Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 10
ISET 2025

Research on a Chinese Text Information Density Evaluation Model Fusing Semantic and Statistical Features

Zhaoyang Ye

Television School, Communication University of China, Beijing, China

285432131@qq.com

Abstract. To address the issue in Chinese text information content evaluation where traditional methods primarily rely on statistical features and overlook semantic and structural complexity, this study proposes a Chinese text information density evaluation model that fuses semantic and statistical features. The model adopts a dual-channel fusion architecture: the semantic channel utilizes the pre-trained language model BERT to extract deep contextual embeddings of the text, combined with a Bidirectional Long Short-Term Memory network (BiLSTM) to capture long-range semantic dependencies; the statistical channel integrates Term Frequency-Inverse Document Frequency (TF-IDF), part-of-speech (POS) distribution, and dependency relations. These two types of heterogeneous features are concatenated and then fed into a fusion gating module for effective combination and non-linear interaction. Finally, a regression layer outputs a standardized information density score. For model training and evaluation, this study operationalized the definition of information density and constructed a manually annotated dataset comprising 210 diverse Chinese texts. Experimental results demonstrate that the proposed model significantly outperforms various baseline models across all evaluation metrics, validating the effectiveness and superiority of the fusion model for the task of Chinese text information density evaluation. This research provides a new analytical tool for applications such as text quality assessment and high-value information localization.

Keywords: Information Density; Chinese Text Evaluation; Semantic Features; Statistical Features; Deep Learning; Fusion Model.

1. Introduction

According to Shannon's "A Mathematical Theory of Communication", information can be defined as the measure of uncertainty that arises when a specific message is selected from a set of possible messages [1], which laid the foundation of information theory. Beyond this technical notion of "information," some communication scholars have turned elements of information theory into tools for studying mass-communication phenomena, such as quantifying communication chains and analyzing the structure and efficiency of communication networks through information-theoretic models [2].

Several studies have applied information theory directly to practical scenarios. Qian Chen and Huang Weidong introduced a "super-information" metric based on information quantity and built a Weibo event situation-awareness model that effectively filters noise and extracts key inflection points [3]. Wang Zheng et al. proposed the Message-Based Information Density (MBID) model for detecting emerging topics, achieving efficient monitoring via a dynamic sliding window and a topic-tree structure [4]. Other investigations focus on the role of information in linguistic communication. Crocker et al.'s IDeaL project employs an information-theoretic framework to explain the relationship among linguistic variation, cognitive processing, and communicative efficiency [5]. Liu et al. use information-theoretic methods — specifically the Kolmogorov-complexity framework — to measure the linguistic complexity of Mandarin Chinese [6].

However, most existing studies rely on a single metric or dimension, and few models capture the richness of information conveyed within Mandarin's flexible word order. The present study transcends the one-dimensional word-frequency paradigm by adopting a dual-channel semantic–

statistical fusion architecture for assessing information density: multidimensional information quantification is achieved through the synergy of a pre-trained language model and multiple statistical features. In the semantic channel, contextual embeddings obtained via a BERT tokenizer are processed by a bidirectional LSTM to capture long-range semantic dependencies; in the statistical channel, TF-IDF weights, part-of-speech distributions, and dependency features are integrated simultaneously. The two heterogeneous feature sets are concatenated and fed into a gated fusion module, after which a regression layer outputs a normalized information-density score. This framework provides a quantifiable tool for locating high-value information in retrieval systems, analyzing the features of misinformation, and assessing the quality of knowledge-base content.

2. Theoretical Framework for Quantifying Information Density in Chinese Text

2.1. Textual Feature Representation and Information Quantification

The division of linguistic components from a linguistic perspective offers the theoretical basis for modeling textual features. Traditional linguistics identifies three fundamental elements of language — phonology, lexicon, and grammar. When focusing solely on written text, the lexical and grammatical layers are generally considered the primary constituents. Accordingly, the present study seeks to capture the full range of lexical and grammatical features by constructing a dual-channel semantic–statistical fusion model.

For quantifying textual information, natural language can be treated as a stochastic process, allowing the entropy of each character to be computed. Using this approach, Feng Zhiwei calculated the average information entropy of Chinese as 9.65 bits [7]. Although this method measures textual entropy scientifically from the standpoint of character encoding, it shows limitations when the research turns to communicative activities or content-specific analysis: measuring a symbol's information solely by its unpredictability within the entire text neglects semantic complexity.

Therefore, this study introduces the concept of information density and builds an information-density evaluation model via a dual-channel semantic–statistical fusion. For a given passage, the amount of information contained per text unit is determined by the cognitive load incurred per unit time while humans process that information. Rather than relying on a fixed, predefined mathematical formula to compute information density directly, the proposed model employs machine learning to capture semantic and statistical features within texts of varying information densities. Once training is complete, the model's learned weights can accurately reflect how different features contribute to overall textual density[8].

