buildings

MDPI

Article

Building Materials Classification Model Based on Text Data Enhancement and Semantic Feature Extraction

Qiao Yan [1], Fei Jiao [1] and Wei Peng [1,2,*]

[1] School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan 250101, China; yanqiao@sdjzu.edu.cn (Q.Y.); jianzhujiao2022@163.com (F.J.)
[2] Anhui Province Key Laboratory of Intelligent Building & Building Energy Saving, Anhui Jianzhu University, Hefei 230601, China
* Correspondence: pengwei19@sdjzu.edu.cn

Abstract: In order to accurately extract and match carbon emission factors from the Chinese textual building materials list and construct a precise carbon emission factor database, it is crucial to accurately classify the textual building materials. In this study, a novel classification model based on text data enhancement and semantic feature extraction is proposed and applied for building materials classification. Firstly, the explanatory information on the building materials is collected and normalized to construct the original dataset. Then, the Latent Dirichlet Allocation and statistical-language-model-based hybrid ensemble data enhancement methods are explained in detail, and the semantic features closely related to the carbon emission factor are extracted by constructed composite convolutional networks and the transformed word vectors. Finally, the ensemble classification model is designed, constructed, and applied to match the carbon emission factor from the textual building materials. The experimental results show that the proposed model improves the $F1_{Macro}$ score by 4–12% compared to traditional machine learning and deep learning models.

Keywords: building materials classification; data enhancement; feature extraction; carbon emission factor

check for updates

Citation: Yan, Q.; Jiao, F.; Peng, W. Building Materials Classification Model Based on Text Data Enhancement and Semantic Feature Extraction. *Buildings* **2024**, *14*, 1859. https://doi.org/10.3390/buildings14061859

Academic Editor: Eugeniusz Koda

Received: 24 May 2024
Revised: 12 June 2024
Accepted: 18 June 2024
Published: 19 June 2024

1. Introduction

Currently, achieving the goal of "carbon peaking and carbon neutrality" is one of China's most important tasks [1]. In fact, the construction industry accounts for 50.9% of China's total carbon emissions and has become the most significant contributor [2]. Therefore, it is critical to accurately calculate and further reduce the carbon emissions from the construction industry to achieve China's "carbon peaking and carbon neutrality" goal.

When calculating carbon emissions during the building material production phase in the construction industry, the carbon emission factor (CEF) method is usually applied to calculate indirect carbon emissions according to national standards. The CEF method, in which the building materials are multiplied by the CEF of the corresponding material types, is applied to quantify carbon emissions from buildings. On the other hand, as the list of materials used in building engineering is sometimes recorded manually, the descriptions and names of the materials are not standardized [3]. Additionally, since there is a wide range of building materials on this list, it is difficult to classify them into appropriate material types and match them with the correct CEF for carbon emission calculations.

In recent years, text classification based on intelligent algorithms, such as Support Vector Machines (SVMs), K-Nearest Neighbor (KNN) algorithms, the Naive Bayes algorithm, and so on, has been widely used in fault diagnosis [4], sentiment analysis [5], fact checking [6], and other fields. However, SVMs are primarily used for handling binary classification problems and hardly applied to multi-class problems. KNN algorithms have a low tolerance for material types and a high dependence on sample quality. The Naive Bayes method may overfit and have poor generalization ability when dealing with many

building material features. Specifically, for textual building material classification, the aforementioned machine learning-based methods often neglect the significance of lexicality and the interplay between sentences and context.

Deep learning models, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM), demonstrate proficiency in extracting high-level contextual features from text. These models can analyze the semantics in different contextual features of building material texts, thus classifying material types more accurately. In [7], Kim pioneered using a CNN to extract text features for text classification. Aslan et al. [8] developed a multi-stage feature extraction model consisting of CNNs to classify online articles. Lu et al. [9] used the attention mechanism to extract text features and label information from different levels to complete a classification task. Zhong et al. [10] automatically classified complaint text by optimizing a CNN. Furthermore, a topic model can extract topic information from text by computing the probability distribution of each word in the text with respect to the topic [11], and word-embedding techniques based on Bidirectional Encoder Representation from Transformers (BERT) can map each word in the text to a low-dimensional vector representation that captures the semantic similarity between words [12,13]. In [14], BERT was used for word vector representation, and a CNN was used to capture static features. Liu et al. [15] analyzed the influence of emotion and cognition on learning by optimizing the BERT-CNN text classification model. Although deep learning offers good performance in text classification, its need for a large number of high-quality corpora restricts its application in building engineering with limited data. Therefore, corpus construction is the key to deep learning text categorization in engineering applications. Li et al. [16] drew inspiration from computer vision techniques, incorporating syntactic and semantic disturbances for data augmentation. Marivate et al. [17] constructed a corpus by randomly replacing words in a sentence using semantic similarity. Şahin et al. [18] enhanced data in the low-resource domain by cropping sentences into smaller fragments to synthesize new sentences. To the best of our knowledge, the aforementioned approaches have not yet been applied to building materials classification.

Kuniyoshi [19] utilized NLP techniques to retrieve and match materials literature based on the names and properties of materials. Song [20] proposed the MatSci-NLP model, which performs various NLP tasks, including classification, on materials texts. However, the materials texts used in these studies were derived from the scientific literature, which is standardized and uniformly formatted, without considering non-standard data sources. Elton [21] employed word-embedding techniques to extract the chemical relationships between materials from their textual descriptions. This method, however, relies on complete text documents and does not account for the limitations of short texts with insufficient contextual information. Yoshitake [22] introduced two MaterialBERT models that effectively reflect the meanings of material names through word embeddings, but they did not consider the development of downstream tasks based on these embeddings. Turhan [23] proposed integrating Large Language Models with Life Cycle Assessment to evaluate the environmental impact of construction materials, but this approach only considers specific environmental factors without quantifying the carbon emissions based on the CEF.

In order to match the CEF and the building material types exactly, a novel building materials classification model incorporating the Latent Dirichlet Allocation (LDA) algorithm, Ngram, and BERT into a CNN method is proposed. It utilizes three levels of data augmentation according to the classification feature, vectorized representation of text, and feature extraction of building materials text to achieve the type matching of building materials. The main innovations and contributions of this study are as follows:

- In order to extract keywords from a corpus of different building materials and enrich the original building material text, a data augmentation method combining the LDA algorithm and Ngram is proposed.
- To specifically capture contextual semantic information, a novel layered feature extraction network was constructed. In this network, the full test features are obtained by

the first convolutional layer; then, the key local features are further extracted by the composite convolutional layers.

- Experimental comparisons with various machine learning and deep learning models were conducted, and the results demonstrate the proposed method's superior performance in classifying building materials.

The remainder of this paper is structured as follows: Section 2 briefly reviews the problem of building material classification, and then the workflow of the proposed LNBC model is given. Section 3 presents a detailed, step-by-step explanation of the processes pertaining to the LNBC model. Both the experiments and corresponding analysis are given in Section 4. The conclusions and future work are presented in Section 5.

2. Problem Formulation and the Proposed Method

In this section, the problem of matching building material types with the CEF is briefly reviewed. Then, the construction of the proposed LNBC model is described.

