International Journal of Geo-Information

MDPI

*Article*

# Extracting Geoscientific Dataset Names from the Literature Based on the Hierarchical Temporal Memory Model

**Kai Wu [1,2], Zugang Chen [1,*], Xinqian Wu [2], Guoqing Li [1], Jing Li [1], Shaohua Wang [1], Haodong Wang [3] and Hang Feng [3]**

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; wk_wk@stu.haust.edu.cn (K.W.); ligq@aircas.ac.cn (G.L.); lijing6@aircas.ac.cn (J.L.); wangshaohua@aircas.ac.cn (S.W.)
[2] School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang 471023, China; wxq1001@haust.edu.cn
[3] School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China; zzuwhd@gs.zzu.edu.cn (H.W.); feng_hang_fh@gs.zzu.edu.cn (H.F.)
* Correspondence: chenzg@aircas.ac.cn

**Abstract:** Extracting geoscientific dataset names from the literature is crucial for building a literature–data association network, which can help readers access the data quickly through the Internet. However, the existing named-entity extraction methods have low accuracy in extracting geoscientific dataset names from unstructured text because geoscientific dataset names are a complex combination of multiple elements, such as geospatial coverage, temporal coverage, scale or resolution, theme content, and version. This paper proposes a new method based on the hierarchical temporal memory (HTM) model, a brain-inspired neural network with superior performance in high-level cognitive tasks, to accurately extract geoscientific dataset names from unstructured text. First, a word-encoding method based on the Unicode values of characters for the HTM model was proposed. Then, over 12,000 dataset names were collected from geoscience data-sharing websites and encoded into binary vectors to train the HTM model. We conceived a new classifier scheme for the HTM model that decodes the predictive vector for the encoder of the next word so that the similarity of the encoders of the predictive next word and the real next word can be computed. If the similarity is greater than a specified threshold, the real next word can be regarded as part of the name, and a successive word set forms the full geoscientific dataset name. We used the trained HTM model to extract geoscientific dataset names from 100 papers. Our method achieved an F1-score of 0.727, outperforming the GPT-4- and Claude-3-based few-shot learning (FSL) method, with F1-scores of 0.698 and 0.72, respectively.

**Keywords:** geoscientific dataset; named-entity recognition; hierarchical temporal memory; word encoding



**Citation:** Wu, K.; Chen, Z.; Wu, X.; Li, G.; Li, J.; Wang, S.; Wang, H.; Feng, H. Extracting Geoscientific Dataset Names from the Literature Based on the Hierarchical Temporal Memory Model. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 260. https://doi.org/10.3390/ijgi13070260

Academic Editors: Wolfgang Kainz and Dev Raj Paudyal

Received: 8 April 2024
Revised: 18 July 2024
Accepted: 19 July 2024
Published: 21 July 2024



## 1. Introduction

Geoscientific datasets include data describing the state, properties, and distribution characteristics of phenomena or entities in specific layers or geographic locations on Earth [1–3]. These datasets are widely used in atmospheric, oceanic, geological, terrestrial surface, and solar–terrestrial space science research [4]. The scientific and technical literature [5] is essential for the acquisition of knowledge by researchers as an important source of recorded scientific discoveries and innovations. With the rise and rapid development of open science [6], an increasing number of geoscientific datasets and papers are being published and shared on the Internet. In the field of geoscience, scientific research is increasingly dependent on geoscientific datasets in the context of a data-intensive scientific research paradigm [7]; thus, the literature contains extensive geoscientific dataset names.

However, in most cases, geoscientific datasets and papers are scattered in different corners of the Internet, and correlations between them have not been established.

Extracting the names of geoscientific datasets from the literature can clarify which datasets are used in research. It can further help us to discover these datasets through the Internet and establish correlations between studies and datasets. Consequently, this can help readers quickly access research data when they read the literature online and rapidly reproduce studies [8,9]. Furthermore, extracting geoscientific dataset names can assist data-publishing journals, such as *Earth System Science Data*, in tracking the citations of their published datasets [10,11].

The name of a geoscientific dataset generally contains elements such as geospatial coverage, temporal coverage, scale or resolution, theme content, and version [12,13]. For example, "San Francisco 5-m resolution land use data v1.0 for the years 2015–2020" represents a typical name for a geoscientific dataset. In this instance, "San Francisco" denotes the geospatial coverage of the dataset, "5-m resolution" indicates the spatial resolution of the dataset, "land use data" describes the content of the dataset, "v1.0" signifies the dataset's version, and "years 2015–2020" indicates the temporal coverage of the dataset. Extracting dataset names from the literature is also the task of domain-specific named-entity recognition methods [14,15]. Although existing methods have made some progress in named-entity recognition (NER), they still face many challenges in extracting geoscientific dataset names. Because the names of geoscientific datasets contain many elements and there can be many combinations of elements, it is hard for existing methods to understand the rules in the names. Therefore, existing methods often can only extract some elements of the name, not the complete name, which results in low extraction accuracy.

