nature of the construction data presents challenges for its management. The different user habits, time-consuming system operation, and long pretraining time all make the data management system full of data in an inconsistent format or even incorrect data. Processing the construction data into a machine-readable format is not only time-consuming but also labor-intensive. Therefore, this study used Taiwan's public construction data as an example case to develop a natural language processing (NLP) and machine learning-based text classification system, coined as automatic correction system (ACS). The developed system is designed to automatically correct the public construction data, meanwhile improving the efficiency of manual data correction. The ACS has two main features: data correction that converts unstructured data into structured data; a recommendation function that provides users with a recommendation list for manual data correction. For implementation, the developed system was used to correct the data in the public construction cost estimation system (PCCES) in Taiwan. We expect that the ACS can improve the accuracy of the data in the public construction database to increase the efficiency of the practitioners in executing projects. The results show that the system can correct 18,511 data points with an accuracy of 76%. Additionally, the system was also validated to reduce the system operation time by 51.69%.

Keywords: natural language processing; construction data management; machine learning

Citation: Yu, M.-L.; Tsai, M.-H. ACS: Construction Data Auto-Correction System—Taiwan Public Construction Data Example. *Sustainability* **2021**, *13*, 362. https://doi.org/10.3390/su13010362

Received: 21 November 2020
Accepted: 18 December 2020
Published: 3 January 2021

1. Introduction

1.1. Construction Data Management

Data management has been considered as one of the most vital tasks in the construction industry [1]. The unstructured nature (e.g., plain text) of the construction data presents challenges for its management. Processing the construction data into a machine-readable format is time-consuming and labor-intensive, as it requires lots of paperwork to structuralize data from various sources. Nowadays, many popular tools and standards have been developed by experts to help people manage construction data. Management tools, such as Microsoft Project and Primavera, are prevalent. The MasterFormat and UniFormat standards are common in the U.S. and Canada. However, engineers still need to follow the rules and form formats provided by standards and tools to manually transform unstructured data into structured data.

Even though many tools and standards are ready to follow, the quality of processed data is not sufficient due to the various backgrounds of the related personnel. Engineers from different fields may have their own interpretations of the standards, thus causing inconsistency of the data format or even incorrect data. For instance, Taiwan government provides a tool to its employees and contractors to manage public construction projects using a coding system similar to MasterFormat. However, due to the different interpretations of the coding standard, the average data accuracy of 7592 public construction projects is only 48%.

With the advancement of information technology, the improvement of computer performance, as well as large data storage, technologies such as artificial intelligence and deep learning have flourished for solving complex data management problems. Such improvement of these technologies can automatically transform unstructured data into structured data.

1.2. Public Construction Cost Estimation System in Taiwan

In Taiwan, it is required by law that for public projects with bids over NTD 10 million, the project documents need to be managed using the public construction cost estimation system (PCCES). Additionally, as the PCCES is a coding system, the law also requires that the accuracy rate of data encoded in the project documents should be at least 40%. However, according to the statistics of the Public Construction Commission, Executive Yuan (PCC), Taiwan, from 1 December 2016 to 30 June 2020, there were 422 public construction works projects that needed to use the PCCES. Among the 422 cases, 215 projects had statistics on the correct rate of coding, and the average correct rate was 37.47% [2].

The Taiwan government has been planning to make public construction work projects more transparent since 1995. The solution to this issue is making its own coding standard for the construction field. The Taiwan government imitated the MasterFormat coding standard formulated by the Construction Specifications Institute (CSI) and formulated its own coding standard [3]. According to the coding structure of MasterFormat, the PCCES coding standard is divided into chapters 00 to 16, for a total of 17 chapters. Its organization is arranged according to engineering practice and the experience of engineers. According to the work breakdown structure (WBS) [4], it is also divided into five code and four layer structures. The coding architecture is shown in Figure 1. In the first layer, the first and the second codes are the numbers of the chapter. In the second layer, the third code is a major category. In the third layer, the fourth code is the detailed classification code under each major category. For the fourth layer, the fifth code is for the work item, which is related to the third layer. Users can customize the code, but it should be recognized by the engineering committee's public engineering technology database control management.

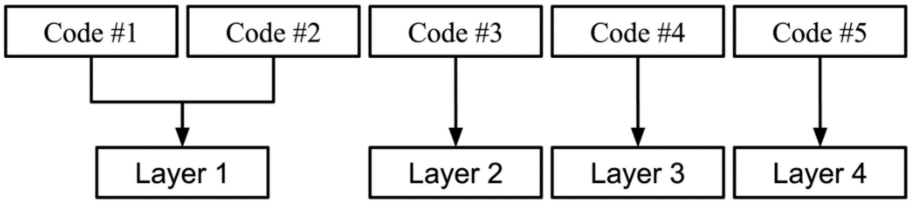

Figure 1. The coding architecture of the public construction cost estimation system (PCCES) [3].

The PCCES code considers work items and resource items. Resource items include materials, human resources, and machines. Work items are included in the composition of materials, machinery, labor, and miscellaneous. The coding of work items and resource items can be used differently. As there are many projects in public works, the scale of these projects varies, ranging from airports, dams, and tunnels to planting sidewalk trees, paving bricks on sidewalks, and dredging trenches, all within the scope of public works. Each project requires the participation of many upstream and downstream manufacturers, office staff, and government officials.

Due to different roles and divisions of labor, the perspectives of project management vary. Projects of various sizes, workers from different sources, and different roles in the project will cause these people to have a different understanding of the PCCES encoding system when using it. These differences in understanding have resulted in non-compliant materials in the project. However, these non-compliant materials have accumulated over time, presenting challenges for the database. Due to these reasons, with the accumulation of days and months, the database has become full of non-compliant data.

1.3. Benefits and Challenges of the PCCES

The PCCES occupies a very important position in the field of public construction works in Taiwan. Public construction projects require the use of the PCCES in accordance with the law, so these projects have uniform specifications that can be followed. The PCCES offers some benefits after collecting a large number of projects and integrating data. For example, according to the PCC announcement [3], the PCCES can improve the efficiency and credibility of fund review and fund comparison, avoid repeating the establishment of system software and hardware by various agencies, save national public funds, and reduce the opportunities for restricting competition and restricting bidding during project bidding.

Although the PCCES does bring these benefits and helps users a lot, some researches have pointed out problems with the system. For instance, users are not used to the standardized work items and resource codes in the PCCES [5]. Users need longer education and training to be familiar with and operate the PCCES [6]. Many practical applications are still not included in the PCCES [7,8]. The operation of the PCCES is time-consuming, requires a lot of manpower, and has a high error rate [9]. Moreover, in the production of PCCES documents there are general contradictions between architects and construction companies in measurement and calculation methods, and some measurement and calculation methods do not conform to the PCCES coding standard [10].

1.4. Research Objectives

This study focuses on the development of a machine-learning-based system to auto-correct a public database of the construction field that contains the PCCES data in Taiwan. People cannot use the data in the public database as the database is filled with messy data. In actuality, while people use many tools for managing construction data, the data are still chaotic. The quality of these data depends on how familiar a user is with the tools. There is no classification method that can consider the meaning of construction specifications in the construction classification system's database and then automatically correct the wrong data. Therefore, this study aims to develop a machine-learning-based system to improve the performance of people who work on the construction files and need to use construction classification systems. The developed method should achieve the following goals:

(1) Develop an auto-correct function to automatically correct the data from public databases related to the construction field and the unstructured construction project data. Users could use this function to correct data that they obtained from the open data database made by the government.
(2) Develop a recommender function, which can help users to perform their job efficiently without having experience in construction classification systems.

2. Literature Review

In order to develop a system that can automatically structuralize the construction data, this study first reviews the characteristics and challenges of construction data management (Section 2.1). Then, the novel methods for unistructural data processing proposed by other studies is discussed in Section 2.2. Lastly, the review of state-of-the-art machine-learning-based approaches to construction data processing is conducted in Section 2.3.

2.1. Challenges in Construction Data Management

Processing unstructured data is considered one of the most critical challenges in construction data management. Project owners, architects, contractors, and suppliers communicate and coordinate through various documents. It is common that documents have different naming conventions for the same object [11]. The different naming conventions increase the time and cost of processing the data as the personnel need to organize the raw materials for further communications. Attempts are being made by some to use a construction classification system (CCS), such as the MasterFormat [12], UniFormat [13], and OmniClass [12], to lower the extra cost of data processing. CCS usually uses layers of

coding to categorize various materials and adds narrative descriptions. This system allows all team members to use the storage efficiently and use retrieval mechanism codes in the system to reference specific parts of any document, reducing the cost [14]. However, even users who are working on the same project will have different CCS systems according to their various roles in the project. It is not feasible to make all users familiar with the CCS systems used by each other: the learning cost is too high. Not all of the team members are familiar with these specifications, which may cause the inputted data to remain unstructured.