2.1. Problems in Matching Building Material Types

The materials in the list are classified according to their types, as defined by the carbon emission factor (CEF) outlined in the "China Products Carbon Footprint Factors Database (CPCFFD)" [24]. This database serves as a generalized carbon emission database for carbon emission calculations in various fields, including industry, energy, and daily life. The classification of material types in the CPCFFD is organized into three hierarchical levels (as shown in Figure 1). However, this database does not provide detailed descriptions of the building materials classified under each type or information about the CEF specific to each material. Furthermore, the types of building materials are diverse, and the textual records are generally not standardized. Therefore, determining how to exactly match building materials with the corresponding CEF is becoming one of the most important problems in assessing buildings' carbon emissions.

Part of the building materials from a construction project

Material names
cotton yarn JC
insulated wire
white calico 32S/2
300*300 glazed floor tiles
insulation nail 8×82
M6×55 conical expansion bolt
0.8mm 60/40 solder
rebar φ6.5
hanging rod 2C14
flat steel 4×45
…

Part of materials type classification in the Carbon Emission Factor Database

Material types(level 1)	Material types(level 2)	Material types(level 3)
pure cotton carded yarn	pure cotton carded yarn	pure cotton carded yarn
wires, cables, optical cables, and electrical equipment	insulated wire	copper core polyethylene insulated wire
textile and garment industry	textile products	average textile products
non-metal	non-metallic mineral products	architectural ceramics - porcelain tiles - wet milling process
polymeric chemicals	synthetic resin	plastic - PVC
metal	ferrous metal smelting and rolling products	steel products
metal	ferrous metal smelting and rolling products	refined tin
metal	ferrous metal smelting and rolling products	rebar
metal	ferrous metal smelting and rolling products	small-sized steel materials
…	…	…

Figure 1. Part of the building materials list and carbon emission factor database.

2.2. The Proposed LNBC Model

As shown in Figure 2, the workflow of the proposed LNBC method consists of five steps, i.e., data preprocessing, data augmentation, word embedding, feature extraction and aggregation, and outputting the final classification.

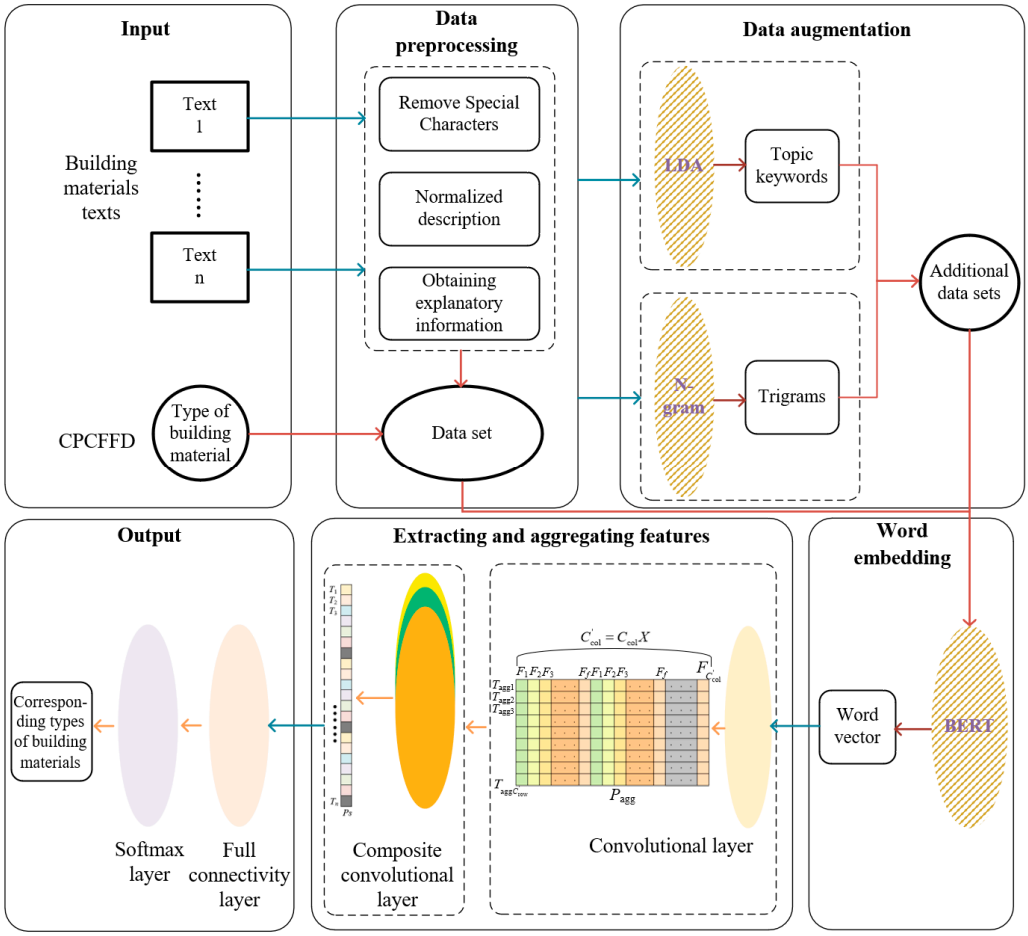

Figure 2. The workflow of the proposed method.

The workflow is as follows:

Step 1: Data Preprocessing. This initial phase involves preprocessing the original data, including eliminating character symbols (e.g., units), normalizing the text, and obtaining a detailed interpretation of each building material. This process yields a comprehensive explanation of each type of building material, facilitating their classification according to the relevant material type. Finally, an experimental dataset for the following steps is constructed.

Step 2: Data Augmentation. Keywords are first identified by applying LDA for each material type, and trigrams are generated using the Ngram algorithm. Then, the trigrams are inserted into the experimental dataset to achieve data augmentation if the relation coefficient between the identified keywords and the trigram exceeds the threshold.

Step 3: Word Embedding. In this step, the augmented data are transformed into word vectors through the Chinese-base-BERT model. This can guarantee that the converted vectors for the words retain the full contextual information and the complete semantic meaning.

Step 4: Feature Extraction and Aggregation. The global features of the building materials text are extracted through a single convolutional layer. Then, local features of neighboring words are identified using a composite convolutional layer. These local features are subsequently aggregated to represent the overall features of the text.

Step 5: Classification Output. The aggregated features are used as the input for the fully connected layer. The final classification result is obtained according to the value of the probability distribution from the output layer.

2.3. Research on the Practicability of the Model

In practical applications, the LNBC model can accurately classify various building materials in engineering project lists and ensure the correct application of the CEF, thereby enabling more precise carbon footprint calculations for building projects. Additionally, the LNBC model significantly enhances data-processing efficiency, reducing the time and effort required to manually match materials with their corresponding CEF. This efficiency is particularly beneficial for large-scale building projects that use a vast array of materials.

The LNBC model can also be integrated with carbon emission assessment systems. In the corresponding approaches, first, the LNBC model is designed as a modular component with developed Application Programming Interfaces (APIs) to enable communication with existing carbon emission assessment platforms. Second, seamless integration with system databases is ensured by connecting the LNBC model to existing building material databases, facilitating smooth data exchange. Finally, the LNBC model processes the classification results regarding the building materials and sends these results to the carbon emission calculation module via inter-module communication, replacing the manual matching of material CEF and, thereby, automating the carbon emission calculation process.