2.2. Semantic Representation Methods

In the field of natural language processing, numerous approaches have been developed to quantify and represent textual semantics.

Traditional semantic processing relies on rule-based or statistical models. Miller's WordNet, for example, is a dictionary-based semantic network that expresses word meanings through synonym sets and semantic relations [9]. Latent Semantic Analysis (LSA) extracts latent semantic dimensions by performing matrix decomposition on a word-document co-occurrence matrix [10]. These classic methods struggle with polysemy, depend heavily on hand-crafted features, and exhibit limited generalization capability.

With advances in machine learning, new representation techniques have emerged. Word2Vec introduced the Skip-gram and CBOW models, encoding semantics as dense vectors [11]. Subsequent work refined word-vector quality by analyzing ratios of word co-occurrence probabilities, thereby capturing both semantic and syntactic relationships more effectively [12].

The attention mechanism of the Transformer architecture overcame the bottleneck of earlier methods that could not capture long-range dependencies. ELMo (Embeddings from Language Models)

leverages attention by employing a bidirectional language model (biLM) to capture both the semantics of words and the contextual polysemy simultaneously [13]. BERT (Bidirectional Encoder Representations from Transformers) further advances the field through large-scale unsupervised pre-training to learn universal semantic representations, which are later fine-tuned for downstream tasks [14]. BERT unifies the NLP task framework and markedly enhances few-shot learning performance; therefore, this study adopts BERT to extract semantic information from Chinese texts.

2.3. Statistical Feature Representation Methods

For representing statistical features of text, prior work covers multiple aspects such as word-occurrence frequency, part-of-speech information, and dependency relations. The present study adopts mature techniques from existing research to extract these statistical features.

TF-IDF (Term Frequency–Inverse Document Frequency) is a classic statistical approach in information retrieval and text mining for measuring a word's importance within a document collection. Term Frequency (TF) assumes that the more often a word appears in a document, the more important it is to that document. Inverse Document Frequency (IDF) discounts words that appear broadly across the corpus, as their discriminative power is lower and their importance should therefore be reduced [15, 16].

For part-of-speech features, POS tagging accurately reflects the grammatical roles of words within the overall text. Large annotated treebanks enable robust statistical models of syntax; notably, the 1989 Penn Treebank standardized both the tag set and the annotation guidelines for POS labeling [17, 18].

Dependency relations describe syntactic links among words in a sentence, such as subject-predicate or verb-object relations. Existing work has established a cross-lingual unified set of dependency labels [19]. Deep-learning-based dependency parsers — e.g., Baidu's open-source DDParser — can accurately annotate dependency relations in Chinese text [20].

3. Model Architecture Design

This study proposes a hybrid neural-network architecture for evaluating the information density of Chinese text. The core design principle is to combine deep semantic understanding with linguistic features, thereby comprehensively capturing the multidimensional textual characteristics that influence information density (see Figure1).