In the field of construction engineering, there is also the problem of dealing with the unstructured data [15–17]. Unstructured data mean the data set is not stored in the structured format in the database; an alternative definition may be that the data do not follow a predefined data model composition. This makes the data irregular, ambiguous, and difficult to understand using traditional computer programs [18]. Many documents are generated in a construction project, including images and text. The text-based unstructured materials include contract templates, construction specifications, quality documents, and material management documents [17]. Among them, only 20% of the available data are structured and stored in a relational database, while approximately 80% are unstructured text and stored in various forms of documents [19]. Traditionally, unstructured data are expected to be converted to structured data by manual work. However, acquiring knowledge from unstructured data is usually painful and expensive [20]. Therefore, several studies have been conducted to determine a simpler and cheaper means of retrieving useful information from unstructured data.

2.2. Unstructured Data Processing

Unstructured data have been considered a critical challenge in data management for decades. Several studies have been conducted on transferring the unstructured data into useful information in many fields. For instance, Kharrazi et al. used natural language processing (NLP) to solve the problem of unstructured electronic health records [21]. Luo et al. used the knowledge from data specialists and computer data modules to extract structured data from unstructured medical records [22]. In the business field, Farhadloo et al. attempted to discover the relative importance of each service or unique product using the Bayesian method for a customer review system [23]. These investigators used an online analytical processing (OLAP) system to analyze unstructured data from multiple perspectives, including text mining (TM), information retrieval (IR), and information extraction (IE), in an attempt to extract business intelligence from unstructured data.

In the field of construction engineering, researchers have also tried to solve the problem of unstructured data. In the study of [16], a view-based method was used, with metadata models to convert documents to structured data. Alsubaey et al. presented a Naïve Bayes text mining approach to identify early warnings of failure from meeting records [24]. Kim and Chi developed a system based on natural language processing (NLP) to extract hidden knowledge from construction accident cases [25]. Even though studies have been conducted that address unstructured data, none of these results can solve the problem that the CCS faces.

The issue for the CCS is in the material codes and in challenges in providing accurate description of materials. For materials in the CCS, there is a coding system for specifications, and the CCS uses specific terms to describe the specifications. However, in actuality, not everyone can master the coding system and become familiar with these terms. People use unprofessional terms in construction files at work, and it works fine as these terms are readable. Even though the codes and descriptions in construction files are invalid in the CCS, these construction files contain data with coding errors yet proper descriptions of unstructured data.

Machine learning is a popular solution to convert unstructured data into structured data. Machine learning is flourishing due to the improvement of hardware computing power, the reduction of data storage costs, and the innovation of various algorithms.

With the advancement of machine learning, various machine learning algorithms enable the model to learn from data, which makes computers able to handle more and more tasks. After training the model with a large number of examples, classifying the data or predicting the model training, the model can extract information from samples to learn and can complete specific tasks or prediction. These abilities allow a computer to complete specific tasks or make predictions for a variety of applications. Machine learning can gradually replace human resources in specific tasks, such as in autonomous vehicles [26], voice recognition [27], weather prediction [28], face recognition [29], lie detection [7], image processing [30], etc.

2.3. Machine-Learning-Based Methods For Construction Data Processing

A typical construction project may have thousands of outstanding issues. An artificial intelligence program that can help humans systematize these problems and the data accompanying them would greatly improve the efficiency of work. There are many things that artificial intelligence can accomplish. For images, in the unstructured data processing of images, some algorithms have performed classification of pictures directly [31], and some studies have algorithms that give these images a text caption, then these images became text data [32]. For text, Wu et al. used natural language analysis to automatically extract keyword lists from pathological examination reports [33]. Nandhakumar et al. used the characteristics of words or sentences and conditional random field (CRF) models to extract important parts of medical reports [34]. The methods described above are attempts made to structure data.

3. ACS Methodology

3.1. System Overview

This study proposes an automatic correction system (ACS) that can correct the data automatically or provide a recommended list to users for manual correction. The proposed system includes three primary modules: a data processing module, a search processing module, and a mapping processing module. Figure 2 shows the system overview of the ACS. The data processing module processes the raw data collected from different resources and then stores them in the result database. The main job of the data processing module is to train a word embedding model and establish a result data set. The search processing module processes the target data input by the user and then puts them into the word embedding model to find data with higher similarity. The data processing module and the search processing module both use the same text processor [35]. The following sections will describe each module sequentially.

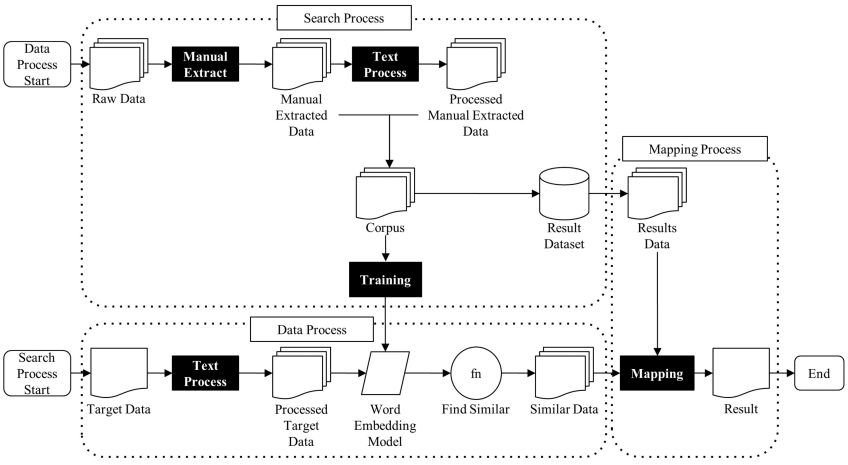

Figure 2. The designed system structures for the automatic correction system (ACS).

3.2. Data Processing Module

The data processing module processes the raw data then uses the processed raw data to train a word embedding model. This study used the corrected blank valuations and the CCS system manual as the raw data. The data processing module uses the text processor to normalize the data that were manually extracted from the raw data. These processed data will then be inserted into a corpus. Subsequently, correct code, correct name, correct name's segmentation, and original name's segmentation are extracted from the corpus as the result data and stored in the result database. At the end of this module, the corpus is taken as the input for training the word embedding model. Figure 3 shows the system overview of the ACS.

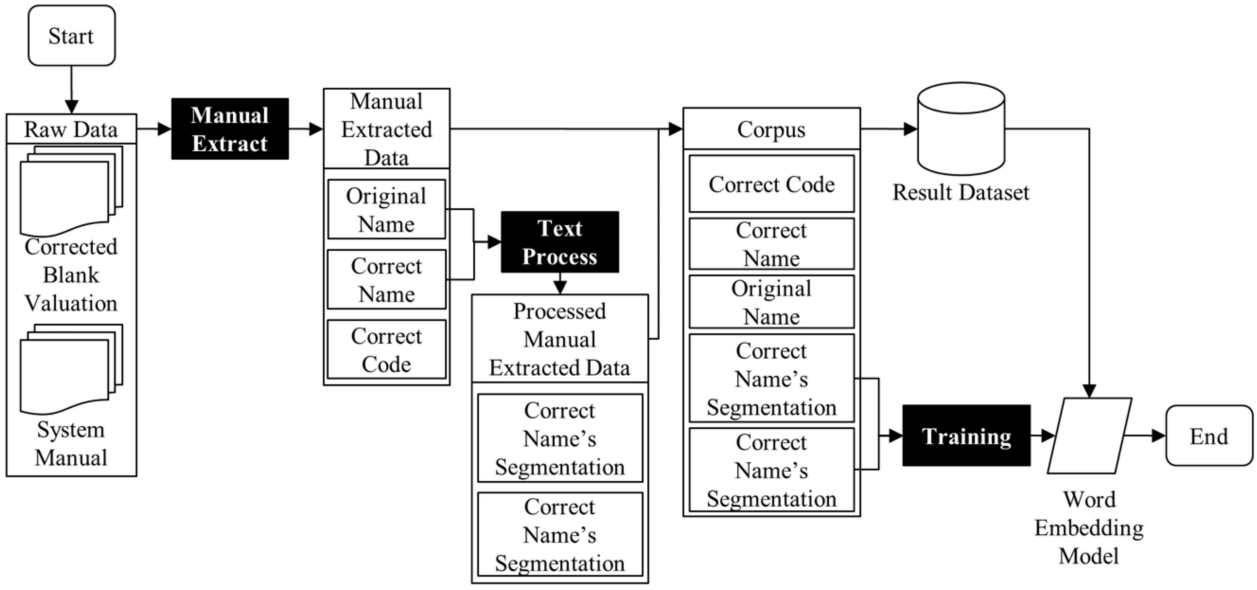

Figure 3. The workflow of the ACS's data processing module.