3. The Detailed Process of the Proposed LNBC Model

3.1. Data Preprocessing

In this study, the dataset was derived from a list of building materials associated with building engineering, containing records of 1700 building materials used. These building materials are used in various applications, including those structural and decorative, and for specific areas of specialization. As a result, they present a remarkable diversity of sizes and models. Moreover, the engineering list included a limited number of building materials, informal recordings, and poorly standardized descriptions, often using abbreviations or shorthand, and the texts are brief and contain limited semantic information. These factors collectively hinder the classification of building materials.

Noun interpretation serves as a translation method and technical approach to data augmentation. This technique improves a model's classification performance by adding extra contextual information and semantic expressions. In the data preprocessing phase, the main goal is to normalize all nouns relating to building materials and employ web crawler technology to obtain interpretations from the internet. The generated text includes the nouns along with their explanations. Furthermore, by analyzing each material's definition and scope of application, the building materials are classified into levels according to their types in the CPCFFD.

Table 1 displays a portion of the preprocessed data. The first column is the text pertaining to building materials, including their names and explanatory text. The subsequent columns are the types of materials.

Table 1. Potion of the data after preprocessing.

Building Materials Text	Material Types (Level 1)	Material Types (Level 2)	Material Types (Level 3)
Cotton yarn is made from cotton fibers through spinning. When processed into ply yarn, it becomes cotton thread. There are two types, Carded yarn: Made with a basic spinning system. Combed yarn: Made with a high-quality spinning system, resulting in smoother, stronger yarn used for premium fabrics.	pure cotton carded yarn	pure cotton carded yarn	pure cotton carded yarn
Insulated wire is wire covered with an insulating layer. It includes magnet wire and general-purpose insulated wire.	wires, cables, optical cables, and electrical	insulated wire	copper core polyethylene insulated wire

Table 1. *Cont.*

Building Materials Text	Material Types (Level 1)	Material Types (Level 2)	Material Types (Level 3)
White calico includes various materials like cotton, linen, silk, taffeta, and satin, each with unique characteristics and uses. Cotton: Comfortable, perfect for everyday clothes and bedding. Linen: Light and cool, ideal for summer wear. Silk: Soft and luxurious, great for fancy dresses. Taffeta: Transparent, used for lingerie. Satin: Shiny and elegant, chosen for wedding gowns and curtains.	textile and garment industry	textile products	average textile products
Floor tiles, made of porcelain or ceramic, are used for indoor and outdoor flooring. The size of the tiles is a key factor and depends on personal preference, design requirements, and room size. Larger tiles make spaces appear more spacious and tidy, and reduce the number of seams, making the floor smoother and easier to clean.	non-metal	non-metallic mineral products	architectural ceramics—porcelain tiles—wet milling process
Insulation nails are special engineering plastic expansion nails used to fasten insulation boards to walls. They are specifically designed for external wall insulation and are widely used in building decoration, particularly for anchoring wall insulation. They consist of a galvanized screw, a nylon expansion tube, and a fixed round plate.	polymeric chemicals	synthetic resin	plastic-PVC
Expansion bolts are devices used to anchor into concrete and other materials. They include a bolt, nut, nut sleeve, and spiral casing that together form an expansion anchoring system. The bolt is inserted into a pre-drilled hole and expands inside the hole through the action of the spiral casing and nut sleeve, providing a strong hold. They are used to fix structures like brackets, bridges, and pipes in construction projects.	metal	ferrous metal smelting and rolling products	steel products
Solder is a common welding material used to join components in electronics, appliances, and communications equipment. It has a low melting point and good wettability and fluidity, enabling reliable welded connections.	metal	ferrous metal smelting and rolling products	refined tin
Rebar, used in reinforced and prestressed concrete, usually has a round cross-section but can sometimes be square with rounded edges. Types include smooth, ribbed, and twisted rebar. Rebar for concrete can be straight or coiled, and comes in two types: smooth and deformed. Smooth round rebar is simply low-carbon steel in small diameters.	metal	ferrous metal smelting and rolling products	rebar
Hanging rod, shaped like an ingot and also known as an ingot bar or Yuanbao rod, is used to transfer concentrated forces from the bottom to the top of concrete beam components. This enhances the beam's ability to resist shear under concentrated loads.	metal	ferrous metal smelting and rolling products	rebar

Table 1. *Cont.*

Building Materials Text	Material Types (Level 1)	Material Types (Level 2)	Material Types (Level 3)
Flat steel is a metal with a large width-to-thickness ratio and a rectangular cross-section. Made of steel, it is thin and wide, used for frames, supports, brake pads, and mechanical parts. It is strong, rigid, and easy to process and cut into various shapes for customization.	metal	ferrous metal smelting and rolling products	small-sized steel materials

3.2. Data Augmentation at Different Levels

In the classification of R levels (where R = 1, 2, 3), building materials text belonging to the same type is grouped into the same corpus. Thus, corpora $X^R{}_1$, $X^R{}_2$, $X^R{}_3$, …, $X^R{}_n$ exist in R-level classification, where n represents the number of material types under R-level classification. A specific corpus $X^R{}_i$ can be expressed as $X^R{}_i = \{x_1, x_2, \ldots, x_u\}$, containing u pieces of text on building materials. Each text x_i (i = 1, 2, 3, …, u) is dissected into its constituent words (w_1, w_2, …, w_l), with l representing the word count in the text x_i.

In the corpus under R-level classification, while ensuring that the number of topic words is still set to 1, the set of keywords $K^R{}_{g,i}$ for each corpus is extracted by the LDA algorithm. The keywords collected from all corpora form the list [$K^R{}_{g,1}$, $K^R{}_{g,2}$, $K^R{}_{g,3}$, …, $K^R{}_{g,n}$], which is called the keywords table. According to the degree of relevance to the topic words, the elements above the constructed keywords table are then ranked, and the initial r keywords from this ranked list are selected, forming the topic keyword set $K^R{}_{z,i}$ for each corpus. The list of topic keywords across all corpora is as follows: [$K^R{}_{z,1}$, $K^R{}_{z,2}$, $K^R{}_{z,3}$, …, $K^R{}_{z,n}$].

For each building materials text x_i under R-level classification, a corresponding trigram is generated using the Ngram algorithm. The generated model is expressed by a set of trigrams, namely, $x_i = (tr^R{}_1, tr^R{}_2, \ldots, tr^R{}_q)$, where q is the total number of trigrams in x_i. Then, the text is expanded by incorporating the trigram if a trigram $tr^R{}_i$ (i = 1, 2, 3, …, q) in a text includes at least one topic keyword from the corpus set of topic keywords $K^R{}_{z,i}$. Thus, the original text will be expanded, and the expanded text is denoted as $A^R x_i = (w_1, w_2, \ldots, w_l, tr^R{}_1, tr^R{}_2, \ldots., tr^R{}_j)$, where $w_1, w_2, \ldots, w_l$ represent words in the original text x_i, and $tr^R{}_1, tr^R{}_2, \ldots., tr^R{}_j$ are the trigrams of that text under R-level classification. After three levels of data augmentation, the expanded text is represented as $A^3 x_i = (w_1, w_2, \ldots, w_l, tr^1{}_1, tr^1{}_2, \ldots., tr^1{}_j, tr^2{}_1, tr^2{}_2, \ldots., tr^2{}_k, tr^3{}_1, tr^3{}_2, \ldots., tr^3{}_z)$, where $w_1, w_2, \ldots, w_l$ are words in the original x_i text, and $tr^1{}_1, tr^1{}_2, \ldots., tr^1{}_j, tr^2{}_1, tr^2{}_2, \ldots., tr^2{}_k, tr^3{}_1, tr^3{}_2, \ldots., tr^3{}_z$ form trigrams of x_i. Here, $tr^1{}_1, tr^1{}_2, \ldots., tr^1{}_j$ contain at least one topic keyword of the first-level corpus; $tr^2{}_1, tr^2{}_2, \ldots., tr^2{}_k$ contain at least one topic keyword of the secondary corpus; and $tr^3{}_1, tr^3{}_2, \ldots., tr^3{}_z$ contain at least one topic keyword of the tertiary corpus where this text x_i is located.