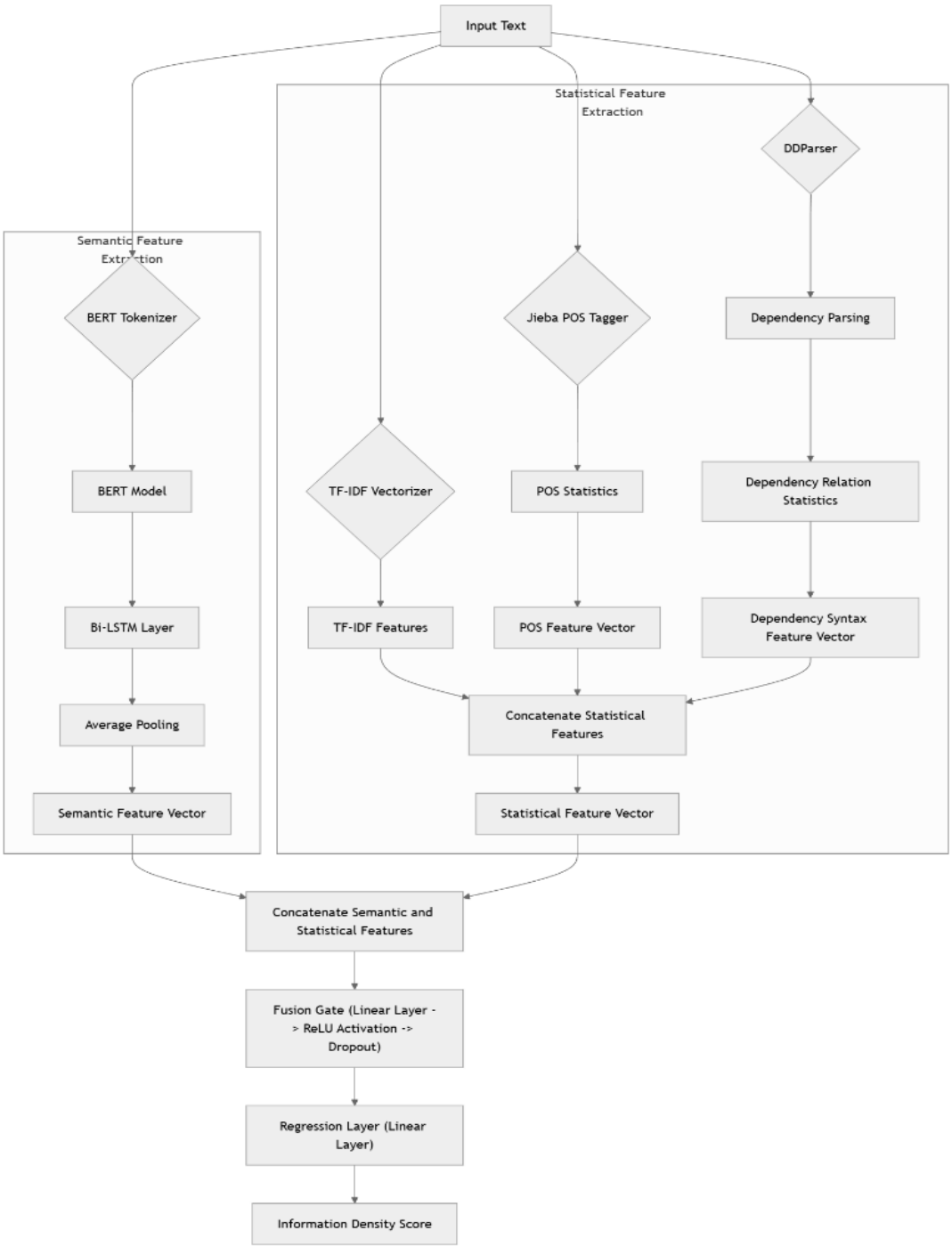

Figure 1. Model Architecture Design

3.1. Semantic Feature Extraction

The goal of this stage is to convert the raw input text into vector representations that are rich in contextual semantic features. Firstly, the input batch of text is processed through tokenization. The tokenizer used is the one that comes with the pre-trained BERT model. It divides the text into tokens and transforms them into a sequence of word embedding vectors that the model can accept.

Subsequently, the processed inputs are sent into the BERT model. The BERT model generates a context-related hidden state for each token in the sequence through its multi-layer self-attention mechanism. The output H_{BERT} of BERT is a three-dimensional tensor, with dimensions defined by the parameters during training as (*batch_size, sequence_length,* d_{bert_hidden})), where (d_{bert_hidden}) is the hidden layer dimension of the BERT model.

To further capture and integrate the sequential dependencies and longer-range context of a sequence, the model introduces a Bidirectional Long Short-Term Memory (BiLSTM) layer based on this foundation. The BiLSTM consists of a forward LSTM and a backward LSTM, allowing it to process sequence information simultaneously from both directions. The H_{BERT} sequence is fed into the BiLSTM layer, and the output of the BiLSTM, H_t^{LSTM}, can be represented as $h_t^{LSTM} = [\vec{h}_t, \overleftarrow{h}_t]$, where $\vec{h}_t$ and $\overleftarrow{h}_t$ are the hidden states of the forward and backward LSTMs at time step t, respectively.

To obtain a fixed-length semantic feature vector that can appropriately represent the entire input text, the model employs Mean Pooling. Mean Pooling averages all hidden state vectors along the sequence dimension. This operation is performed along the sequence length dimension, denoted as $(\ sequence\_length\ )$ (dim=1), and the formula can be expressed as:

$$V_{semantic} = \frac{1}{sequence\_length} \sum_{t=1}^{sequence\_length} h_t^{LSTM}$$

Through the aforementioned stages, the model is able to gradually extract and abstract high-level semantic representations from the original text, providing crucial semantic dimension input for accurately assessing the information density of the text.

3.2. Statistical Feature Extraction

The goal of this stage is to extract multi-dimensional statistical features from the original input text and integrate them into a comprehensive statistical feature vector. These features aim to capture the statistical characteristics of the text at different granularities, mainly including TF-IDF features, part-of-speech (POS) features, and dependency (DEP) features.

3.2.1. TF-IDF Features

TF-IDF features are used to evaluate the importance of terms for a single text and the entire corpus. The model implements this using the TF-IDF vectorizer from sklearn. The term frequency (TF) is calculated as follows:

$$tf(t_j, d) = f_{t_j,d}$$

$f_{t_i,d}$ is the number of times the term t_j appears in the document d.