To address these problems, this study proposes a method for extracting geoscientific dataset names based on the hierarchical temporal memory (HTM) model. The HTM model is a brain-inspired artificial neural network [16,17] that has excellent abilities in high-level cognitive activities, such as language understanding, spatial cognition, and navigation [18]. The elements of dataset names are encoded into binary vectors by our proposed new encoding method. The sequence of binary vectors of a dataset name is input to train the HTM model and make it understand the naming rule. We then use the trained HTM model to recognize the names of geoscientific datasets from text in the literature. Compared with existing methods such as GPT-4 and Claude-3, the results show that our method has a higher F1-score.

The main contributions of the paper are as follows:

(1) An artificial neural network method developed specifically for extracting the names of geoscientific datasets is proposed. Compared with the GPT-4-based few-shot learning (FSL) method, this method has a higher F1-score.
(2) A new word-encoding method for the HTM model is proposed. This method uses the Unicode values of characters to determine their relative positions in the semantic vector and employs a numerical mapping approach to reduce the encoding length. Compared to the semantic folding method, this approach demonstrates higher accuracy in encoding Chinese characters.
(3) A new decoding structure for the HTM model is proposed. This decoding structure uses a BP neural network to decode the prediction vector for the encoding of the next word. By calculating the similarity of the encoding of the predicted word and the actual next word, the name of the geoscientific dataset is extracted word by word.

The remainder of the paper is organized as follows. Section 2 describes the state of the art. Section 3 presents the general idea and methodology. Section 4 mainly describes the experimental design for extracting geoscientific dataset names using the proposed method and two large language models—GPT-4 and Claude-3. Section 5 evaluates and compares our method with existing methods. The last section concludes the research and discusses the future research direction.

## 2. Related Work

We propose a geoscientific dataset name-extraction method based on the HTM model. Therefore, the reviewed literature includes geoscientific dataset name-extraction methods and HTM-based algorithms.

### *2.1. Progress in Geoscientific Dataset Name Extraction*

Different from traditional personal or place-name extraction, most existing named-entity recognition (NER) methods are not suitable for geoscientific datasets because these datasets have complex and compound elements. Even so, some researchers have proposed a few geoscientific dataset name or related name-entity extraction methods. These methods can be classified into three categories.

The first category is rule-based methods. A rule-based method requires the construction of a rule database. For example, Cao et al. [19] used regular expressions to create dataset name-extraction rules and successfully extracted dataset names from the geoscience literature with 62% accuracy. Afzal et al. [20] proposed a rule-based citation mining technique. This technique detected data such as author, title, and conference location from documents and extracted the relevant titles from the computer science literature database of the Digital Bibliography Library Project (DBLP). Rule-based methods require the manual creation of explicit rules and are time-consuming [21,22]. Moreover, for complex named entities, it is difficult to establish a complete rule database, so the accuracy is not high.

The second category is traditional machine learning methods. Machine learning methods generally use the inside, outside, beginning (IOB) annotation system to manually annotate the corpus and then use machine learning models for training and prediction [23]. Commonly used machine learning methods for named-entity recognition include the hidden Markov model (HMM) [24], support vector machine (SVM) [25], and conditional random fields (CRF) [26]. Han et al. [27] proposed an SVM-based method for extracting metadata from a structured corpus that outperformed other machine learning methods. Although HMM, SVM, and CRF can theoretically be used for geoscientific data name extraction, in reality, to the best of our knowledge, no studies have used these methods to directly extract geoscientific data names.

The third category is deep learning-based methods. In recent years, deep learning methods [28] such as the large language model (LLM) have achieved remarkable results in the field of natural language processing. Deep learning-based methods can automatically learn features from data and effectively process large text data. Commonly used deep learning methods for named-entity recognition include convolutional neural network (CNN) [29], bidirectional long short-term memory (Bi-LSTM) [30], transformer models [31], and GPT-4 [32]. Yao et al. [33] extracted dataset names and methods based on the Bi-LSTM model from the papers of PAKDD conferences (2009–2019). Kumar et al. [34] tested dataset name extraction based on the bidirectional encoder representations from transformers (BERT) model on research papers in a popular social science corpus and achieved an F1-score of 56.2%. Younes and Scherp [35] proposed one-step and two-step methods to extract unknown dataset names from scientific papers. The one-step method had higher accuracy, while the two-step method could extract more potential datasets. Heddes et al. [15] annotated 6000 sentences from AI conferences and used the SciBERT method, a pretrained LLM for scientific text [36], to automatically detect dataset names from scientific articles. Geogalactica [37], which was trained on a geoscience-related text corpus, also demonstrated advanced performance in various NLP tasks, including the extraction of key information in geoscience.

From the research above, it can be found that little research has been conducted on directly extracting geoscientific dataset names from unstructured text, such as the geoscientific literature. The existing methods either have low accuracy or are not specifically designed for geoscientific datasets. LLMs [37,38], which are reported to have impressive performance in NLP tasks including NER, may not perform as well in the extraction of geoscientific datasets, as geoscientific datasets involve a lot of specialized knowledge and have complex combination patterns. Thus, high-precision methods for extracting geoscientific data names are urgently

needed to associate the data with scientific studies, to measure the usage of scientific data in the scientific literature, and to reproduce and reuse the research results of the studies.