3.2.1. Raw Data Collection

For the raw data collection, the system manuals and blank valuations were collected that related to a specific CCS. The ACS has a particular use in correcting a public database associated with one specific CCS, and the performance of the ACS depends on the quality of the collected data. System descriptive documents were collected from the institutions that manage CCS, and blank valuations that use CCS were received from a private company's real cases. For the descriptive documents of the ACS, they at least contain specifications, descriptions, and coding systems. These correct codes and descriptions were extracted as a data column to two data fields in the "Manual Extracted Data," termed as "Correct Code" and "Correct Name." The inaccurate description was also extracted as a data column, termed as "Original Name".

3.2.2. Text Processor

The text preprocessing is advantageous for the subsequent classification results and can reduce the complexity of the calculation. Figure 4 shows workflow of the ACS's text processor. The text processor handles the text preprocessing and contains the following tasks: stop word removal, lowercase conversion, normalization, and tokenization. After users enter data, such as a sentence, the text processor removes stop words [36], converts uppercase and lowercase letters to be consistent, normalizes the preprocessing data obtained thus far, and finally uses tokenization to segment the data. An alias dictionary is utilized for the tokenizing. The dictionary includes synonyms, which can increase the quality of statements [35].

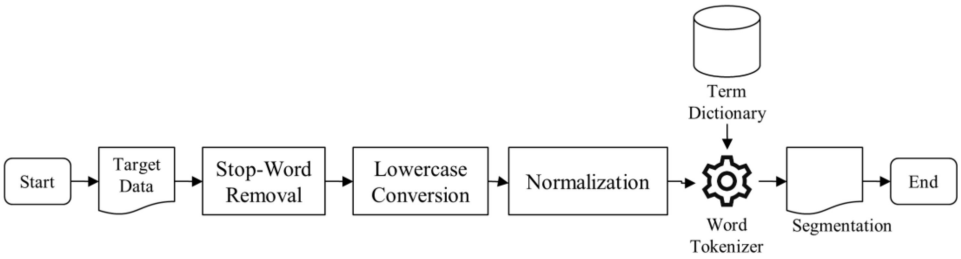

Figure 4. The workflow of the text processor.

3.3. Search Processing Module

For the search processing module, the input is the target data that the user keys in, and the output data are data with higher similarity obtained from the word embedding model. After the data processing module trains the word embedding model, the search processing module uses the text processor to handle the input target data and convert the target data into segmentations. Subsequently, the segmentation is taken as the input to the word embedding model to find similar data. The output of this module is the similar data that the model found. Figure 5 shows the system overview of the ACS.

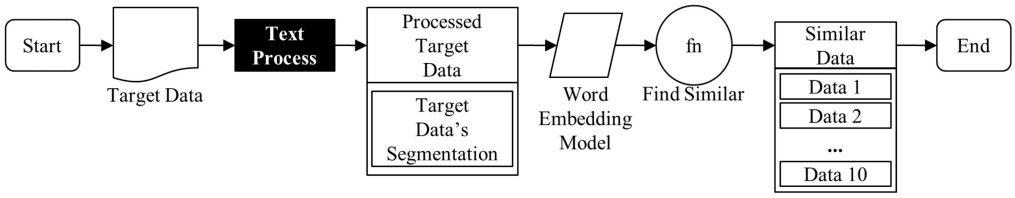

Figure 5. The workflow of the search process module.

3.3.1. Word Embedding

Word embedding is one of the most popular representations in documentation vocabulary [37]. It can capture the context, semantic, and syntactic similarity of words in documents, and the relationships with other words. Roughly speaking, it is vector representations of specific words. The program cannot directly use the text contained in an electronic file, and thus the text needs to be converted into a format that the computer can handle, with word vector representation being one of these conversion methods. A series of processing steps will be performed on the text until the text becomes a sentence or word. These sentences or words are given an independent code, and the code is a vector.

There are many ways to do word embedding, such as hashing vectorizer [38], count vectorizer [39], and Term Frequency – Inverse Document Frequency (TF-IDF) vectorizer [40]. The method we used in this system was a shallow neural network, where by learning a large amount of text, words are transformed into vectors in vector space. Then, a distribution of a large number of vectors in vector space is used to calculate similarity and find words with similar meanings. Words with the same meanings have identical representations. This representation is considered a fundamental breakthrough in machine learning for natural language processing problems [41]. Here, the characteristics of word embedding technologies are taken as the core application of the system.

Word embedding methods include dimensionality reduction of word co-occurrence matrices [42–44], probability models [45], and explicit representation of the context in which words are located [46]. In the ACS, we used the continuous bag of words (CBOW) [47] as the language model to obtain the vector matrix. Figure 6 shows the system overview of the ACS.

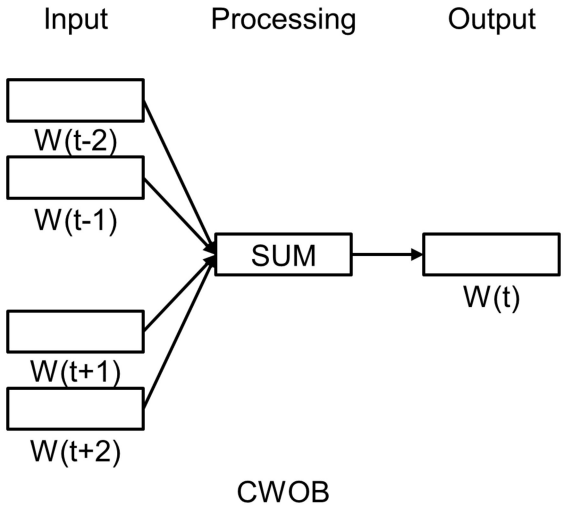

Figure 6. The architecture of the continuous bag of words (CBOW) model.

In the CBOW model, the context of surrounding words is used to predict the word in the middle. The input layer is one-hot encoding [48] and the size is equal to N. Each element in the input layer corresponds to the words in a vocabulary. Zero means no input, and a 1 means a word is input. The hidden layer is the input layer multiplied by the weight matrix ($1 \times N * N \times V = 1 \times V$). Since the input is a hot-coded vector, the hidden layer is the result of superimposing multiple rows from the weight matrix. The row index of these rows is equal to the index, with the input element being 1. Therefore, the input layer is like a lookup table for the weight matrix row search.

3.3.2. Similarity Calculation

The ACS uses text similarity to correct the data. For example, there are data that have been classified as wrong data, since the description is not sufficiently accurate even if it is semantically close. The system can replace the incorrect data with the most similar data if they can be found from the model.

The ACS deals with the text that users entered as a result and then inserts the results into the word embedding model. The result is the segmentation of the text. The model will give a vector to the result, which can represent the result in the model's vector space. Then, this study uses the cosine similarity [49–51] to calculate the similarity of vectors. The ACS takes the results in the result database, inputs them into the model to obtain vectors, and then uses the vectors from the results and the vectors from user model entries. After the vectors are input to the model, the ACS calculates the normalized dot-product from the cosine angle. Given two data a and b represented as two vectors V_a and V_b, the cosine similarity can be calculated as Equation (1).

$$cosine(V_a, V_b) = \frac{\sum_{i=1}^{N} V_a \cdot V_b}{\sqrt{\sum_{i=1}^{N} {V_{ai}}^2} \sqrt{\sum_{i=1}^{N} {V_{bi}}^2}}. \quad (1)$$

V_a is the frequency of each word in the user statement after being disassembled. V_b is the frequency of each word in the corpus statement after being disassembled. The $cosine(V_a, V_b)$ can represent the similarity of these two vectors. The closer the angle is to zero degrees, the more similar the two vectors are.

3.4. Mapping Process Module

The mapping process module finds the correct code from the result data set depending on the similar data from the search processing module. The system obtains a data set and data after finishing the above two modules. The result database contains the data processed by the text processor in the data processing module. The data in the database are similar to

the target data processed by the search processing module. The mapping process module will pick the data from the similar data one by one as keywords to query the result database. If it finds the data that contain the segmentation that 100% matches the keyword, it takes the correct code from the data as the output. Figure 7 shows the system overview of the ACS.