The calculation method for Inverse Document Frequency (IDF) is as follows:

$$idf(t_j, D) = \log \frac{N_D + 1}{df(t_j, D) + 1} + 1$$

Here, N_D is the total number of documents in the training corpus D, and $df(t_j, D)$ is the number of documents in the training corpus D that contain the term t_j.

Thus, for the term t_j in document d, its TF-IDF value is the product of the term's TF value and its IDF value:

$$tfidf(t_j, d, D) = tf(t_j, d) \times idf(t_j, D)$$

For a given document d, the unnormalized TF-IDF vector $v'_{tfidf}(d)$ is composed of each term from a vocabulary of size M:

$$v'_{tfidf}(d) = [tfidf(t_1, d, D), tfidf(t_2, d, D), \ldots, tfidf(t_M, d, D)]$$

The TF-IDF vectorizer from sklearn by default normalizes the TF-IDF vector for each document using L2 norm. The resulting TF-IDF feature vector $v_{tidf}(d)$ for a given document d is:

$$v_{tfidf}(d) = \frac{v'_{tfidf}(d)}{\left\| v'_{tfidf}(d) \right\|_2}$$

3.2.2. Part-of-Speech (POS) Features

Let $Count(tag_j, d)$ be the number of times the part-of-speech tag tag_j appears in the text d, and let $N_{words}(d)$ be the total number of words used for part-of-speech statistics in the text d. Then, the j-th element $(v_{pos})_j$ of the part-of-speech feature vector $v_{pos}(d)$ is calculated as follows:

$$\left(v_{pos}\right)_j = \frac{Count\left(tag_j, d\right)}{N_{words}(d)} \ , N_{words}(d) > 0$$

P is the total number of pre-defined part-of-speech categories, and the complete feature vector $v_{pos}(d)$ is expressed as follows:

$$v_{pos}(d) = \left[\frac{Count(tag_1, d)}{N_{words}(d)}, \frac{Count(tag_2, d)}{N_{words}(d)}, \ldots, \frac{Count(tag_P, d)}{N_{words}(d)}\right]$$

3.2.3. Dependency Relation (DEP) Features

Let $Count(rel_k, d)$ be the number of times the dependency relation label rel_k appears in the dependency parsing results of the text d. Let $N_{relations}(d)$ be the total number of relations used for dependency relation statistics in the text d.Then, the k-th element of the dependency feature vector $v_{dep}(d)_k$ is calculated as follows:

$$\left(v_{dep}\right)_k = \frac{Count(rel_k, d)}{N_{relations}(d)} \ , N_{relations}(d) > 0$$

Q is the total number of pre-defined dependency relation categories, and the complete dependency feature vector $v_{dep}(d)$ can be expressed as:

$$v_{dep}(d) = \left[\frac{Count(rel_1, d)}{N_{relations}(d)}, \frac{Count(rel_2, d)}{N_{relations}(d)}, \ldots, \frac{Count\left(rel_Q, d\right)}{N_{relations}(d)}\right]$$

3.2.4. Concatenation of Statistical Features

Finally, the extracted TF-IDF feature vector, part-of-speech distribution feature vector, and dependency relation feature vector are concatenated along the feature dimension to form the final combined statistical feature vector $V_{statistical}$.

$$V_{statistical} = concat(v_{tfidf}, v_{pos}, v_{dep})$$

3.3. Fusion of Features and Regression Layer Score Prediction

After extracting the semantic features $V_{semantic}$ and statistical features $V_{statistical}$ separately, the model needs to effectively combine these two types of features and ultimately map them to the information density score. This process is completed by the fusion gating layer and the regression layer.

3.3.1. Concatenation of Feature Vectors and Fusion Gating Layer

First, the semantic feature vectors $V_{semantic}$ and statistical feature vectors $V_{statistical}$ in the batch are concatenated along the feature dimension ($dim = 1$) to form the input of the fusion layer, denoted as V_{fused_input}.

$$V_{fused\_input} = concat(V_{semantic}, V_{statistical})$$

The main purpose of the fusion gating layer is to learn an effective and nonlinear way to combine semantic features and statistical features. The fusion gate allows the model to automatically learn these interactions and weights through one or more layers of neurons.