### 2.2. Progress in HTM Application

Time is an important aspect of vocabulary embedding and neural network training. Time not only influences the semantics of text but also reveals the dynamic changes in text. Therefore, many time-aware text embedding methods have been proposed, such as time-aware text embedding approach to generate subgraphs (TEAGS) [39], short-text author linking through multi-aspect temporal-textual embedding (SoulMate) [40], the multi-aspect time-related influence (MATI) model [41], which integrates multiple time features and combines the temporal subset property (TSP) [42], and the value-wise ConvNet for transformer models [43]. These methods highlight the importance of the temporal dimension in text embedding and demonstrate significant advantages in time-sensitive recommendation tasks.

Realizing the importance of the temporal dimension in data processing, George and Hawkins [16,17] proposed the hierarchical temporal memory (HTM) model, which was designed to simulate the structure and function of the human brain and had excellent performance in time series data processing. Compared to the traditional neurons of an artificial neural network, HTM has a great number of dendritic connections, and the output of HTM neurons depends not only on feedforward inputs (proximal connections to the soma) but also on lateral connections between the neurons. Thus, it can perform high-level cognitive activities, such as language analysis, spatial cognition, etc. It is widely applied in anomaly detection [44,45], image processing [46], natural language processing [47], and other areas. For instance, Afaf et al. [48] applied the HTM algorithm to traffic congestion detection, demonstrating a 7.4% improvement in detection accuracy compared to state-of-the-art techniques, with an average F-score of 98.83%. Szoplák et al. [49] tested the HTM algorithm on the PAN plagiarism corpus, achieving 70.15% accuracy in detecting anomalous texts. Hamid et al. [50] proposed a text anomaly detection framework based on the improved semantic folding theory (SFT), achieving 96% accuracy on the Yelp dataset. HTM has unique advantages in handling continuous streaming data and capturing temporal patterns, although it may require more training when adapting to new data [51,52]. The initial version of HTM has now been upgraded to the third generation [53,54].

In our research, we relied on the excellent ability of HTM to process time-series data to realize the extraction of geoscientific dataset names from text. This is a newly proposed method that encodes words to binary vectors and uses a large amount of geoscientific dataset names for training the HTM model, which allows the model to master the naming rules of geoscientific datasets and the combination rules of different name elements. Then, we used the trained HTM model to extract geoscientific dataset names from the literature. Finally, we compared our method with existing methods in relation to extraction accuracy.

## 3. Methodology

### 3.1. General Idea

The input of the HTM model is binary vectors. In order to deal with the name text of geoscientific datasets, we propose a new word-encoding method to convert words to binary vectors. Then, using the HTM model, we can extract geoscientific dataset names based on the word-encoding method. We collected geoscientific dataset names from geoscientific data-sharing websites as a corpus or sample to train the created HTM model. We segmented the geoscientific dataset names into a sequence of words by using the NLPIR tool [55]. Each word represented a feature of the geoscientific dataset and was encoded and input into the HTM model, enabling it to learn the naming rules and combination patterns of various features. A trained HTM neural network was obtained by the offline section. Then, we used the trained HTM model to extract geoscientific names by the online section as follows: For arbitrary unstructured text from geoscientific papers, we segmented a sentence from the text into word sequences using the NLPIR tool [55]. The word was encoded and input to the trained HTM model, and the HTM model generated a predictive vector. In this

research, we decoded the predictive vector for the next word's encoding vector by using a BP neural network as the classifier for the HTM model so that we could use the similarity of encoding of the predictive next word and real next word to determine whether the real word was part of a geoscientific dataset name. Finally, the combination of consecutive multiple words was regarded as a geoscientific dataset name, thereby achieving the goal of automatically extracting a geoscientific dataset name from unstructured text.

Finally, we evaluated our method by comparing it with zero-shot learning (ZSL) and few-shot learning (FSL) methods, which were based on GPT-4 and Claude-3, and we evaluated the precision and recall of the proposed method in the task of extracting geoscientific dataset names. The general idea is shown in Figure 1.