Taking the building material text "hanging rod" as an example, Figure 3 shows the first-level text augmentation process. In Figure 3, the text "hanging rod" is assumed to have a serial number of 9 in the dataset, and its type under the first-level classification is assumed to be 6.

After the first-level augmentation, the expanded text of "hanging rod" is as follows: $A^1 x_9$ = ("Hanging", "rod", "shaped", "ingot", "Yuanbao", …, "concentrated", "loads", "common, welding, material", "electronics, appliances, communications", "low, melting, point", …, "Made, of, steel").

Based on the first-level text augmentation shown in Figure 3, Figure 4 illustrates the tertiary text augmentation process for building materials text. First-level (R = 1) data augmentation was conducted using the LDA algorithm and Ngram, and the corresponding expanded text $A^1 x_i$ was obtained. Based on $A^1 x_i$, $A^2 x_i$, which is the result of the second-level (R = 2) data augmentation, can be further calculated through the same process until

the third-level (R = 3) augmented text, A^3x_i, is obtained, constituting the final form of data augmentation.

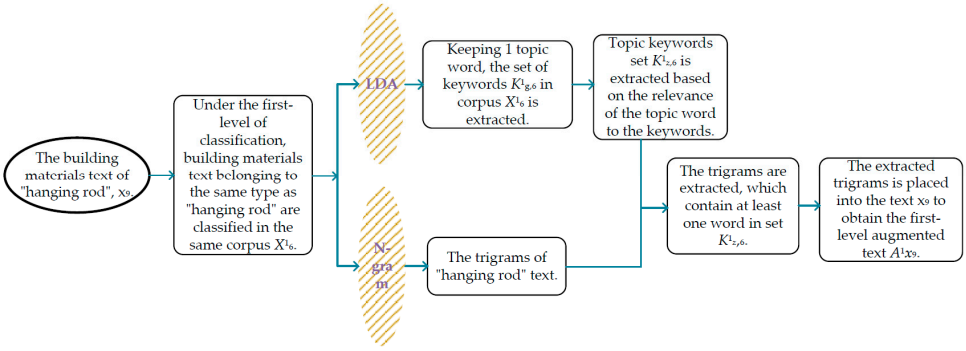

Figure 3. Example of first-level data augmentation.

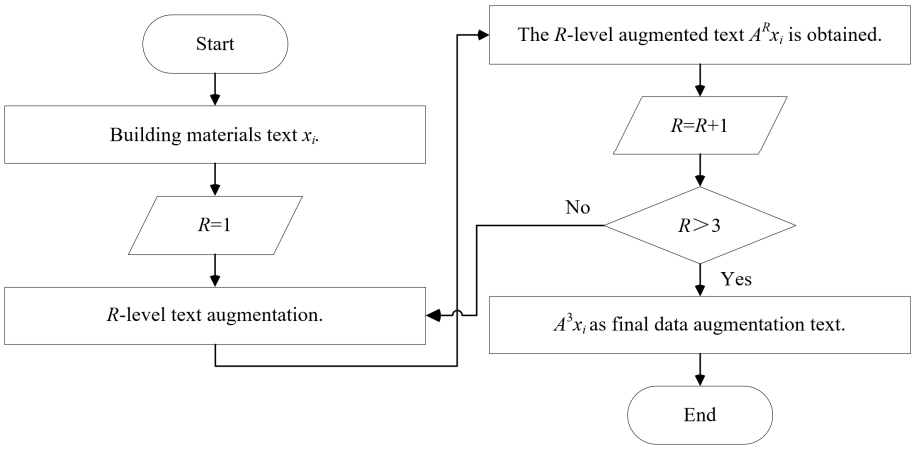

Figure 4. Data augmentation flowchart.

After the third-level iterated text augmentation, the final result of the expanded text "hanging rod" is shown in Figure 5.

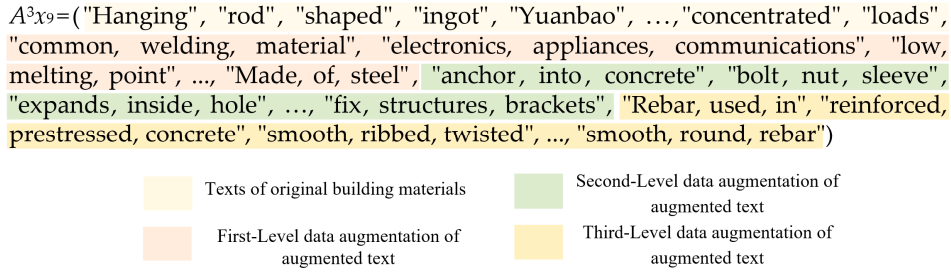

Figure 5. Example of data augmentation.

3.3. Word Embedding

In this study, a pre-trained Bert-base-Chinese model, a version of the popular dynamic word-embedding model BERT, is utilized to transform the expanded text A^3x_i into word vectors. The BERT model is pre-trained on a bidirectional language on text using the Masked Language Model and Neighborhood Sentence Prediction techniques. This training process enriches text semantic representations with intricate semantic information derived from a large corpus. Moreover, the BERT model considers the contextual nuances of word meanings within an entire text, offering the ability to generate word vectors that vary based on the surrounding words [25].

The augmented building materials text A^3x_i is processed using the pre-trained Bert-base-Chinese model, specifically designed for Chinese language processing. Employing the bidirectional capabilities of the transformer architecture, BERT produces an array P_{token} consisting of n tokens, where each word of Ax_i regarded token serves as the fundamental input unit for the model. The process of transforming text into word vectors is illustrated in Figure 6. Automatic padding with pad characters takes place when the text length is insufficient. Each token possesses a dimensionality of v, signifying the existence of v features. The resulting array P_{token} is represented according to Equation (1). The BERT-based word vectors enhance the model's ability to extract contextual information, which is crucial for representing semantic nuances in building materials text.

$$P_{token} = (T_1, T_2, T_3, \ldots\ldots T_n) \tag{1}$$

where

$$T_i = (F_1, F_2, F_3, \ldots\ldots F_f) \tag{2}$$

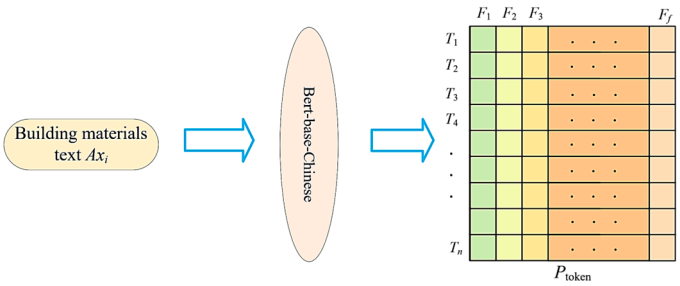

Figure 6. Flowchart depicting the conversion of text into a word vector array.