The input to the fusion layer first passes through a fully connected linear layer, projecting it into a lower-dimensional space and learning a weighted combination of features. Here, w_{gate}^{T} is the weight matrix of the linear layer, b_{gate} is the bias vector, and the final output is given by H_{linear} .

$$H_{linear} = V_{fused\_input} W_{gate}^{T} + b_{gate}$$

Subsequently, the model introduces non-linearity through a ReLU activation function, enabling the model to learn more complex feature relationships.The definition of the ReLU function is $\mathrm{ReLU}(x) = \max(0, x)$, applied element-wise to H_{linear} .

$$H_{relu} = {}_{ReLU}(H_{linear})$$

To prevent the model from overfitting during the training process, the output after the ReLU activation is passed through a Dropout layer. During the training phase, Dropout randomly sets the output of neurons to zero with a probability specified by a parameter.

$$H_{gate\_out} = {}_{Dropout}(H_{relu})$$

3.3.2. Regression Layer

The feature representation $H_{\text{gate\_out}}$ obtained from the fused gated layer is finally sent to the regression layer. The regression layer is a fully connected linear layer designed to map the fused high-dimensional feature representation to a single continuous scalar value. Here, W_{reg}^{T} is the weight matrix of the regression layer, b_{reg} is the bias term, and the final output score is denoted as *S*.

$$S = H_{gate\_out} W_{reg}^{T} + b_{reg}$$

In this way, the model first combines features from different sources, then refines and transforms these combined features through a learnable gating mechanism, and finally utilizes a linear transformation to output a quantitative assessment of information density. The entire process can be end-to-end trained using the backpropagation algorithm and gradient descent to adjust the model's weights.

4. Experiments and Evaluation

4.1. Dataset Construction

A multidimensional scheme—covering lexical richness and specificity, conciseness of expression, and structural compactness—was devised to guide the manual annotation of information density. A five-point Likert scale (1–5) was adopted as the scoring standard, with clear verbal descriptions and at least three prototypical text examples provided for each level.

A total of 220 Chinese passages were sampled from books and corpora spanning diverse domains, including Classical Chinese prose, everyday dialogue, and specialist materials from multiple fields. After preliminary cleaning (removing malformed, overly short, or excessively long texts), three native-Chinese annotators with at least a bachelor's degree were recruited. All annotators underwent unified training to ensure a consistent understanding of the scoring criteria.

Prior to formal annotation, a pilot round was conducted. Inter-annotator agreement, measured by Krippendorff's Alpha (α), reached 0.77, indicating good consistency and validating the rubric as well as the annotators' shared conception of information density.

During the main annotation phase, the mean of the three annotators' scores served as each passage's initial information-density rating. Samples with a standard deviation greater than 1.5 were flagged for adjudication, in which a researcher reviewed the text and rendered a final decision.

The aggregated scores were linearly transformed to the (0, 1) interval and used as target values for model training. After filtering, the final dataset comprised 210 annotated passages, randomly partitioned into 110 for training, 50 for validation, and 50 for testing.

4.2. Training and Evaluation

The proposed model was trained on the annotated training set described above.

Model configuration: The hidden size of the BiLSTM layer was set to 256. The initial maximum dimensionality of the TF-IDF feature vector was 500 and was dynamically updated during adaptation. The dropout rate of the fusion-gating layer was 0.1.

Training details: All experiments were conducted in a Python 3.9 environment using the PyTorch framework. The optimizer was AdamW, and the loss function was mean squared error loss (MSELoss). The initial learning rate was 1×10^{-5}; the batch size was 8, and the model was trained for 60 epochs on an NVIDIA Tesla P40 GPU.

Validation and model selection: After every epoch, the model was evaluated on the validation set; the checkpoint with the lowest validation loss was retained as the final model. Performance was then assessed on the held-out test set of 50 passages that were never seen during training.

Evaluation metrics: Standard regression metrics were used: root-mean-square error (RMSE), mean absolute error (MAE), and the coefficient of determination (R^2). Mean squared error (MSE) measures the average squared difference between predicted and true values; lower values indicate higher predictive accuracy. RMSE is the square root of MSE and is more sensitive to large errors.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

The Mean Absolute Error (MAE) calculates the average of the absolute differences between predicted values and true values. Unlike Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), MAE gives the same weight to all sizes of errors and is therefore insensitive to outliers. A smaller MAE value indicates higher prediction accuracy of the model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

R^2 represents the degree to which a model explains the variability of the dependent variable. Its values typically range from 0 to 1. The closer R^2 is to 1, the better the model fits the data. An R^2 of 0 indicates that the model performs as poorly as predicting by the mean. R^2 can also be negative, meaning that the model's predictions are worse than simply using the mean.

$$R^2 = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

The performance metrics of the model on the test set are evaluated as shown in Table 1.

Table 1. Model Performance Metrics Evaluation Scores

Performance metrics	Numerical value
$RMSE$	0.1020
MAE	0.0850
R^2	0.7986

The comparison between the model's predicted values and the actual values is shown in Figure 2. It can be seen that the predicted values fluctuate around the actual values, but the overall upward trend aligns with the rising trend of the actual values.