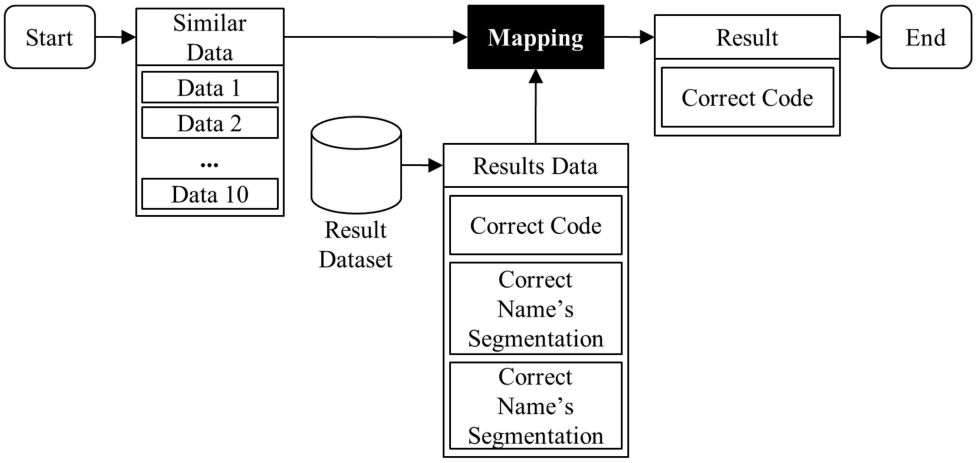

Figure 7. The workflow of the mapping process module.

4. Implementation

This research used the PCCES data as the training data to implement the developed system. The following subsections will describe the training data and the implementation of the ACS's three modules.

4.1. Training Data

For data training, in this study two types of data were collected for training the word embedding model: the PCCES manual and manually corrected actual project data. The latter was used, as these data were more in line with real work scenarios.

4.1.1. PCCES System Manuals

In the PCCES manuals, there are many of the specification codes and instructions. Anyone who wants to use the PCCES to estimate the project budget needs to know how to use the PCCES manuals. People need to be trained to know which objects they want to evaluate and where to place the objects, such as human resources, machines, materials, methods, and the environment. Then, they need to select the correct manual chapter. The specification code and description contain different layers, and thus a code and specification are needed for picking objects layer by layer. Finally, after combining these picked codes and specifications, the data processing is completed with 100% accuracy.

One-hundred percent correct information was obtained from the manual. There are 18 chapters in PCCES manuals, which cover the tender documents and contract items such as general requirements and fieldwork. This study only selected concrete chapters for implementations.

In a chapter there are multiple sections, and each section has a code designed for the name, specification, and unit of the material. Permutations were used to generate all possible data. The grid shown in Table 1 was used to choose one code from each column and combine them as a specification code with a description. For example, the code "03330/4/2/0/0/2" means "Building Concrete/Ready-Mixed Underwater Concrete/140 kgf/cm^2/M^2." The exhaustive method in the table was used to list all specification codes, with these specification codes being one of the training data for the word embedding model.

Table 1. The example of the PCCES manuals.

Chapter No.	Chapter Name	Class (6th)	Compressive Strength (7th)	Cement Type (8th)	Chemical Admixture (9th)	Valuation Unit (10th)
03330	Building Concrete	(0)	(0)	(0)	(0)	M (1)
		Machine-mixed (1)	80 kgf/cm^2 (1)	-	-	M2 (2)
		Ready-mixed (2)	140 kgf/cm^2 (2)	-	-	3 M (3)
		Machine-mixed underwater (3)	175 kgf/cm^2 (3)	-	-	Lump (4)
		Ready-mixed underwater (4)	280 kgf/cm^2 (4)	-	-	T (5)
		Ready-mixed, under 10F (5)	315 kgf/cm^2 (5)	-	-	Piece (6)
		Ready-mixed, under 20F (6)	350 kgf/cm^2 (6)	-	-	Each (7)
		Ready-mixed, under 30F (7)	400 kgf/cm^2 (7)	-	-	Set (8)
		-	-	-	-	KG (9)

4.1.2. Blank Valuations

Real project documents were obtained from two companies, Knowledge Analysis Space Exploration, Inc., Taipei, Taiwan and United Geotech, Inc., Taipei, Taiwan, which were undertaking public works, and these documents were one of the word embedding model training data sets. Each construction project contained a large number of documents and data files. The desired files in these documents and files were the price valuation files. The price valuation file contained names, terms, and aliases of objects or materials in the construction files. In actuality, the price valuation file may contain vague information that people can understand while the PCCES cannot. The system is unable to process the information and needs humans to correct it for the system to understand. Those corrected files were the desired files for collection as they contained two types of information: vague information and manually structured data. These two types of information can point to each other. Table 2 lists the example of the valuation we collected.

Table 2. The example of corrected blank valuation.

Wrong Data		Correct Data	
Work Item/Material	**Code**	**Work Item/Material**	**Code**
Technician	L00000520A5	Senior Worker	L000006200001
Unskilled Laborer	L0000061005	Junior Worker	L000006100001
Plastic Road Marking	M02898A003	Product, Road Marking, Glass Ball	M0289801009
Reflective Glass Ball	M02898B003	Road Marking, Glass Ball	02898B0009
Adhesive	M02900C000F	Product, Road Marking Adhesive	M0289800019
Equipment Fee	E0512450001	Not Classified Machinery	E000001000004
Tool Wear	W0127120004	Tool Wear	W0127120004

4.2. System Implementation

The ACS can be divided into four parts: the text processor, the database, the model training, and the searching and mapping. In this section, the implementation of these four parts is introduced.

4.2.1. Text Processing

The text processor used in the ACS was developed and used in our previous work [35]. In the previous work, a proper term dictionary was already constructed to improve the quality of data segmentation. This dictionary was used to develop a text processor, which could be used to handle any data imported into the system. Since the PCCES is written in Chinese, we implemented specific Chinese text processors, especially for Taiwan's construction field.

We constructed a dictionary for terms in Taiwan's construction field to solve these issues. This dictionary was used in the text processor to normalize the data. The text processor unified inconsistent units in Chinese, removed and replaced Chinese symbols and stop words, and replaced unit symbols with the real letter or number. The results are shown in Table 3. For example, if the input text was "Building Concrete and Ready-Mixed Underwater Concrete/140 kgf/cm^2/m^2," the output would be "Building Concrete/Ready-Mixed Underwater Concrete/140 kgf/cm^2/M^2."

Table 3. The characteristics that this study processed in the text processor.

	Before	**Action**
Stop words	"." (period)	Remove
Symbols	"m^2" (square meter)	Replace with "m^2"
Unnecessary prepositions	"and", "or", "included"	Remove
Full width character	"(" (left parenthesis), "。" (period)	Remove
Unify units	"3000psi"	Replace with "210 kgf/cm^2"

4.2.2. Database

Specific information was extracted from the raw data and stored in a database for use in training the word embedding model. For the training data, there were already the PCCES manuals and blank valuations. However, these data were raw data that could not be used to train the word embedding model directly, and thus there was a need to first preprocess these data. One-hundred percent correct data were already generated from PCCES manuals. Furthermore, there were the original and corrected blank valuations. The 100% accurate data had two data types: specification code and specification description. For the fixed blank valuations, there were four data types: correct specification code, accurate specification description, invalid specification code, and invalid specification description. These four different data types were combined as four data table fields. The correct specification description and incorrect specification description were used to extend the other data fields, termed as "correct description segmentation" and "original description segmentation." This data field was used to store the result of the text preprocessing module after processing the specification text.

For example, consider three data strings. The first was from 100% correct data and was "0331043003, Building Concrete and Ready-Mixed Underwater Concrete 140 kgf/cm^2 m^3." The others were from the corrected blank valuations and were "0331000003, 140 kgf/cm^2 premix concrete" and "0331023003, Building Concrete and Ready-Mixed Concrete 140 kgf/cm^2 m^3." After segmenting the data in the text processor, the data and data segmentation were placed in the data table, as shown in Tables 4 and 5.

Table 4. The example of 100% correct data that this study made from the PCCES manuals.

Field Name	Value
Correct_Code	0331043003
Correct_Desc	Building Concrete and Ready-Mixed Underwater Concrete 140 kgf/cm^{2}m^2
Original_Code	N/A
Original_Desc	N/A
Correct_Desc_Seg	"building concrete," "ready-mixed," "underwater," "concrete," "140 kgf/cm^3,""m^3"
Original_Desc_Seg	N/A

Table 5. The example of the corrected blank valuation that this study obtained.