3.4. Feature Extraction and Aggregation for Full Text

3.4.1. Convolutional Calculations

A CNN network in which the convolutional layers are equipped with X filters with a size of 1**Fs* is constructed; with each pass over the array P_{token}, it extracts the feature information of a vector from the array, capturing the global features of the text to varying degrees. Additionally, the number of channels in the convolutional kernel is configured as X. Feature extraction is accomplished by the filters of each channel, generating weighted output values. Subsequently, these output values are passed through an activation function for nonlinear transformation, aiming to obtain a more comprehensive feature representation.

Each filter is designed to aggregate v local features from each token, consolidating them into a global feature.

In the BERT model, the self-attention mechanism allows each token in a sequence to express information from the entire text based on this information's contextual meaning. When applying the filter, each token's feature vector is multiplied by a corresponding filter value. This process enables the filter to aggregate the feature information from each token's vector, effectively capturing the token's relevance relative to the entire text. Consequently, the resulting array represents a comprehensive global feature representation of the text. As demonstrated by Equations (3) and (4), when the P_{token} array passes through a filter, the array of C_{row} rows and C_{col} columns is exported by the filter, which contains the global features pertinent to each token of the P_{token} array. After the features of the array are extracted by all the filters, X feature arrays representing global features are obtained.

$$C_{\text{row}} = \frac{n - Fh + 2p}{s} + 1 \tag{3}$$

$$C_{\text{col}} = \frac{v - Fs + 2p}{s} + 1 \tag{4}$$

where Fh is the filter height; Fs is the filter width, which is equal to the dimensions of P_{token}; p is the padding value; and s is the stride value.

3.4.2. Aggregation Features

The feature arrays output by each filter are aggregated together to augment the semantic expression within the building materials text. The aggregation output at this stage is an array with C'_{row} rows and C'_{col} columns, as expressed in Equations (5) and (6). The aggregated array P_{agg} is shown in Figure 7, in which the global features of building material texts are expressed more comprehensively by vectors.

$$C'_{\text{row}} = C_{\text{row}} \tag{5}$$

$$C'_{\text{col}} = C_{\text{col}} X \tag{6}$$

$$P_{\text{agg}} = (T_{\text{agg}1}, T_{\text{agg}2}, T_{\text{agg}3}, \ldots\ldots T_{\text{agg}C'_{\text{row}}}) \tag{7}$$

where

$$T_{\text{agg}i} = (F_1, F_2, F_3, \ldots\ldots F_{C'_{\text{col}}}) \tag{8}$$

$$F_i = f(w \cdot T_i + b) \tag{9}$$

where f is a nonlinear activation function. w is the parameter matrix of the filter. b is the bias.

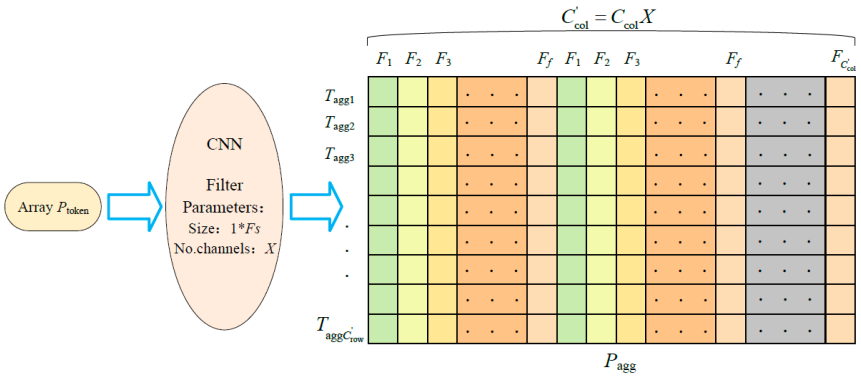

Figure 7. Array after aggregation of features.

3.5. Feature Extraction for Pre- and Post-Semantics

In the previous section, the full-text features of building materials were extracted and then aggregated by the convolutional computation. In this stage, the local features of building materials text are extracted using a parallel multi-CNN network. Convolutional kernels of different sizes are used to extract the semantic relationships of neighboring words at different granularities, enabling the model to learn various features more comprehensively, reducing the dependence on a single feature extraction method and improving the robustness of the model.

The parallel multi-CNN feature extraction process is shown in Figure 8. The first CNN, CNN1, with a filter size of Fh_1*Fw (where $Fh_1 = 2$), emphasizes the semantic relationship between neighboring words in the building materials text. The second CNN, CNN2, with a filter size of Fh_2*Fw (where $Fh_2 = 3$), extracts expressive features from every three words, highlighting their contextual meanings. The third CNN, CNN3, with a filter size of Fh_3*Fw (where $Fh_3 = 4$), extracts the optimal features that reveal the contextual meanings of every four words. All three pooling layers have a size of Fh_{p}*Fw, with $Fh_{\text{p}} = 1$.

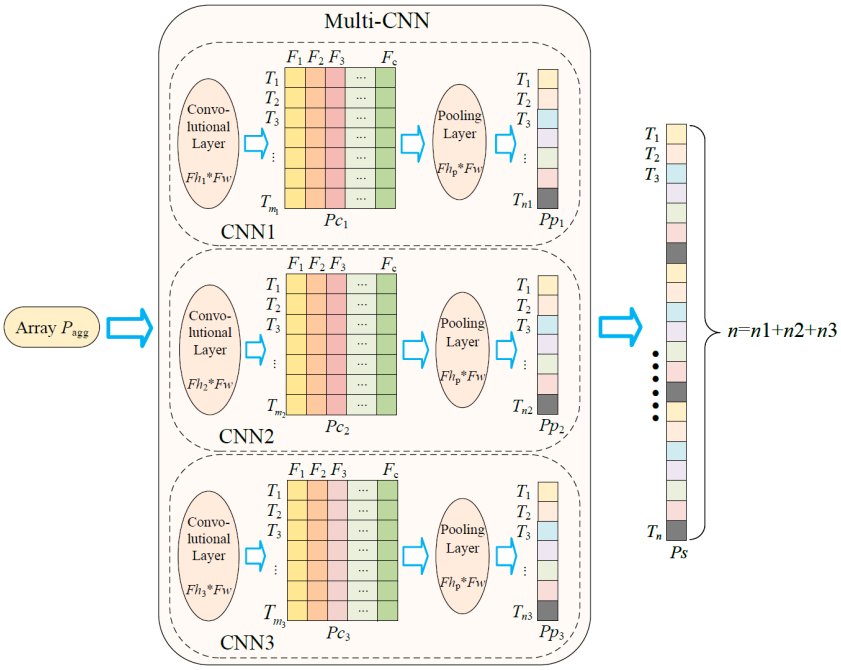

Figure 8. Multi-CNN feature extraction.