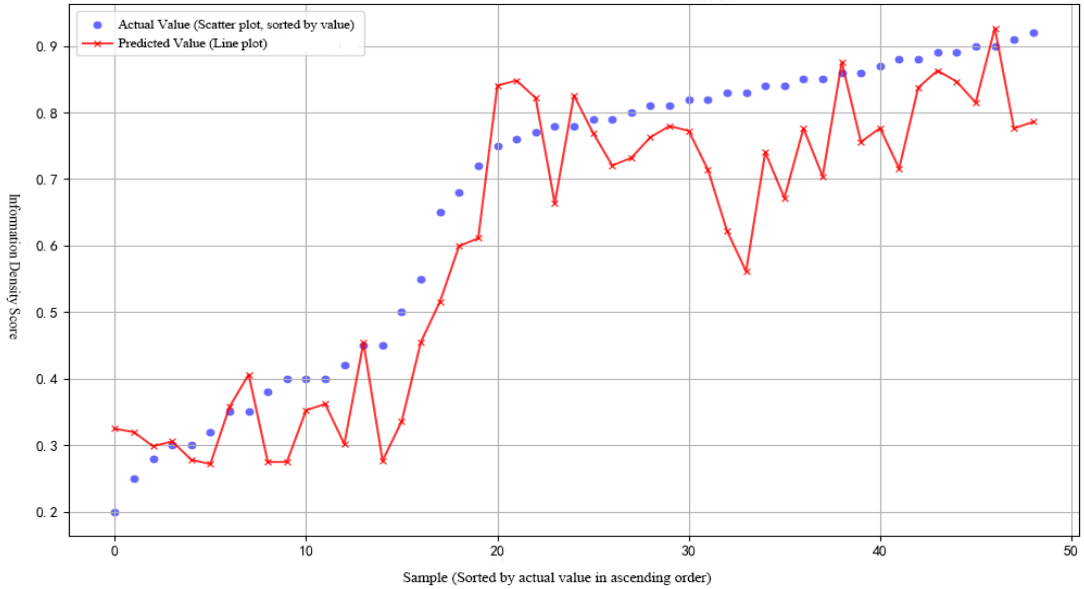

Figure 2. Scatter plot of actual values - Line chart of predicted values

4.3. Comparative Experiments

To more comprehensively evaluate the effectiveness of the model proposed in this study for the task of assessing Chinese information density, and to highlight its ability to learn complex text features, the performance of this model was compared with a series of baseline models with different design approaches.

The following baseline models were employed for comparison with the proposed model:

a. Random Baseline: This baseline model randomly assigns a predicted score uniformly distributed in the range (0,1) to each text sample in the test set. This baseline aims to provide a performance lower bound.

b. Mean Baseline: This baseline model predicts the average information density score from the training set for all test samples. This baseline represents the simplest prediction strategy based on the distribution of historical data.

c. Statistical Feature Regression: This baseline model utilizes all the statistical features employed in this study and trains a linear regression model on the same training set with the same parameters. This baseline represents a strategy for scoring the information density of Chinese texts based solely on statistical features.

d. TF-IDF Feature Regression: This baseline model uses the TF-IDF vectorizer utilized in this study to transform the text data and trains a linear regression model on the same training set using these features. This baseline represents a strategy for scoring the information density of Chinese texts based solely on TF-IDF features.

e. Lexical Entropy Normalized Score: The calculation method of this baseline model is derived from the definition of entropy in information theory, reflecting the uncertainty of vocabulary in the

vocabulary sequence in a statistical sense. First, use jieba to perform word segmentation on the vocabulary list W to obtain the total number of words M after word segmentation. N is the number of different vocabularies in M, w_i is the i-th different vocabulary, $p(w_i)$ is the probability of vocabulary w_i appearing in W, and based on this, the information density score D is obtained.

$$D = \frac{-\sum_{i=1}^{N} p(w_i) \log_2 p(w_i)}{M}$$

Subsequently, the distribution standard of the scores was determined by calculating the information density scores of the training set data, which was then used for normalization. This baseline represents a strategy for scoring information density based on the information entropy at the vocabulary level of the text.

f. BERTScore Model: BERTScore is a text generation evaluation metric based on the BERT model, which assesses similarity between the target text and the reference text in the BERT embedding space. [21] This baseline model employs the BERTScore method for evaluation and represents a strategy for scoring information density based on the semantic features extracted by BERT.

The performance of each baseline model and the main model on the test set is summarized in Table 2.