Field Name	Value
Correct_Code	0331023003
Correct_Desc	Building Concrete and Ready-Mixed Concrete 140 kgf/cm^{2}m^2
Original_Code	331000003
Original_Desc	140 kgf/cm^2 premix concrete
Correct_Desc_Seg	"building concrete," "ready-mixed," "concrete," "140 kgf/cm^3,""m^3"
Original_Desc_Seg	"140 kgf/cm^2," "ready-mixed"

4.2.3. Model Training

In this study, data segmentation and the CBOW were used to train the model. The segmentation was stored in the previously constructed database. The language model used was CBOW, which was developed by Tomas Mikolov [47]. Because the scope of application of this research was for a small field, the use of CBOW was enough for the goal of solving this research, and compared with other models such as BERT [52] and GPT-2 [53], the cost of CBOW was low, due to reduced hardware requirements, reduced data volume, and more constant training time, so this study chose CBOW. The CBOW language model has been implemented in the gensim package provided in Python. This study used the gensim package to train the CBOW model.

In the database, the correct description segmentation and original description segmentation were already present. The segmentation was combined as a huge list that contained 18,513 data rows. This list was the input of the CBOW, and the output was the word embedding model. The listing method is shown in Table 6.

Table 6. The list of segmentation that this study used to train the word embedding model.

Index	Segmentation List
1	0331023003
2	"building concrete," "ready-mixed," "underwater," "concrete," "140 kgf/cm^3,""m^3"
...	...
n	"140 kgf/cm^2," "ready-mixed"

4.2.4. Searching

The goal of the search function was to find the 10 most similar data and return the data to the user. In the search function, after the user entered a term, sentence, or messy text, the system used a text processor to process the text entered by the user. After processing the input text using a text processor, the text processor generated a word segmentation, with the quality of the word segmentation equal to that of the training data. With these word segmentations, the training data with the correct data were extracted from the database and segmented. Then, in the word embedding model, one by one, the user input

text segmentation and training data segmentation performed calculations to find the 10 most similar data.

4.2.5. Mapping

The comparison function found the correct coding and narrative from the database according to the 10 most similar words. In the search function, the system completed the search and provided the 10 most similar words to the comparison function. The type of data was word segmentation, and it could not be used directly. At this time, the comparison function was needed to restore the word segmentation to the original text narrative, which was why the segmentation was stored in the database. SQL statements were used to compose database query commands, to query 10 segmentations of similar data separately, to find the correct code and correct description, and then to display it on the user interface for the user.

5. Validation

To validate that the ACS could structuralize and correct the existing data in the PCCES, this research conducted a system evaluation to evaluate the performance of the ACS. Additionally, a user test was also conducted to test if the recommendation function of the ACS could help related personnel to correct the data with higher efficiency. Two tests were designed to verify the usability of the ACS, one for the system evaluation and one for the user test. Whether the auto-correction function was complete and feasible was first tested, followed by testing of the recommendation function.

5.1. System Evaluation

A system evaluation test was conducted to validate the accuracy of the developed auto-correction function of the ACS. This study used the PCCES as an example and used real data from two companies in Taiwan as the data sources. The following subsections will describe the data source, classification, and the results of the test.

5.1.1. Data Source

The real data was obtained from two companies in Taiwan: Knowledge Analysis Space Exploration, Inc. and United Geotech, Inc. Furthermore, we generated 10,906 pieces of 100% correct data from the PCCES manuals. Totals of 5847 and 1382 raw data points were obtained from Knowledge Analysis Space Exploration, Inc. and United Geotech, Inc., respectively. The reason that these data were desirable was that they were from actual work projects; regardless of whether the code or the description of these data were correct, the project is still working fine, which means that the semantics in the data is accurate or close to the object.

For the 100% correct data, as per the codes and specifications listed in Table 1, six layers were included in the table. The PCCES manuals were used by picking one code and one description from each layer, depending on whether the description met the target material, and then combining these components. Finally, the data that perfectly fit the PCCES manuals were obtained. The user needed to insert this material in the project documentation if there was new material used in the construction project. For example, for a premixed concrete that had no additives and a strength of 4000 psi, then Table 1 could be used to generate the code and description for this material. The code of this material would be "0333024003," and the description would be "Building Concrete, Ready-Mixed Concrete, 280 kgf/cm^2, M3."

5.1.2. Classification

The corrected data were obtained after the raw data were processed by the developed system. These corrected data were used to calculate the accuracy of the automatic construction function to validate the performance and accuracy of the ACS. The obtained data included the raw data and correct data to calculate the accuracy of the corrected data.

Furthermore, the raw data and accurate data could correspond to each other. With the correct answers, we could know whether the system modified the data correctly or not.

Four rules were used to determine whether the automatically corrected data was correct to evaluate the accuracy of the automatic correction function: (1) if the raw data, manual correction data, and system correction data were all the same, it was correct; (2) if the manual calibration data and the machine calibration data were the same, it was accurate; (3) if the original data and the machine calibration data were the same, it was correct; and (4) after finishing the comparison with the first three rules, the information that was not included in the above three rules would be manually checked.

5.1.3. Results and Discussion for System Evaluation

In this section, the test result of the auto-correction function is presented. For the auto-correction function, the above four rules were used to judge whether the corrected data were correct. After completing the comparisons, 7392 and 1532 data points were obtained based on the first and second rules, respectively. Additionally, 4268 data points were obtained based on the third rule. Then, 1025 data points were confirmed manually. A total of 14,217 correct data points were obtained after the system processed the 18,551 input data points for a 76.64% accuracy. The results are shown in Table 7.

Table 7. The results of the auto-correction function evaluation.

Total (A)	Rule 1	Rule 2	Rule 3	Rule 4	Subtotal (B)	Correct Rate (B/Ax100)
18,551	7392	1532	4268	1025	14,217	76.64%

In the automatic correction function, the system took out the most similar data from the 10 most similar data and used these data to correct the wrong data. But in the failure cases, we observed that the correct answers to some of the failure cases were actually included in the 10 most similar data, but they were not the number one answers. In addition, there were some failure cases because they did not exist in the training data, such as tremie pipe and sprayed concrete, etc., so these cases were not applicable to this system.

5.2. User Test

Besides the auto-correction function, the ACS also provided a recommendation function to help the user correct the data manually. In order to test the usability of the recommendation function, this study designed a test and invited eight actual practitioners and students from the Department of Civil and Construction Engineering to use the ACS. The subjects were asked to conduct nine tasks by using both the PCCES and ACS. The operation times of each task were recorded for further analysis.

5.2.1. Background of the Subjects

For the user test, we invited eight users who had many years of industry experience and could use the PCCES proficiently at work, as well as students who had no work experience and no experience in using the PCCES. These users tested whether people with the same background in a specific field were familiar with the PCCES differently and whether there were differences in test results. The subjects included five civil engineering students without any experience in using the PCCES, and three civil engineers with 1, 4, and 8 years' experience. The details of these users are shown in Table 8.

Table 8. The background information of the invited subjects.

Subject	Background	Experience in Using PCCES (Years)
A	Undergrad student from Civil Engineering Department	0
B	Graduate students from Civil Engineering Department	0
C	Graduate students from Civil Engineering Department	0
D	Graduate students from Civil Engineering Department	0
E	Graduate students from Civil Engineering Department	0
F	Civil Engineer	1
G	Civil Engineer	4
H	Civil Engineer	8

5.2.2. Testing Scenario

In actuality, users often use email and Microsoft Excel when they are working. The resulting electronic documents contain various kinds of information. Users need to retrieve useful information from the electronic documents and then manually extract this information from the manual, finding the correct code and specification description in the PCCES to meet the work specification.

In this test scenario, we simulated the user's process of using the PCCES to create data in accordance with the description of materials, manpower, and equipment in the Excel file. Ten raw materials were extracted from some actual work project data to simulate the real work situation (Table 9). The eight testers were asked to use the PCCES and ACS to find their correct codes and specifications. At the same time, the time when they found the code was recorded, giving a usage time, and these statistics were further used to evaluate the benefits of the ACS for its users.

Table 9. List of the 10 tasks that were used in the user test.

Task	Raw Data
T1	Structure concrete, ready-mixed, 210 kgf/cm^2
T2	Structure concrete, including placing and compacting
T3	210 kg/cm^2 ready-mixed concrete
T4	Concrete placing and compacting
T5	Structure concrete, ready-mixed, 210 kgf/cm^2, nighttime construction
T6	Structure concrete, ready-mixed, 140 kgf/cm^2, nighttime construction
T7	140 kgf/cm^2 ready-mixed concrete
T8	Structure concrete, ready-mixed, 140 kgf/cm^2, daytime construction
T9	175 kg/cm^2 ready-mixed concrete
T10	280 kg/cm^2 ready-mixed concrete

5.2.3. Results and Discussion for the User Test

In this section, the results of asking users to check the recommendation function are presented. For the recommendation function, we mainly compared whether the ACS could help users work more efficiently than the PCCES, so the operating time was examined.