The array P_{agg} is simultaneously handled by the convolutional layers of CNN1-CNN3 in parallel style, and the corresponding output array is Pc_i, with m_i rows and c columns, where i ranges from 1 to 3, as expressed in Equations (10)–(12). Subsequently, the outputs Pc_i are fed to the respective pooling layers, generating arrays Pp_i with n_j rows and one column, where i ranges from 1 to 3. Now, both the semantic relationships and contextual meanings in the building materials text can be accurately extracted through the above process.

$$m_i = \frac{C'_{\text{row}} - Fh_i + 2p}{s} + 1 \quad i = 1, 2, 3 \tag{10}$$

$$c = \frac{C'_{\text{col}} - Fw + 2P}{s} + 1 \tag{11}$$

$$Pc_i = [T_1, T_2, T_3, \ldots\ldots, T_{m_i}] \quad i = 1, 2, 3 \tag{12}$$

where

$$T_i = [F_1, F_2, F_3, \ldots\ldots, F_c] \quad i = 1, 2, 3 \tag{13}$$

$$n_j = \frac{m_i - Fh_p + 2p}{s} + 1 \quad i, j = 1, 2, 3 \tag{14}$$

$$Pp_i = [T_1, T_2, T_3, \ldots\ldots, T_{n_i}] \quad i = 1, 2, 3 \tag{15}$$

where Fw is the width of the convolutional and pooling layers, equal to the dimensions of the input array; p is the padding value; and s is the step value.

The output Pp_i of the pooling layer is concatenated into the array Ps, as expressed in Equation (16). Since the extraction of local features is performed based on global feature extraction, the array Ps encompasses both the global and local features of neighboring words. The text feature information extracted from the building materials is aggregated, providing comprehensive integrated features for the subsequent task of building material classification.

$$\begin{aligned} Ps &= [T_1, T_2, T_3, \ldots\ldots, T_{n_1}, T_1, T_2, T_3, \ldots\ldots, T_{n_2}, T_1, T_2, T_3, \ldots\ldots, T_{n_3}] \\ &= [T_1, T_2, T_3, \ldots\ldots, T_n] \end{aligned} \tag{16}$$

where $n = n1 + n2 + n3$.

3.6. Classification Output

The text array Ps, which is created through data augmentation and feature extraction for a certain building material, is used as an input for the fully connected layer, as depicted in Figure 9.

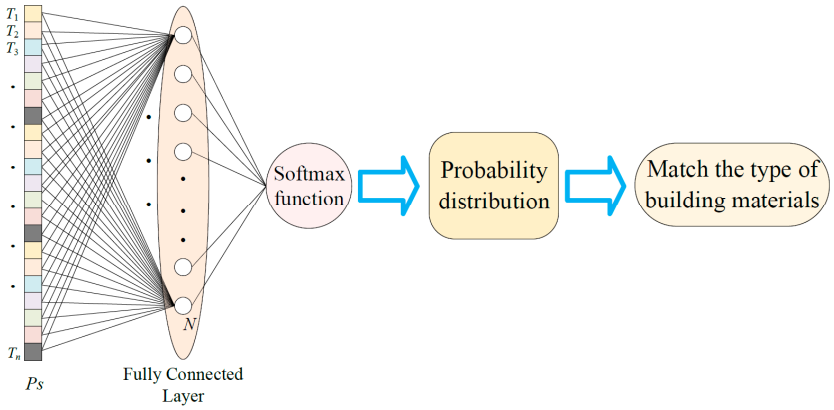

Figure 9. Classification stages of fully connected layers.

The rectified linear unit (ReLU) function is applied for nonlinear transformation in the fully connected layer. The fully connected layer is composed of N neurons, where N represents the number of categories for the three-level classification of all building materials. Each neuron contributes to the output, which is then subjected to the *Softmax* function for multi-classification task activation. This process yields a probability distribution for building materials text across all three levels of classification, as denoted in Equation (17). The classification with the highest probability within this distribution identifies the type of material for the input building materials in the three-level classification system.

$$y_i = Softmax(w_i \cdot P_s + b_i) i \in [1, N] \tag{17}$$

where

$$\sum_{i=1}^{N} y_i = 1 \tag{18}$$

where w_i is the weight parameter in a neuron and b_i is the bias.

4. Experiments and Discussion

4.1. Comparisons and Experimental Environment

The configuration of the experimental environment is as follows: the operating system was Windows 10, the programming language was Python 3.6, and Tensorflow 1.14.0 was used as the deep learning framework. The hardware used included an i5-9300H processor clocked at 4.1 GHz, 32 GB of RAM, and an NVIDIA GeForce 1060 graphics card. The building materials text dataset was divided into training, testing, and validation sets in a ratio of 6:2:2.

The experiment parameters comprise a learning rate of 0.00001, a maximum text length of 256, 768 word vector dimensions, a batch size of 16 for input sentences in each round, ten training rounds, and convolutional window sizes of 3, 4, and 5 for the composite CNN network. The activation function is ReLU, and the optimizer is Adam. The configuration of the model is specified in Table 2.

Table 2. The detailed configuration of parameters.

Parameter	Value
Learning rate	0.00001
Max_len	256
Dimensions of a word vector	768
Batch_size	16
Epochs	10
Convolution window size	2, 3, 4
Activation function	Relu
Optimizer	Adam

4.2. Evaluation Indicators

In the text classification of building materials, there is an imbalance in the number of samples for each material type. This can lead to certain material types dominating the training process, thus affecting a model's predictive performance for other material types. The overall evaluation metrics tend to favor material types with higher sample sizes and ignore the performance of material types with lower sample sizes. Therefore, the macro average precision, the macro average recall, and the macro average F1 score were chosen as evaluation criteria for this experiment. Macro precision measures the average prediction accuracy across all classes, reflecting a model's precision. Macro recall evaluates the average detection capability across all classes, indicating a model's coverage ability. The macro F1 score assesses the average balance between precision and recall, providing a comprehensive performance evaluation.

$$P_{Macro} = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{19}$$

$$R_{Macro} = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{20}$$

$$F1_{Macro} = \frac{2 \times P_{Macro} \times R_{Macro}}{P_{Macro} + R_{Macro}} \tag{21}$$

where P_i and R_i denote precision and recall for type i building materials, and P_{Macro}, R_{Macro}, and $F1_{\text{Macro}}$ denote macro average precision, the macro average recall, and the macro average $F1$.

4.3. Comparative Experiments

In order to verify the efficiency of the proposed method, traditional machine learning models, such as SVMs [26], KNN [27], and Naive Bayes [28], and deep learning models and their variants, including CNNs [29], LSTM [30], BERT-CNN [31], LSTM-CNN [32], and LDA-Ngram-BERT-LSTM, were used for comparison.

Moreover, the experiments were also conducted using the proposed method LNBC and the variants with different numbers of parallel multi-CNNs, abbreviated as LNBC (none), LNBC (2.3), LNBC (2.4), and LNBC (3.4). The experimental results for the test set are presented in Table 3.

In order to more intuitively reflect the advantages of the proposed method, a visualization of the result is depicted in Figure 10.

Furthermore, the results regarding execution time are given in Figure 11 to compare the computational efficiency of LNBC with that of the other models.

Table 3. Comparison of the experimental results.