Table 2. Evaluation scores of each model on the test set

Model	$RMSE$	MAE	R^2
Random Baseline	0.2775	0.2062	-0.4892
Mean Baseline	0.2274	0.2056	0.0000
Statistical Feature Regression	0.1886	0.1758	0.3121
TF-IDF Feature Regression	0.2187	0.1943	0.0751
Lexical Entropy Normalized Score	0.3513	0.3098	-1.3874
BertScore Model	<u>0.1313</u>	<u>0.1120</u>	<u>0.6623</u>
Main Model	**0.1020**	**0.0850**	**0.7986**
IMP*	22.31%	24.11%	20.58%

*IMP represents the performance improvement of the main model relative to the second-best model.

Firstly, the performances of the random baseline and mean baseline were unsatisfactory, as their RMSE and MAE values were higher than those of all other models, except for the vocabulary entropy normalization score. The random baseline had a negative R^2, while the mean baseline was 0. Notably, the performance of the vocabulary entropy normalization score was even lower than that of the random baseline, which can be attributed to its contrary predictions for almost all classical Chinese corpus scores: due to the characteristic of single characters as words and the high frequency of commonly used function words, the tokenization may result in a relatively low variety of tokens and some tokens having extremely high frequencies, leading to lower statistical entropy values [22].

Next, TF-IDF feature regression, statistical feature regression, and the BERTScore model exhibited gradual performance improvements compared to the mean baseline. The BERTScore model, capable of capturing deep semantic information in the text, displayed superior performance among the models, aside from the main model, indicating that a significant part of the information density of Chinese texts is reflected in the semantic dimension.

Although these models captured certain aspects of text features, they were unable to fully model the complexity of information density, which is a multidimensional and deep composite concept. The model proposed in this study significantly outperformed all baseline models across all evaluation metrics. Specifically, the RMSE of this model was reduced by 22.31% compared to the BertScore model, which had the next best performance, the MAE decreased by 24.11%, and R^2 improved by 20.58%. This remarkable advantage thoroughly validates the effectiveness of the proposed model architecture.

4.4. Ablation Experiments

To verify the effectiveness of key layers in this model, a series of ablation experiments were conducted on the test set. The experimental setup was consistent with that of the main model, and performance changes were evaluated on the test set by removing or replacing certain components of the model while training with the same parameters. The following variants were specifically compared: (1) removing the BiLSTM layer and directly pooling the BERT output; (2) using only the semantic feature channel; (3) using only the statistical feature channel; (4) simply concatenating the semantic and statistical features before directly inputting them into the regression layer without using a fusion gating mechanism.

Table 3. Evaluation scores of ablation models on the test set.

Model	$RMSE$	MAE	R^2
Remove BiLSTM layer	0.1102	0.0910	0.7653
Semantic feature channel only	0.1247	0.1038	0.6995
Statistical feature channel only	0.1752	0.1603	0.4210
No fusion gating	0.1162	0.0968	0.7481
Main model	**0.1020**	**0.0850**	**0.7986**

As shown in Table 3, the results indicate that the combination of the semantic channel and statistical channel, along with the use of BiLSTM and fusion gating, all contribute positively to the model's performance, corroborating the rationality of the design of various layers in the model.

5. Conclusion and Reflection

Effective assessment of text information density is significant for various applications, such as quality evaluation of texts and locating high-density information in information retrieval systems. However, the subjectivity of its definition and the lack of measurement standards pose core challenges. This study first operationally defined information density by elucidating the multidimensional characteristics of information and established a detailed manual annotation guideline based on this foundation. Through a standardized annotation process, a small sample dataset of manually annotated Chinese information density was constructed. Based on this, a dual-channel fusion architecture model was designed and implemented to capture key features affecting information density from both semantic and statistical perspectives. A fusion gating layer effectively combines these two types of heterogeneous information, which is then output by the regression layer as a normalized information density prediction score in the range of (0,1).

This study explored the quantification and assessment of Chinese text information density, providing a feasible option, though there is still room for further optimization. Firstly, while training on a small sample allowed the model to possess a certain level of evaluation and generalization capability, constructing a larger and more diverse annotated dataset is necessary to improve model accuracy. Additionally, exploring methods such as active learning and semi-supervised learning could help reduce annotation costs and enhance model generalization ability. Secondly, the practical utility of this model in specific application scenarios, such as retrieval ranking optimization, early identification of misinformation, and difficulty classification of educational resources, requires further validation and deepening.

References

[1] Shannon C E. A mathematical theory of communication[J/OL]. The Bell System Technical Journal, 1948, 27(3): 379-423. DOI:10.1002/j.1538-7305.1948.tb01338.x.

[2] Schramm W. Information Theory and Mass Communication[J/OL]. Journalism Quarterly, 1955, 32(2): 131-146. DOI:10.1177/107769905503200201.

[3] QIAN Chen , HUANG Wei-dong. Model of Weibo Event Trend Based on Information Quantity [J/OL]. Information Science, 2019, 37(2): 46-51. DOI:10.13833/j.issn.1007-7634.2019.02.008.