For the recommendation function, the times taken by users to process 10 raw data points using the two methods of the PCCES and ACS were recorded, and the processing times of the users according to each topic were averaged, as shown in Tables 10 and 11. Table 10 shows that after repeated operation of the PCCES, all the users minimized their operating times, and this minimum value could not be lowered. Additionally, as a result,

as shown in Table 11, no matter how good a user was at operating the PCCES, users had really low operating times in the ACS. The average operating times of each question in the two systems are shown in Figure 8. Comparing these two systems, even if the users were allowed to repeat the operation 10 times in the PCCES, the average working time of the PCCES was much higher than that of the ACS.

Table 10. Results of the user test with the PCCES.

User	Operation Time (Sec.)											
	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**Total**	**Avg.**
Q1	62.80	59.40	72.60	61.60	65.20	37.70	31.30	113.11	35.90	52.50	592.11	59.21
Q2	74.77	64.20	52.54	43.08	51.73	33.18	29.81	37.28	29.64	27.45	443.68	44.37
Q3	102.64	50.81	68.09	33.33	59.56	44.06	31.66	33.97	32.54	36.55	493.20	49.32
Q4	62.75	85.04	72.15	45.26	53.25	45.78	41.16	40.51	41.15	32.60	519.65	51.97
Q5	69.04	87.15	51.87	42.07	53.88	49.01	37.02	42.18	35.76	41.19	509.17	50.92
Q6	83.82	71.67	49.50	56.35	51.79	47.19	45.06	39.62	33.76	29.30	518.06	51.81
Q7	43.23	37.09	30.50	31.44	33.35	29.57	35.95	28.88	28.82	28.92	327.75	32.78
Q8	86.27	97.47	99.77	63.55	67.59	46.86	49.89	52.99	52.99	39.44	646.57	64.66
Total	585.32	552.83	497.01	376.68	446.35	333.35	301.85	388.54	388.54	287.95	-	-
Avg.	73.17	69.10	62.13	47.09	55.79	41.67	37.73	48.57	48.57	35.99	-	-

Table 11. Results of the user test with the ACS.

User	Operation Time (Sec.)											
	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**Total**	**Avg.**
Q1	12.15	17.90	22.11	22.72	30.81	28.11	27.25	21.60	23.02	22.30	227.97	22.80
Q2	11.14	22.83	20.38	19.20	24.52	21.78	21.29	21.96	20.44	21.37	204.91	20.49
Q3	18.30	15.12	29.43	30.68	28.01	24.92	22.82	27.51	25.14	21.39	243.32	24.33
Q4	16.76	13.09	24.14	22.76	28.61	27.13	37.92	23.62	18.42	22.33	234.78	23.48
Q5	15.61	20.36	31.42	25.92	27.95	28.04	29.01	26.26	25.32	27.97	257.86	25.79
Q6	21.34	16.82	38.53	38.23	32.84	31.97	28.09	28.72	26.69	25.01	288.24	28.82
Q7	12.01	19.42	17.52	16.43	23.14	17.75	17.57	22.18	15.68	16.66	178.36	17.84
Q8	25.80	16.63	23.10	28.34	37.57	20.23	20.33	25.06	23.69	19.50	240.25	24.03
Total	133.11	142.17	206.63	204.28	233.45	199.93	204.28	196.91	178.40	176.53	-	-
Avg.	16.64	17.77	25.83	25.54	29.18	24.99	25.54	24.61	22.30	22.07	-	-

In this testing, we found that for a continuously operating system, as the familiarity increases, the user's operating time will decrease to a point at which, finally, there is a bottleneck that prevents further reductions. The decline rate of the operating time of the PCCES was significant; however, the rate of decrease for the ACS was less obvious, and the user's operating time was almost the same. It could be seen that the recommendation function of the ACS greatly eliminated this proficiency factor. Compared with the PCCES, using the recommendation function of the ACS helped users save 54% of the operation time.

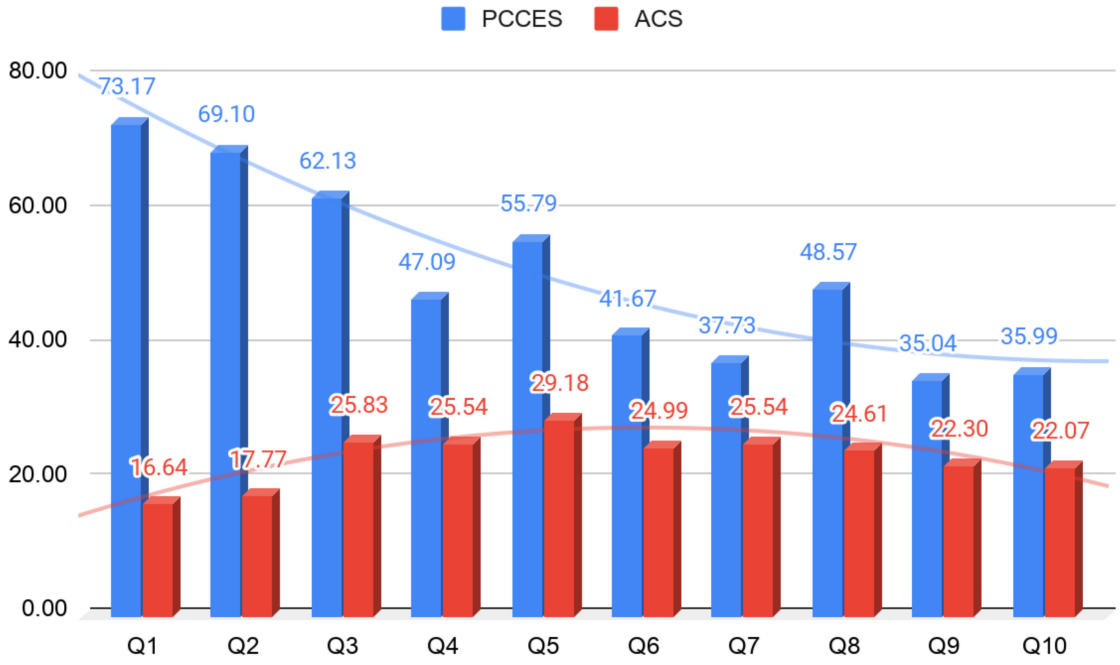

Figure 8. The average operating time for each task by eight users in two systems.

6. Discussions

The main contribution of this research is the development of a system that can automatically correct public databases in the field of construction engineering in Taiwan for a database that is full of chaotic data. Two main functions are implemented in the ACS: the auto-correction function and the recommendation function. No manual intervention is required during the processing of the auto-correction function, which can reduce the user's workload. The recommendation function allows users to input different keywords or similar words. The function will process the input text data and automatically check to find the most similar and standard data to provide to the user. The language model was trained based on the collected 18,551 data points, used to calculate the similarity between the user input text and the correct data, and then used the accurate data in the automatic correction function or provided a list of user recommendations. The auto-correction function was used to summarize 18,551 data points at an accuracy of 76%. Ten data out of the 18,551 were taken for user operation. Compared with the PCCES, using the recommendation function can help users save 54% in operation time.

6.1. Contributions

In the existing public database in Taiwan, there is a large amount of messy data, so that the Taiwan government needs to stipulate the accuracy rate of the data by law. We attempted to use the auto-correction function of the ACS to solve the problem of this messy data.

(1) The ACS can automatically correct and structuralize the untrusted construction data. Many studies have been conducted to structuralize the semi-structured data. For instance, Woo et al. used the text clustering method to clean the large-scaled medical report data [54]. Soto et al. proposed the ViTA-SSD system that allows the user to explore the insightful patterns in the semi-structured data by providing a visualized analysis method [55]. However, these methods still require lots of human interactions on results checking, correcting, and exploring. Instead, the ACS used a machine-learning-based method that can automatically correct and structuralize the construction data. It allows the ACS to improve quality and output efficiency and does not require human intervention in the processing process, which can reduce the workload of users.

Furthermore, the automation of operations to achieve the maximum benefit is a current industrial target.

(2) The ACS provides a recommendation function that can improve the related personnel's working efficiency and accuracy. As not everyone can be proficient in understanding the PCCES coding specifications, the produced data are often non-standard. Even for users who are skilled in using the PCCES, due to the operation steps of the PCCES, the operation time will encounter a bottleneck and cannot continue to be reduced. The recommendation function of the ACS is an attempt to solve the above two problems. By providing a more user-friendly experience and removing the need for familiarity of coding standards, the user's operating time and data accuracy are improved.