Model	Evaluation Indicators			Model	Evaluation Indicators		
	P_{Macro} (%)	R_{Macro} (%)	$F1_{Macro}$ (%)		P_{Macro} (%)	R_{Macro} (%)	$F1_{Macro}$ (%)
SVM	69.13	73.72	71.35	LDA-Ngram-BERT-LSTM	73.91	81.02	77.30
KNN	65.78	72.26	68.87				
Naive Bayes	68.96	73.72	71.26	LNBC (none) *	75.54	80.29	77.84
CNN	71.97	75.91	73.89	LNBC (2.3) *	77.93	83.21	80.48
LSTM	71.66	75.18	73.38	LNBC (2.4) *	73.92	82.48	77.97
BERT-CNN	73.58	77.37	75.43	LNBC (3.4) *	78.13	83.94	80.93
LSTM-CNN	70.68	78.10	74.20	LNBC *	78.89	83.94	81.33

* The proposed model is referred to as LNBC. The variant LNBC (none) denotes the model without the local feature extraction phase. Variants LNBC (2.3), LNBC (2.4), and LNBC (3.4) indicate the models that exclusively extract local features from sequences of two and three, two and four, and three and four words during the local feature extraction phase.

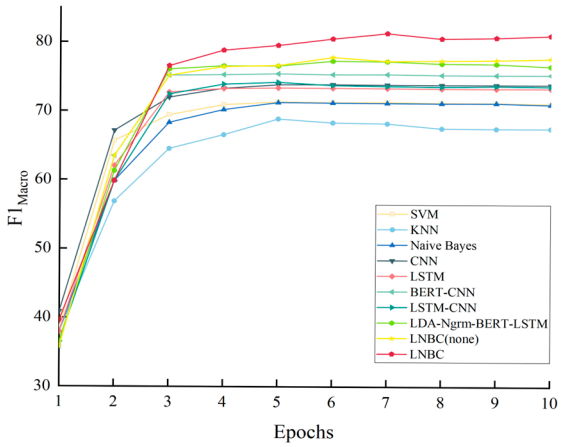

Figure 10. $F1_{Macro}$ with the number of iterations.

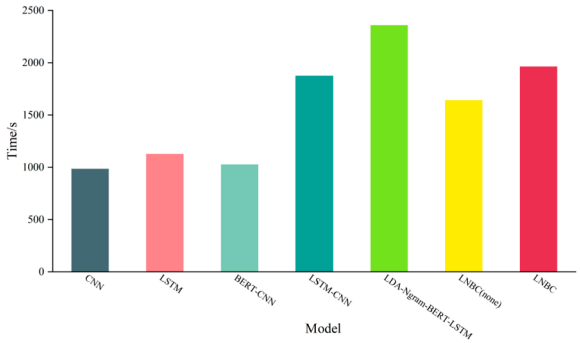

Figure 11. Training times of various models.

4.4. Analysis of the Experimental Results

The experimental results in Table 3 demonstrate the superior performance of the proposed LNBC model in building materials text classification, achieving an $F1_{Macro}$ of 81.33%. Even if local feature extraction was not conducted exactly in accordance with the proposed model, i.e., LNBC (none), the results are significantly superior to those yielded by the traditional machine learning models and most of the deep learning models.

Overall, the classification performance of the deep learning model is superior to that of traditional machine learning models, such as SVMs, KNN, and Naive Bayes. In particular, compared with the proposed LNBC model, the machine learning model reduces the $F1_{Macro}$ by 9.98%, 12.46%, and 10.07%, respectively. This performance gap primarily arises because traditional machine learning models employ shallow structures, which limit feature-learning capabilities and involve cumbersome training processes. Then,

these models struggle to effectively extract complex semantic relationships in building materials text. Furthermore, these models cannot effectively process contextual features in building materials text rich in technical terms and descriptions. This deficiency prevents them from fully exploring potential feature connections. In addition, the LNBC model is based on LDA, N-gram, BERT, and CNN deep learning models for the specific learning scenario of building material classification, models that can capture the features of text more comprehensively and accurately and improve classification performance. In terms of data sources, the LNBC model discovers the topics in the text through LDA, which enriches the semantic information of the data and indicates the classification performance. Meanwhile, generating more text fragments through Ngram enhances the model's ability to capture contextual information, improves data diversity, and realizes data enhancement. In terms of data processing, the word vectors generated by BERT can determine the word–context relationship. Combined with different CNNs, the global features of the text and the local features between neighboring words are extracted, thus improving the effect of building materials text classification.

On the other hand, it can be clearly observed that the proposed method outperforms the deep learning methods. For instance, the P_{Macro}, the R_{Macro}, and the $F1_{\text{Macro}}$ of the proposed LNBC increased by 7.23%, 8.76%, and 7.55% compared with those of the LSTM, one of the most popular deep learning methods. The main reasons are, first, that the records of building materials data in engineering are characterized by insufficient accuracy and incomplete information. These deficiencies restrict the learning ability of deep learning models such as LSTM. Second, LSTM models usually perform well when dealing with sequential data with long-term dependencies, but the features of building materials text are primarily independent or concentrated in a certain part of a sentence. Consequently, the contextual feature information captured by LSTM is disorganized and lacks coherence. In addition, the proposed model is normalized for unstandardized building materials text. The LDA-Ngram model is used to expand the building materials text data to enhance the diversity of the training data. The BERT model is used to transform word vectors so that each word vector corresponds to the information of the whole text to a different degree, enhancing this model's ability to extract profound semantic information and the local relevance of a sentence. The proposed model utilizes different convolution layers to extract global and local features, resulting in richer feature representation. Different sizes of convolutional kernels also impact the classification effect during the local feature extraction stage. For a declarative sentence in building materials text, kernel sizes of 2, 3, and 4 optimally extract local features from sequences of two, three, and four consecutive words, respectively. This approach is well suited to the length and structure of building materials text, resulting in optimal classification outcomes.

Figure 10 illustrates the variation trend of the $F1_{\text{Macro}}$ scores of various types of models with respect to the test set, showing that the LNBC model stands out by achieving convergence starting from the fourth epoch, maintaining stability, and consistently outperforming the other models in subsequent epochs.

Figure 11 illustrates the time consumption for the deep learning models, and it is shown that the LDA-Ngram-BERT-LSTM model exhibits the longest time, while the CNN model has the shortest. This discrepancy can be attributed to the inherent complexity of LSTM's structure, which is less amenable to parallel processing. In contrast, the simplicity of the CNN model enables a more efficient training process. The time consumption of the LNBC model falls between these extremes. Although integrating composite convolution into the traditional CNN architecture increases complexity, thereby extending training time, the corresponding performance benefits and superior classification outcomes justify the investment in time.

5. Conclusions and Future Work

To accurately calculate the carbon emissions of buildings, in this study, we propose a building materials text classification model operating via data enhancement and layered

feature extraction. Experimental datasets obtained from an engineering project list were investigated and preprocessed. Subsequently, data enhancement was performed on the building materials texts classified under different types using LDA and N-gram models. Word vectors were then generated using the BERT model. Subsequently, both the global and the local features of the building materials texts were extracted via designing different convolutional kernels. Finally, the classification model was employed to match building materials with their respective material types. Compared to the traditional machine learning and deep learning models, the proposed model demonstrated superior classification performance.

In future work, a modular component will be developed using the proposed LNBC model and embedded into a carbon emission assessment platform. Additionally, some new techniques, e.g., transfer learning, optimization methods, and data-processing methods, will be applied to further improve the performance of the LNBC model in order to enhance training efficiency and extend the application scenarios.