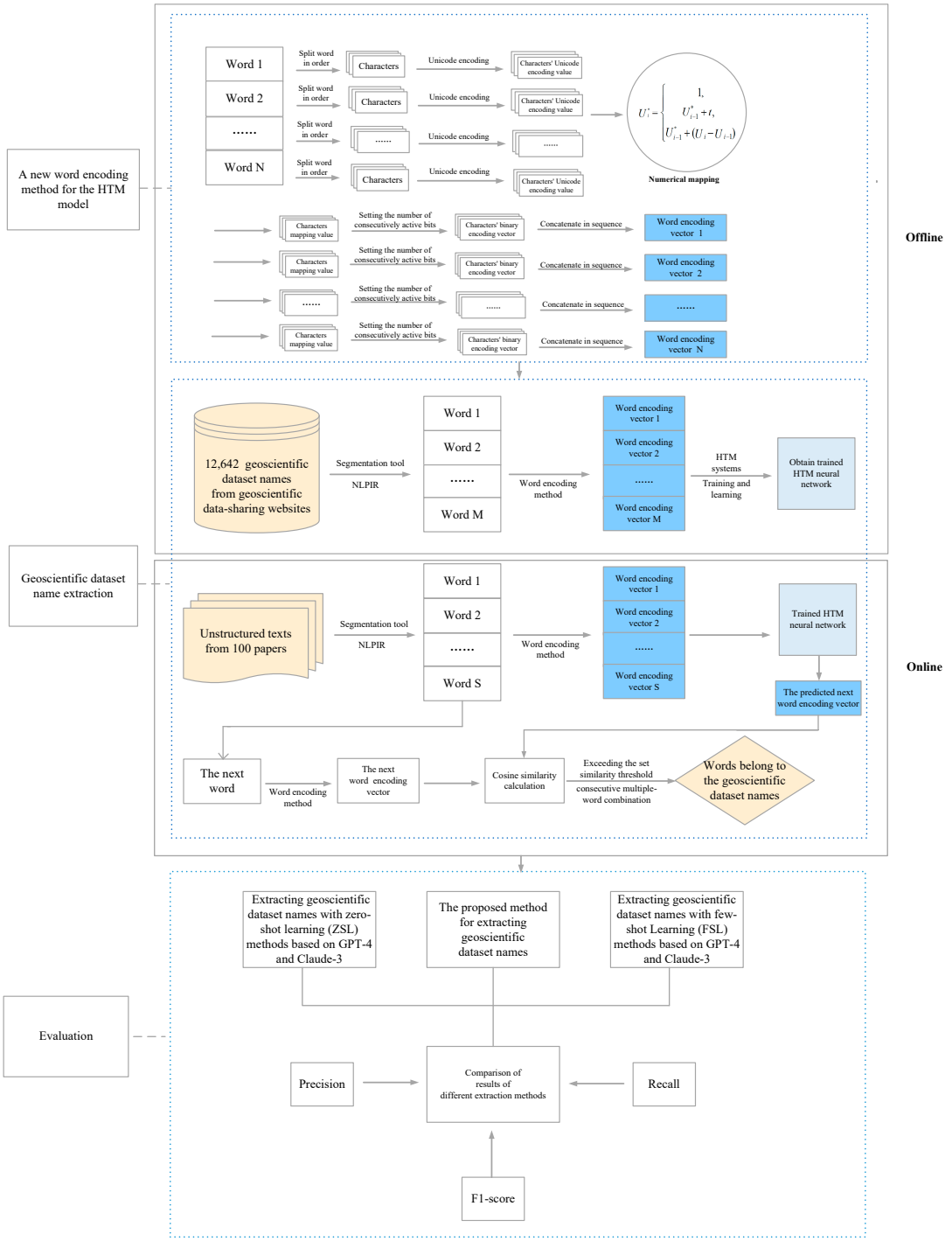


**Figure 1.** General idea of the paper.

### *3.2. HTM Model*

Hierarchical temporal memory (HTM) is a biomimetic machine intelligence algorithm [52] that was first proposed by George and Hawkins [16,17]. HTM is inspired by the structure of the cerebral cortex of the human brain and has more fidelity than traditional artificial neural networks [56].

A primary feature of HTM is its hierarchical structure; it is composed of several layers with distinct regions (Figure 2A). The spatial pooler (SP) and the temporal memory (TM) are core components of HTM (Figure 2B). The SP converts all input patterns into sparse distributed representations (SDRs), which are used as the input of the TM. The TM learns the temporal sequences of the SDRs and can predict the next input of the HTM. HTM neurons (Figure 2C) are similar to biological neurons, with thousands of synapses on activated dendrites, and the model learns by simulating the growth of new synapses and the decay of inactive synapses. The proximal and distal dendritic segments of the HTM neurons have different functions [57]. Patterns detected on the proximal dendrites lead to action potentials (i.e., they are activated) (Figure 2D) and define the classic receptive field of neurons. Patterns recognized by the distal synapses of neurons act as predictors by depolarizing the cell without directly causing an action potential [58]. The output of HTM neurons depends not only on feedforward inputs (Figure 2E) but also on lateral connections between neurons. The structure of HTM is shown in Figure 2.