(3) The ACS successfully reduces the threshold of operating the data management system. It can be seen from the test of the recommendation function that, regardless of whether users have used the PCCES or not, when operating the PCCES, as the number of operations increases, the proficiency will increase, and the operation time will not continue to reduce past a certain level. This is because the PCCES code is attached and the operation interface has a lower operating time, even if the user is already skilled. The recommendation function implemented by the ACS relies on the user entering divergent keywords or similar words, such as material names, aliases, or specifications, and other obscure data. The system then provides the results that meet the PCCES specifications for the user to select. The benefit of the recommendation function is that it removes the variable proficiency levels. Regardless of whether the user has a professional background or is familiar with coding standards, he or she can rely on keywords to find the correct result.

(4) The system we developed is not limited to the PCCES, it can be applied to different CCS systems. Since the system is based on the CBOW language model, the data we collect will determine the application direction of this system. In this study, we use the data related to the PCCES system to obtain the model after training and use this model to classify and correct the wrong PCCES data. In other words, if the training data are replaced by another CCS system, this system can also be utilized on other CCS systems.

6.2. Limitations

(1) In this study, the data collected limited the scope of the ACS in the PCCES. In machine learning, data are the foundation of everything, where only by having more data can the application scope and accuracy be expanded. In this study, we only implemented one section of the PCCES coding specification. Most of the collected data belong to this category, as it is one of the most commonly used codes in this section. Furthermore, the specification description contained in this section has a description of the strength of the concrete. If this section can be improved, the coding accuracy rate will be beneficial for the PCCES as a whole.

(2) The accuracy of the ACS is not sufficient. In this study, we used 18,551 PCCES data to train the CBOW model, applied it to a small range of the PCCES, and got a 76.64% correct rate. The application range of the ACS in the PCCES depends on the data we collected, and although the accuracy of 76.64% was much higher than the 37.47% announced by PCC, it is not good enough to meet expectations. The application range of the ACS in the PCCES is not that wide. The average accuracy rate will decrease when the application range of the ACS is expanded and extended to a range that is difficult to automatically correct if the higher accuracy rate is not achieved in the application range for now. For the system that uses machine learning technology, the amount of data collected will affect the trained model and then affect its accuracy. If we can obtain more information, we may be able to improve the accuracy rate.

7. Conclusions

This research proposed a data correction system, the ACS, for automatically correcting the public construction data. The system we developed provides data auto-correction and recommendation features to improve human working performance and reduce the threshold of operating the data management system. A text classification system, the ACS, was developed using language models based on natural language processing and machine learning to correct public databases in the field of construction in Taiwan; at the same time, this system is also proposed to help users produce correct data more efficiently. By using a machine-learning-based language model to analyze text semantics and provide higher accuracy and efficiency information, the ACS can improve the efficiency of actual users and the accuracy of data in construction projects. In the automatic calibration test, a 76% accuracy rate was obtained after correcting 18,551 data points. A user test was also conducted on the recommendation function. A question was provided containing 10 real data points as well as a questionnaire to perform a user test on eight participants to observe them solving the problem under the two systems. After the trial, for the average processing time for each data point, 51.95% of the time was saved. From the test results, it was found that users using the ACS were more efficient than when using the original system and could accurately produce materials that meet the specifications. The results show that the ACS can effectively save operation time of the CCS and thus reduce the threshold of operating the data management system by providing a recommendation function. The proposed method can not only be used in the PCCES but can also be deployed to different CCS systems.

Author Contributions: Conceptualization, M.-H.T.; data curation, M.-H.T. and M.-L.Y.; methodology, M.-H.T. and M.-L.Y.; software, M.-L.Y.; validation, M.-L.Y.; writing—original draft, M.-L.Y.; writing—review and editing, M.-H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Taiwan's Ministry of Science and Technology (MOST) under contract 109-2124-M-002-005.

Acknowledgments: We would like to thank Ming-Dar Tsai of Knowledge Analysis Space Exploration and Liang-Yuan Li of National Taiwan University of Science and Technology for providing their feedback and for their assistance in the development of the system.

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

References

1. Aziz, R.F.; Hafez, S.M. Applying lean thinking in construction and performance improvement. *Alex. Eng. J.* **2013**, *52*, 679–695. [CrossRef]
2. Public Construction Commission. Turnkey Cases That Already Been Awarded between 1 December 2017 and 30 June 2020. Available online: https://pcces2.pcc.gov.tw/PCC_MRP/Announcement/AnnDetail/9dec7761-dc2c-4dad-8cd4-be61fa44e5a7 (accessed on 18 November 2020). (In Chinese)
3. Public Construction Commission 2020. Public Construction Cost Estimation System. Available online: https://pcces.pcc.gov.tw/CSInew/Default.aspx?FunID=Fun_12&SearchType=E (accessed on 18 November 2020).
4. Norman, E.S.; Brotherton, S.A.; Fried, R.T. *Breakdown Structures: The Foundation for Project Management Excellence*; John Wiley and Sons, Inc.: Hoboken, NJ, USA, 2008. [CrossRef]
5. Liao, M. Study of Current State of Software for Drafting and Estimating in Construction. National Kaohsiung University. 2011. Available online: https://hdl.handle.net/11296/t7u695 (accessed on 18 November 2020).
6. Yu, S.H. Evaluation of the Public Works Procedure Efficiency of E-Procurement of Government of Kaohsiung. National Sun Yat-sen University. 2004. Available online: https://hdl.handle.net/11296/w4h5xd (accessed on 18 November 2020).
7. Davatzikos, C.; Ruparel, K.; Fan, Y.; Shen, D.G.; Acharyya, M.; Loughead, J.W.; Gur, R.C.; Langlebenb, D.D. Classifying Spatial Patterns of Brain Activity with Machine Learning Methods: Application to Lie Detection. *NeuroImage* **2005**, *28*, 663–668. [CrossRef] [PubMed]
8. Lin, C.C. Study on Budget Rationality and Supervision Practice of Construction Safety and Health. 2014. Available online: https://www.grb.gov.tw/search/planDetail?id=8390698 (accessed on 18 November 2020).