Author Contributions: Conceptualization, Q.Y.; formal analysis, Q.Y.; methodology, Q.Y.; software, F.J.; validation, F.J.; visualization, F.J.; writing—original draft, F.J.; writing—review and editing, W.P.; supervision, W.P.; project administration, W.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (61903226, 62173216), the Key Research and Development Program of Shandong Province (2021CXGC011205), Shandong Provincial Science, Technology SME Innovation Capability Improving Project (2022TSGC2157), and the open Foundation of the Anhui Province Key Laboratory of Intelligent Building & Building Energy Saving (IBES2024KF01).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare that there are no conflicts of interest.

References

1. Wei, Y.; Chen, K.; Kang, J.; Chen, W.; Zhang, X.; Wang, X. Policy and management of carbon peaking and carbon neutrality: A literature review. *Engineering* **2022**, *14*, 52–63. [CrossRef]
2. China Association of Building Energy Efficiency. China Building Energy Consumption and Carbon Emissions Research Report. 2022. Available online: https://finance.sina.com.cn/tech/roll/2023-03-12/doc-imykpzhc2296343.shtml (accessed on 15 May 2024).
3. Standard for Terminology of Building Materials. 2010. Available online: https://www.doc88.com/p-7768454939608.html (accessed on 9 June 2024).
4. Jing, X.; Wu, Z.; Zhang, L.; Li, Z.; Mu, D. Electrical fault diagnosis from text data: A supervised sentence embedding combined with imbalanced classification. *IEEE Trans. Ind. Electron.* **2024**, *71*, 3064–3073. [CrossRef]
5. Garg, M. WELLXPLAIN: Wellness concept extraction and classification in Reddit posts for mental health analysis. *Knowl. Based Syst.* **2024**, *284*, 111228. [CrossRef]
6. Tufchi, S.; Yadav, A.; Ahmed, T. A comprehensive survey of multimodal fake news detection techniques: Advances, challenges, and opportunities. *Int. J. Multimed. Inf. Retr.* **2023**, *12*, 28. [CrossRef]
7. Kim, Y. Convolutional neural networks for sentence classification. *EMNLP* **2014**, 1746–1751. [CrossRef]
8. Aslan, S. A deep learning-based sentiment analysis approach (MF-CNN-BILSTM) and topic modeling of tweets related to the Ukraine-Russia conflict. *Appl. Soft Comput.* **2023**, *143*, 110404. [CrossRef]
9. Lu, G.; Liu, Y.; Wang, J.; Wu, H. CNN-BiLSTM-Attention: A multi-label neural classifier for short texts with a small set of labels. *Inf. Process Manag.* **2023**, *60*, 103320. [CrossRef]
10. Zhong, B.; Xing, X.; Love, P.; Wang, X.; Luo, H. Convolutional neural network: Deep learning-based classification of building quality problems. *Adv. Eng. Inform.* **2019**, *40*, 46–57. [CrossRef]
11. Abulaish, M.; Sah, A.K. A Text Data Augmentation Approach for Improving the Performance of CNN. *Comsnets* **2019**, 660–665. [CrossRef]
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
13. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805. [CrossRef]

14. Bao, T.; Ren, N.; Luo, R.; Wang, B.J.; Shen, G.Y.; Guo, T. A BERT-Based Hybrid short text classification model incorporating CNN and Attention-Based BiGRU. *J. Organ. End User Comput.* **2021**, *23*, 21. [CrossRef]
15. Liu, S.; Liu, S.; Liu, Z.; Peng, X.; Yang, Z. Automated detection of emotional and cognitive engagement in MOOC discussions to predict learning achievement. *Comput. Educ.* **2022**, *181*, 104461. [CrossRef]
16. Li, Y.; Trevor, C.; Timothy, B. Robust training under linguistic adversity. *EACL* **2017**, *2*, 21–27.
17. Marivate, V.; Sefara, T. Improving short text classification through global augmentation methods. *Mach. Learn. Knowl. Extr.* **2020**, *4*, 385–399. [CrossRef]
18. Sahin, G.; Steedman, M. Data augmentation via dependency tree morphing for low-resource languages. *arXiv* **2018**, arXiv:1903.09460. [CrossRef]
19. Kuniyoshi, F.; Ozawa, J.; Miwa, M. Analyzing research trends in inorganic materials literature using NLP. *arXiv* **2021**, arXiv:2106.14157. [CrossRef]
20. Song, Y.; Miret, S.; Liu, B. MatSci-NLP: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023; Volume 1, pp. 3621–3639. [CrossRef]
21. Elton, D.C.; Turakhia, D.; Reddy, N.; Boukouvalas, Z.; Fuge, M.D.; Doherty, R.M.; Chung, P.W. Using natural language processing techniques to extract information on the properties and functionalities of energetic materials from large text corpora. In Proceedings of the 22nd International Seminar in New Trends in Research of Energetic Materials, Pardubice, Czech Republic, 10–12 April 2019. [CrossRef]
22. Yoshitake, M.; Sato, F.; Kawano, H.; Teraoka, H. Materialbert for natural language processing of materials science texts. *Sci. Technol. Adv. Mater.* **2022**, *2*, 372–380. [CrossRef]
23. Turhan, G.D. Life Cycle Assessment for the Unconventional Construction Materials in Collaboration with a Large Language Model. In Proceedings of the International Conference on Education and Research in Computer Aided Architectural Design in Europe, Graz, Austria, 20–22 September 2023.
24. China Products Carbon Footprint Factors Database. Available online: https://lca.cityghg.com/ (accessed on 15 May 2024).
25. El-Rashidy, M.; Farouk, A.; El-Fishawy, N.; Aslan, H.; Khodeir, N. New weighted BERT features and multi-CNN models to enhance the performance of MOOC posts classification. *Neural Comput. Appl.* **2023**, *35*, 18019–18033. [CrossRef]
26. Al-Fuqaha'a, S.; Al-Madi, N.; Hammo, B. A robust classification approach to enhance clinic identification from Arabic health text. *Neural Comput. Appl.* **2024**, *36*, 7161–7185. [CrossRef]
27. Huang, A.; Xu, R.; Chen, Y.; Guo, M. Research on multi-label user classification of social media based on ML-KNN algorithm. *Technol. Forecast. Soc. Chang.* **2023**, *188*, 122271. [CrossRef]
28. Berkin, A.; Aerts, W.; Van Caneghem, T. Feasibility analysis of machine learning for performance-related attributional statements. *Int. J. Account. Inf. Syst.* **2023**, *48*, 100597. [CrossRef]
29. Luo, X.; Li, X.; Song, X.; Liu, Q. Convolutional neural network algorithm-based novel automatic text classification framework for construction accident reports. *J. Constr. Eng. Manag.* **2023**, *149*. [CrossRef]
30. Gu, D.; Li, M.; Yang, X.; Gu, Y.; Zhao, Y.; Liang, C.; Liu, H. An analysis of cognitive change in online mental health communities: A textual data analysis based on post replies of support seekers. *Inform Process Manag.* **2023**, *60*, 103192. [CrossRef]
31. Hasib, K.; Towhid, N.; Faruk, K.; Al Mahmud, J.; Mridha, M. Strategies for enhancing the performance of news article classification in Bangla: Handling imbalance and interpretation. *Eng. Appl. Artif. Intel.* **2023**, *125*, 106688. [CrossRef]
32. Yilmaz, S.; Toklu, S. A deep learning analysis on question classification task using Word2vec representations. *Neural Comput. Appl.* **2020**, *32*, 2909–2928. [CrossRef]