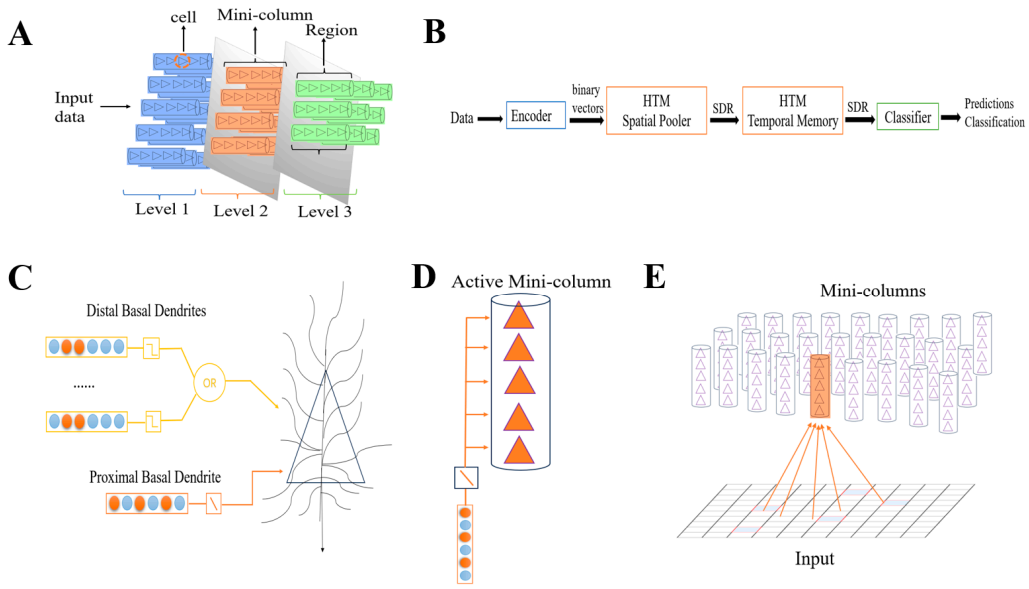


**Figure 2.** HTM structure [57,58]. (**A**) HTM has a three-level hierarchy. The smallest unit is an HTM cell. In each layer, there are a large number of cells, multiple cells form mini-columns, and multiple mini-columns form regions. (**B**) The end-to-end HTM system includes an encoder, HTM SP, HTM TM, and a classifier. (**C**) An HTM neuron has one proximal dendrite and several distal dendrites, and dendrites have different functions. Proximal dendrites receive feedforward inputs, while distal dendrites receive contextual information from nearby cells in the layer. (**D**) All cells in the same mini-column share the same synapses that receive feedforward inputs, which means they receive the same information. (**E**) Each layer of the HTM model consists of several mini-columns of cells that can read and form synaptic connections with input data.

#### 3.2.1. Sparse Distributed Representations

A sparse distributed representation (SDR) is a sparse, high-dimensional binary vector consisting of a large number of zeros and ones [59]. At every point in time, only a few encoding sites are activated and have a value of 1, which corresponds to active neurons in the mammalian brain, while the rest of the coding sites have a value of 0. Although a single

encoding site does not represent valid information, the overall combination of encoding sites has a certain semantic meaning.

#### 3.2.2. A New Word-Encoding Method for the HTM Model

The HTM encoder encodes the input data into a binary vector that can be processed by the SP layer. The purpose of the encoding process is to determine which output bits are zeros and which are ones, to capture the semantic features of the data. The encoder should conform to four principles: the encoder of input data has the same dimension, the same data are encoded identically, similar data are encoded similarly, and the output sparsity of different data remains essentially the same [60].

A statistical analysis of the Unicode distribution of Chinese characters shows that the Unicode values of semantically similar characters are arranged in nearby locations [61,62]. This study proposes a new word-encoding method for HTM based on the Unicode and arranging law of characters. First, the Unicode values of the 3596 commonly used Chinese characters are obtained; the minimum Unicode value is 40, and the maximum is 65,311. To decrease the dimension of the encoder, the Unicode values of the 3596 characters are mapped to new values with a more compact distribution. The following method is used:

$$U_i^* = \begin{cases} 1, & U_1 \\ U_{i-1}^* + t, & |U_i - U_{i-1}| > t, \\ U_{i-1}^* + (U_i - U_{i-1}), & 0 < |U_i - U_{i-1}| \le t \end{cases} \quad i \text{ is a positive integer.} \tag{1}$$

where $U_1$, $U_2$, …, $U_{i-1}$, $U_i$, …, $U_n$ represents the Unicode values for the 3596 commonly used Chinese characters, letters, symbols, and numerals, sorted in ascending order. $U_i^*$ is the conversion value, and $t$ is a threshold that controls the total number of encoding bits and the degree of semantic generalization.

Secondly, we use the NLPIR tool to segment the names of geoscientific datasets into word sets (the maximum number of characters in the word is set) [55]. For every word, the Unicode value of each character contained in it is obtained and converted by using Formula (1). The following steps are used to encode words: (1) For all $U_i^*$, the minimum $U_i^*$ is $U_1^*$, the maximum is denoted as $U_{\max}^*$, and the range is $[U_1^*, U_{\max}^*]$. (2) We create $U_{\max}^* - U_1^*$ buckets to split the character encoder values. (3) We choose the number of consecutive active bits $w$ in each representation and compute the total number of bits $n$ of the encoded character: $n = U_{\max}^* + w - 1$. For each character and its $U_i^*$, the encoded representation is created by setting $n$ unset bits with $w$ consecutive active bits starting at index $U_i^*$. (4) All encodings of characters in the word are combined in order. Words that do not reach the maximum length are encoded as empty with zeros to ensure that all words have the same number of bits in their encoding. An example of encoding the names of geoscientific datasets is shown in Figure 3.