9. Chen, Y.C. Combination of Public Construction Coding System and Building Information Modeling for Budget Estimate. Master's Thesis, National Taiwan University, Taipei, Taiwan. 19 June 2013. Available online: https://hdl.handle.net/11296/khec2y (accessed on 18 November 2020).
10. Huang, C.H.; Yang, I.T.; Wang, C.Y.; Wu, M.J. A Study of Introducing Omniclass on BIM-based Building Design Checking, Architecture and Building Research Institute, Ministry of the Interior, ROC (Taiwan): Taipei, Taiwan. 2017. Available online: https://www.grb.gov.tw/search/planDetail?id=12066805 (accessed on 18 November 2020).
11. Caldas, C.H.; Soibelman, L. Automating hierarchical document classification for construction management information systems. *Autom. Constr.* **2003**, *12*, 395–406. [CrossRef]
12. Construction Specifications Institute. CSI MasterFormat. 2008. Available online: https://www.csiresources.org/home (accessed on 18 November 2020).
13. Charette, R.P.; Marshall, H.E. *UNIFORMAT II Elemental Classification for Building Specifications, Cost Estimating and Cost Analysis*; U.S. Department of Commerce: Washington, DC, USA, 1999; NIST Interagency or Internal Reports (NISTIR) 6389.
14. Ioannou, P.G.; Liu, L.Y. Advanced Construction Technology System—ACTS. *J. Constr. Eng. Manag.* **1993**, *119*, 288–306. [CrossRef]
15. Russell, A.D.; Chiu, C.Y.; Korde, T. Visual Representation of Construction Management Data. *Autom. Constr.* **2009**, *18*, 1045–1062. [CrossRef]
16. Mao, W.; Zhu, Y.; Ahmad, I. Applying Metadata Models to Unstructured Content of Construction Documents: A View-Based Approach. *Autom. Constr.* **2007**, *16*, 242–252. [CrossRef]
17. Soibelman, L.; Wu, J.; Caldas, C.; Brilakis, I.; Lin, K.Y. Management and Analysis of Unstructured Construction Data Types. *Adv. Eng. Inform.* **2008**, *22*, 15–27. [CrossRef]
18. Sint, R.; Schaffert, S.; Stroka, S.; Ferstl, R. Combining unstructured, fully structured and semi-structured information in Semantic Wikis. In *CEUR Workshop Proceedings*; Heraklion: Crete, Greece, 2009; pp. 73–87.
19. Chen, H.M.; Schütz, R.; Kazman, R.; Matthes, F. Amazon in the Air: Innovating with Big Data at Lufthansa. In Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 5–8 January 2016; pp. 5096–5105. [CrossRef]
20. Rusu, O.; Halcu, I.; Grigoriu, O.; Neculoiu, G.; Sandulescu, V.; Marinescu, M.; Marinescu, V. Converting Unstructured and Semi-Structured Data into Knowledge. In Proceedings of the 2013 11th RoEduNet International Conference, Sinaia, Romania, 17–19 January 2013; pp. 1–4. [CrossRef]
21. Kharrazi, H.; Anzaldi, L.J.; Hernandez, L.; Davison, A.; Boyd, C.M.; Leff, B.; Kimura, J.; Weiner, J.P. The Value of Unstructured Electronic Health Record Data in Geriatric Syndrome Case Identification. *J. Am. Geriatr. Soc.* **2018**, *66*, 1499–1507. [CrossRef]
22. Luo, L.; Li, L.; Hu, J.; Wang, X.; Hou, B.; Zhang, T.; Zhao, L.P. A Hybrid Solution for Extracting Structured Medical Information from Unstructured Data in Medical Records via a Double-Reading/Entry System. *BMC Med. Inform. Decis. Mak.* **2016**, *16*, 114. [CrossRef]
23. Farhadloo, M.; Patterson, R.A.; Rolland, E. Modeling Customer Satisfaction from Unstructured Data Using a Bayesian Approach. *Decis. Support Syst.* **2016**, *90*, 1–11. [CrossRef]
24. Alsubaey, M.; Asadi, A.; Makatsoris, H. A Naïve Bayes Approach for EWS Detection by Text Mining of Unstructured Data: A Construction Project Case. In Proceedings of the IntelliSys 2015—Proceedings of 2015 SAI Intelligent Systems Conference, London, UK, 10–11 November 2015; pp. 164–168. [CrossRef]
25. Kim, T.; Chi, S. Accident Case Retrieval and analyses: Using natural language processing in the construction industry. *J. Constr. Eng. Manag.* **2019**, *145*, 04019004. [CrossRef]
26. Navarro, P.J.; Fernandez, C.; Borraz, R.; Alonso, D. A Machine Learning Approach to Pedestrian Detection for Autonomous Vehicles Using High-Definition 3D Range Data. *Sensors* **2016**, *17*, 18. [CrossRef]
27. Sainath, T.N.; Weiss, R.J.; Senior, A.; Wilson, K.W.; Vinyals, O. Learning the Speech Front-End with Raw Waveform CLDNNs. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Dresden, Germany, 6–10 September 2015; pp. 1–5. Available online: https://www.isca-speech.org/archive/interspeech_2015/i15_0001.html (accessed on 18 November 2020).
28. Krasnopolsky, V.M.; Fox-Rabinovitz, M.S. Complex Hybrid Models Combining Deterministic and Machine Learning Components for Numerical Climate Modeling and Weather Prediction. *Neural Netw.* **2006**, *19*, 122–134. [CrossRef]
29. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1528–1540. [CrossRef]
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
31. Lu, D.; Weng, Q. A Survey of Image Classification Methods and Techniques for Improving Classification Performance. *Int. J. Remote Sens.* **2007**, *28*, 823–870. [CrossRef]
32. Lu, J.; Yang, J.; Batra, D.; Parikh, D. Neural Baby Talk. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7219–7228. [CrossRef]
33. Wu, P.H.; Yu, A.; Tsai, C.W.; Koh, J.L.; Kuo, C.C.; Chen, A.L.P. Keyword Extraction and Structuralization of Medical Reports. *Health Inf. Sci. Syst.* **2020**, *8*, 18. [CrossRef]

34. Nandhakumar, N.; Sherkat, E.; Milios, E.E.; Gu, H.; Butler, M. Clinically Significant Information Extraction from Radiology Reports. In Proceedings of the 2017 ACM Symposium on Document Engineering—DocEng '17, Valletta, Malta, 4–7 September 2017; pp. 153–162. [CrossRef]
35. Yu, M.L.; Chan, H.Y.; Tsai, M.H. NLP-Based Method for Auto-Correcting Public Constructions Data. In Proceedings of the 2019 4th International Conference on Civil and Building Engineering Informatics, Sendai, Miyagi, Japan, 6–9 November 2019.
36. Hao, L.; Hao, L. Automatic Identification of Stop Words in Chinese Text Classification. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, IEEE, Wuhan, China, 12–14 December 2008; pp. 718–722. [CrossRef]
37. Yin, Z.; Shen, Y. On the Dimensionality of Word Embedding. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, Canada, 2–8 December 2018; pp. 887–898. Available online: http://papers.nips.cc/paper/7368-on-the-dimensionality-of-word-embedd (accessed on 21 December 2020).
38. Hasan, M.; Islam, I.; Hasan, K.M.A. Sentiment Analysis Using Out of Core Learning. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 1–6. [CrossRef]
39. Kulkarni, A.; Shivananda, A. Converting Text to Features. In *Natural Language Processing Recipes*; Apress: Berkeley, CA, USA, 2019; pp. 67–96. [CrossRef]
40. IEEE Photonics Technology Letters Information for Authors. *IEEE Photonics Technol. Lett.* **2009**, *21*, C3. [CrossRef]
41. Li, W.; Zhu, L.; Guo, K.; Shi, Y.; Zheng, Y. Build a Tourism-Specific Sentiment Lexicon via Word2vec. *Ann. Data Sci.* **2018**, *5*, 1–7. [CrossRef]
42. Lebret, R.; Collobert, R. Word Emdeddings through Hellinger PCA. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics Gothenburg, Gothenburg, Sweden, 26–30 April 2014; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 482–490. [CrossRef]
43. Levy, O.; Goldberg, Y. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems*; Mit Press: Montreal, QC, Canada, 2014; pp. 2177–2185. Available online: http://papers.nips.cc/paper/5477-neural-word-embedding-as (accessed on 21 December 2020).
44. Li, Y.; Xu, L.; Tian, F.; Jiang, L.; Zhong, X.; Chen, E. Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 3650–3656. Available online: https://www.ijcai.org/Proceedings/15/Papers/513.pdf (accessed on 21 December 2020).
45. Globerson, A.; Chechik, G.; Pereira, F.; Tishby, N. Euclidean Embedding of Co-Occurrence Data. *J. Mach. Learn. Res.* **2007**, *8*, 2265–2295. Available online: https://www.jmlr.org/papers/v8/globerson07a.html (accessed on 21 December 2020).
46. Qureshi, M.A.; Greene, D. EVE: Explainable Vector Based Embedding Technique Using Wikipedia. *J. Intell. Inf. Syst.* **2019**, *53*, 137–165. [CrossRef]
47. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013—Workshop Track Proceedings, Scottsdale, AZ, USA, 2–4 May 2013; Available online: http://arxiv.org/abs/1301.3781 (accessed on 21 December 2020).
48. Hatamian, M.; Hung, D.; Kurmas, Z.; Frenzel, J.; Pinter-Lucke, J.; and Zhao, P. In Praise of Digital Design and Computer Architecture. In *Digital Design and Computer Architecture*; Morgan Kaufmann: Burlington, MA, USA, 2016; pp. i–ii. [CrossRef]
49. Sidorov, G.; Gelbukh, A.; Gómez-Adorno, H.; Pinto, D. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Comput. Sist.* **2014**, *18*, 491–504. [CrossRef]
50. Basu, T.; Murthy, C.D. Effective Text Classification by a Supervised Feature Selection Approach. In Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops, Brussels, Belgium, 10 December 2012; pp. 918–925. [CrossRef]
51. Kim, K.; Chung, B.S.; Choi, Y.; Lee, S.; Jung, J.Y.; Park, J. Language Independent Semantic Kernels for Short-Text Classification. *Expert Syst. Appl.* **2014**, *41*, 735–743. [CrossRef]
52. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies—Proceedings of the Conference 1 (October), Minneapolis, MN, USA, 2–7 June 2019; Available online: http://arxiv.org/abs/1810.04805 (accessed on 21 December 2020).
53. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models Are Unsupervised Multitask Learners. *OpenAI Blog* **2020**, *1*, 9.
54. Woo, H.; Kim, K.; Cha, K.; Lee, J.Y.; Mun, H.; Cho, S.J.; Chung, J.I.; Pyo, J.H.; Lee, K.C.; Kang, M. Application of Efficient Data Cleaning Using Text Clustering for Semistructured Medical Reports to Large-Scale Stool Examination Reports: Methodology Study. *J. Med. Internet Res.* **2019**, *21*, e10013. [CrossRef]
55. Soto, A.J.; Kiros, R.; Kešelj, V.; Milios, E. Exploratory visual analysis and interactive pattern extraction from semi-structured data. *ACM Trans. Interact. Intell. Syst.* **2015**, *5*, 1–36. [CrossRef]