[4] Wang Zheng, Wang Linsen, Zhao Lei.Research on the Microblog Bursty Topic Detection Model Based on Information Density [J/OL]. Information Studies:Theory & Application, 2016, 39(3): 125-129. DOI:10.16353/j.cnki.1000-7490.2016.03.025.

[5] Crocker M W, Demberg V, Teich E. Information Density and Linguistic Encoding (IDeaL)[J/OL]. KI - Künstliche Intelligenz, 2016, 30(1): 77-81. DOI:10.1007/s13218-015-0391-y.

[6] Liu X, Li F, Xiao W. Measuring linguistic complexity in Chinese: An information-theoretic approach[J/OL]. Humanities and Social Sciences Communications, 2024, 11(1): 1-12. DOI:10.1057/s41599-024-03510-7.

[7] Mekheimer A M, Fageeh I A. Prioritizing information over grammar: a behavioral investigation of information density and rhetorical discourse effects on EFL listening comprehension[J]. Discover Education,2025,4(1):24-24.

[8] Sweller J. CHAPTER TWO - Cognitive Load Theory[M/OL]//Mestre J P, Ross B H. Psychology of Learning and Motivation: 55. Academic Press, 2011: 37-76[2025-05-08]. https://www.sciencedirect.com/science/article/pii/B9780123876911000028. DOI:10.1016/B978-0-12-387691-1.00002-8.

[9] Miller G A, Beckwith R, Fellbaum C, et al. Introduction to WordNet: An On-line Lexical Database*[J/OL]. International Journal of Lexicography, 1990, 3(4): 235-244. DOI:10.1093/ijl/3.4.235.

[10] Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis[J/OL]. Journal of the American Society for Information Science, 1990, 41(6): 391-407. DOI:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[11] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[A/OL]. arXiv, 2013[2025-05-10]. http://arxiv.org/abs/1301.3781. DOI:10.48550/arXiv.1301.3781.

[12] Pennington J, Socher R, Manning C. GloVe: Global Vectors for Word Representation[C/OL]//Moschitti A, Pang B, Daelemans W. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1532-1543[2025-05-10]. https://aclanthology.org/D14-1162/. DOI:10.3115/v1/D14-1162.

[13] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[A/OL]. arXiv, 2018[2025-05-10]. http://arxiv.org/abs/1802.05365. DOI:10.48550/arXiv.1802.05365.

[14] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C/OL]//Burstein J, Doran C, Solorio T. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171-4186[2025-05-10]. https://aclanthology.org/N19-1423/. DOI:10.18653/v1/N19-1423.

[15] SPARCK JONES K. A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL[J/OL]. Journal of Documentation, 1972, 28(1): 11-21. DOI:10.1108/eb026526.

[16] Luhn H P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information[J/OL]. IBM Journal of Research and Development, 1957, 1(4): 309-317. DOI:10.1147/rd.14.0309.

[17] Santorini B. Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision)[C/OL]. 1990[2025-05-11]. https://www.semanticscholar.org/paper/Part-of-Speech-Tagging-Guidelines-for-the-Penn-(3rd-Santorini/a145854ede2f62098bf4e92de1584ab270b676c9.

[18] Marcus M, Kim G, Marcinkiewicz M A, et al. The Penn Treebank: annotating predicate argument structure[C/OL]//Proceedings of the workshop on Human Language Technology. USA: Association for Computational Linguistics, 1994: 114-119[2025-05-10]. https://dl.acm.org/doi/10.3115/1075812.1075835. DOI:10.3115/1075812.1075835.

[19] Nivre J, de Marneffe M C, Ginter F, et al. Universal Dependencies v1: A Multilingual Treebank Collection[C/OL]//Calzolari N, Choukri K, Declerck T, et al. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC`16). Portorož, Slovenia: European Language Resources Association (ELRA), 2016: 1659-1666[2025-05-11]. https://aclanthology.org/L16-1262/.

[20] Zhang S, Wang L, Sun K, et al. A Practical Chinese Dependency Parser Based on A Large-scale Dataset[A/OL]. arXiv, 2020[2025-05-11]. http://arxiv.org/abs/2009.00901. DOI:10.48550/arXiv.2009.00901.

[21] Zhang T, Kishore V, Wu F, et al. BERTScore: Evaluating Text Generation with BERT[A/OL]. arXiv, 2020[2025-05-20]. http://arxiv.org/abs/1904.09675. DOI:10.48550/arXiv.1904.09675.

[22] Horne B D, Adali S. This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News[A/OL]. arXiv, 2017[2025-05-01]. http://arxiv.org/abs/1703.09398. DOI:10.48550/arXiv.1703.09398.