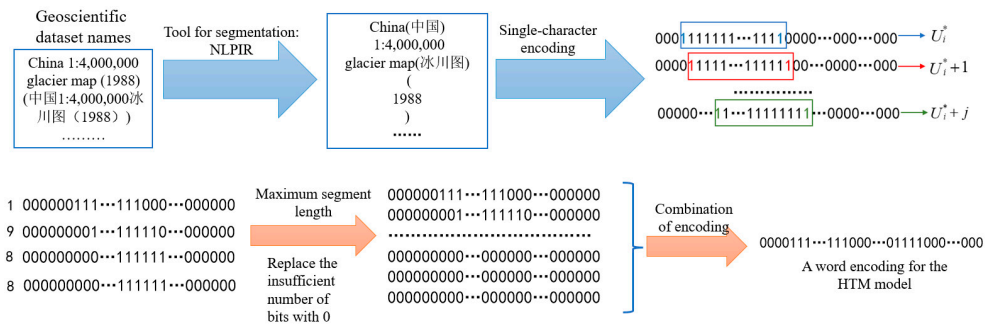


**Figure 3.** Example of encoding words in names of a geoscientific dataset using our method.

### 3.2.3. HTM Spatial Pooler

The main purposes of the spatial pooler are, first, to transform input vectors into sparse distributed representations, and second, to learn to better detect repeated input patterns. The spatial pooler algorithm is shown in Figure 4.

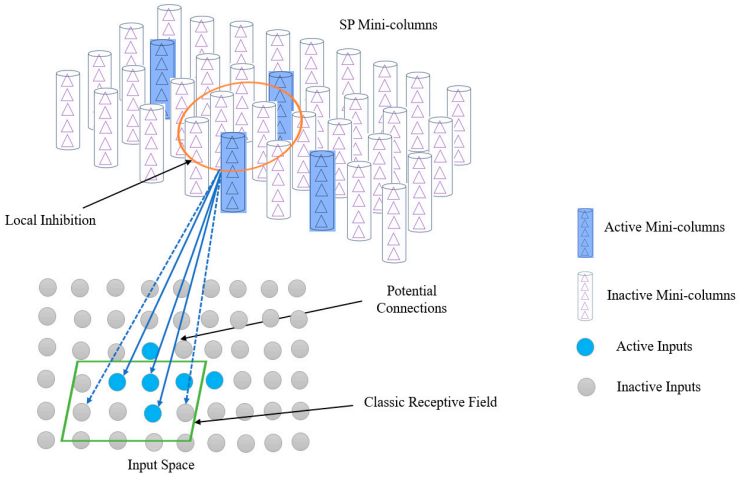


**Figure 4.** Structure of HTM spatial pooler [58].

The spatial pooler (SP) algorithm consists of three steps: the formation of classic receptive field (for example, the dashed green rectangle in Figure 4), the activation of mini-columns, and the learning of proximal synapse permanence. The classic receptive field formation stage determines the key parameters of the HTM model and establishes potential connections between the input data and the synapses of the proximal dendrites of mini-column $k$. The potential connection between mini-column $k$ and the input data can be denoted as

$$P_k = \{i | I(x_i; x_k^c, \lambda) \; and \; (\alpha_{ki} \sim U(0,1))\}, \tag{2}$$

where $x_i$ denotes the $i$-th input neuron (data), $x_k^c$ is the central neuron of the classic receptive field of mini-column $k$, and $\lambda$ is the length of the classic receptive field. $\alpha_{ki}$ is a random number chosen from a uniform distribution $U(0,1)$ and denotes the synaptic permanence between the $k$-th mini-column and input neuron $x_i$. For potential connections to become synaptic connections, the permanence of the potential synapses must exceed a given threshold $\eta_c$. $I(x_i; x_k^c, \lambda)$ is an indicator function: if and only if $x_i$ is located in the classic receptive field of mini-column $k$, then the input neuron $x_i$ is selected as the potential input of the mini-column.

Given an input pattern $x$, the activation of the SP mini-columns is determined based on the computation of the feedforward input, which is called overlap value $O_k$:

$$O_k = \beta_k \sum_i B_{ki} x_i, \tag{3}$$

$$B_{ki} = \begin{cases} 1, & \alpha_{ki} > \eta_c \\ 0, & otherwise' \end{cases} \tag{4}$$

where $\beta_k$ is a positive boost factor, which controls the excitability of the SP mini-columns and is adjusted for learning during the SP training process, and $B_{ki}$ is a binary indicator matrix that represents connected synapses, with $B_{ki} = 1$ when $\alpha_{ki}$ is greater than $\eta_c$; otherwise, $B_{ki} = 0$. Suppose the connection threshold for synaptic permanence $\eta_c$ (values in [0, 1]) is set to be 0.7, such that initially 70% of the potential synapses are connected. The value of $\eta_c$ determines the threshold for whether a synapse is considered to be connected to a neuron. If the permanence value for a synapse is greater than $\eta_c$, it is considered to be connected. According to the local inhibition mechanism, the $k$-th SP mini-column needs to

satisfy two conditions to be activated: first, the overlap value $O_k$ of the $k$-th mini-column must be at the top $n$ $(n < m)$ of its $m$ neighboring mini-columns, and second, its input overlap needs to be greater than the given activation threshold $\theta$. To ensure that a certain number of mini-columns are activated, activation threshold $\theta$ is usually set to a small positive number. Let $C_k$ denote the activation state of the $k$-th SP mini-column; that is:

$$O_{Nk} = \{O_k | k \in N_k\}, \tag{5}$$

$$C_k = \begin{cases} 1, & O_k \geq prctile(O_{Nk}, 1-d) \; and \; O_k \geq \theta \\ 0, & otherwise \end{cases}. \tag{6}$$

where $N_k$ denotes the set of neighboring mini-columns and $O_{Nk}$ denotes the overlap value of all neighboring mini-columns of the $k$-th SP mini-column; $prctile(\cdot)$ denotes the percentile function; and $d$ denotes the activation density of the target mini-column.

In the proximal synaptic permanence learning stage, the synaptic permanence values of the activation mini-columns are updated based on Hebb's learning rule. This rule means that synapse permanence is strengthened for active input connectivity bits and weakened for inactive input connectivity bits, and the synapse permanence value is limited to between zero and one.

#### 3.2.4. HTM Temporal Memory

Temporal memory [63,64] is a predictive mechanism based on time sequence data that uses synaptic connections between cells to construct temporal associations of things with different features. Temporal memory has two main purposes: the first is to transform the SDRs of the output of the SP process into a representation that captures the temporal background of the current input, and the second is to predict future inputs based on previous sequences.

The temporal memory (TM) algorithm consists of three parts (Figure 5): determining the activated cells in the mini-column, learning the permanence of distal synapses, and obtaining the predicted cells. First, after the SP process, partially activated mini-columns are generated, and the TM process will activate a few cells in the active mini-columns based on historical information. The status of activated cells is calculated as follows:

$$\beta_{kl}^t = \begin{cases} 1, & if \; k \in c\_cols^t \; and \; \mathbb{C}_{kl}^{t-1} = 1 \\ 1, & if \; k \in c\_cols^t \; and \; \sum_l \mathbb{C}_{kl}^{t-1} = 0 \\ 0, & otherwise \end{cases}, \tag{7}$$

where $c\_cols^t$ denotes the mini-column activated at moment $t$, $\mathbb{C}_{kl}^{t-1}$ denotes the predicted state of the $l$-th cell of the $k$-th mini-column at moment $t-1$, and $\beta_{kl}^t$ denotes the activation state of the $l$-th cell of the $k$-th mini-column at moment $t$. If a cell in the activation column at a previous moment is in a predicted state, then that cell will be activated. If no cell in the activation column is in a predicted state, then all cells in the mini-column will be activated. The TM algorithm differs from the SP algorithm in that it learns sequence correlations by activating cells in the active mini-column and represents different contexts.

Like the SP process, the TM process adjusts the permanence of distal synapses using Hebb's learning rule. Dendritic segments on learning cells connected to active cells from the previous period are reinforced, while dendritic segments on learning cells connected to inactive cells are punished. The reinforcement of dendritic segments increases the permanence value of active synapses by a larger-value $\tau^+$ while decreasing the permanence value of inactive synapses by a smaller-value $\tau^-$.

$$\Delta S_{kl}^d = \tau^+ \hat{S}_{kl}^d \circ A^{t-1} - \tau^- \hat{S}_{kl}^d \circ (1 - A^{t-1}), \tag{8}$$

$$\hat{S}_{kl}^d = \begin{cases} 1, & if \; S_{kl}^d > 0 \\ 0, & otherwise \end{cases}, \tag{9}$$

Suppose that the number of mini-columns in an HTM neural network is $N$ and each column has $M$ cells. $S_{kl}^{d}$ is an $M \times N$ matrix denoting the persistence of the dendritic segment of the $l$-th cell of the $k$-th mini-column, $A^{t}$ is an activated status matrix of $M \times N$ at moment $t$, and $\beta_{kl}^{t}$ denotes the activation state of the $l$-th cell of the $k$-th mini-column. $\circ$ denotes the multiplication of matrix elements at corresponding positions, and $\hat{S}_{kl}^{d}$ denotes the matrix containing the positive terms in $S_{kl}^{d}$.

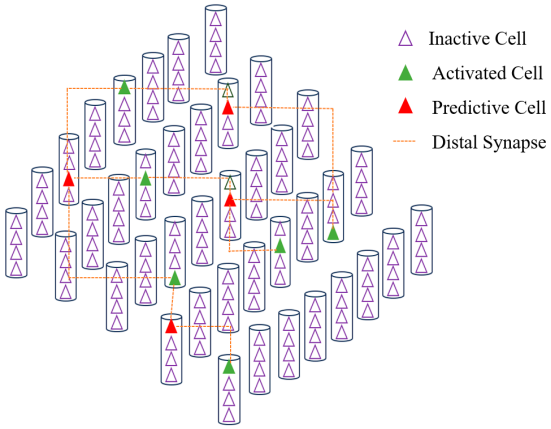


**Figure 5.** Structure of HTM temporal memory [63,64]. Cells in the TM process can exist in three states: inactive, active, or predictive. When a cell does not receive any feedforward input, it is in an inactive state (purple triangle), and when it receives feedforward input, it is in an active state (green triangle). Sufficient lateral activity in a contextual dendrite leads to a predictive state (red triangle).

Finally, the TM process will make predictions for the next time of input based on the number of activated cells. Whether a distal dendritic segment is active or not depends on the number of active synapses connected to active cells, and if this number exceeds a threshold $\zeta$, the dendritic segment is activated. The dendritic segment activation threshold $\zeta$ indicates how many cells need to be activated within a time step to activate the entire mini-column, and its value is correlated with the number of cells contained in the mini-column. The value of $\zeta$ cannot exceed the number of cells contained in the mini-column. Therefore, the predicted state of the cell at moment $t$ is:

$$\mathbb{C}_{kl}^{t} = \begin{cases} 1, & if\ \exists_d \| \ DS_{kl}^{d} \circ A^{t} \ \|_1 > \zeta \\ 0, & otherwise \end{cases}. \tag{10}$$

where $DS_{kl}^{d}$ is the $d$-th distal dendritic segment of the $l$-th cell of the $k$-th cell column, and $A^{t}$ is a 0–1 matrix with size $M \times N$.

#### 3.2.5. Classifier

We can use common classifier algorithms such as support vector machine and naive Bayes for HTM models [65]. In this paper, text data are encoded and input into the HTM model. For the SDRs generated by the SP or TM process, the classifier can transform them into human-readable data, which is similar to the decoding process. In this paper, the BP algorithm [66,67] is used as a classifier, i.e., a decoder.

In contrast to the traditional HTM classifier, which directly transfers the prediction encoder of TM to data humans can understand, in this research, we use BP to transfer the prediction encoder of TM to the next word's encoder in the training stage. In the application stage of the trained HTM, when we encode the former word and input the encoder to the HTM, the next word's encoder is predicted. The word vector similarity between the encoders of the predictive word and real next word is computed. If the similarity is greater than a set threshold, the concatenation of the two words is deemed to conform to the naming pattern of a geoscientific dataset.

## 4. Experimental Section

In this section, we create a corpus (sample) and use the word-encoding method to train the created HTM model, enabling the HTM model to learn the naming rules and understand the combination patterns of different elements of the geoscientific dataset.

### *4.1. Training Corpus*

To make the HTM model learn the naming rules and understand the combination patterns of different elements of the geoscientific dataset, we collected various geoscientific dataset names as a corpus or sample for training. The National Earth Observation Data Center (NODA) is the biggest Earth observation data center in China. About 8000 Earth observation datasets are shared in its data platform, ChinaGEOSS (https://www.chinageoss.cn/datasharing (accessed on 12 January 2024)). The National Earth System Science Data Center (NESSDC) is a key repository for Earth science data, covering a variety of data including atmospheric, oceanic, terrestrial, and ecological sciences. It aims to support Earth system research by providing a centralized platform for collecting, archiving, and disseminating data. NESSDC has shared more than 30,000 geoscientific datasets (https://www.geodata.cn/ (accessed on 12 January 2024)). We randomly selected about one-third (12,642) of the dataset names from the two national platforms to make sure that the sample has wide coverage. The NLPIR [55] was used to segment the 12,642 dataset names into words. The number of characters in a word was limited to 6. Starting and ending labels were added to all segmented names.

### *4.2. Creating and Training the HTM Model*

We conducted experiments using the open-source HTM framework on the Numenta Platform for Intelligent Computing (NuPIC) (https://github.com/numenta/htm.java (accessed on 15 January 2024)). First, we created an HTM model. Then, we encoded all 12,642 geoscientific dataset names using the method described in Section 3.2.2, and the encodings of every word in every dataset name were input to the HTM in sequence. We trained the model on a computer with two 2.1 GHz Intel processors, 512 GB memory, 8 NVIDIA GeForce RTX 3080, and Windows 10 Professional. To obtain an HTM model with optimal performance, we tested the relationship between model size, training iteration times, and prediction accuracy. We set the number of columns of the HTM to 1000, 2000, 3000, 4000, 5000, and 6000, and set the cycles of the HTM to 50, 100, 150, 200, 250, and 300 times the number of geoscientific dataset names in the sample. We used the following formula to indicate the accuracy of training [64]:

$$accuracy = avg(\frac{|SP\_words|}{|All\_words|}) \tag{11}$$

where $|SP\_words|$ is the number of successfully predicted words for a geoscientific dataset name, $|All\_words|$ is the number of total words in the geoscientific dataset name, and $avg()$ is the average accuracy for 12,642 sample names. The prediction accuracy under different iterations and model sizes is shown in Figure 6.

As shown in Figure 6, as the model size increases, the training performance of the HTM model generally improves. When the number of columns of the HTM model is 5000, we can obtain the performance for different training iteration times. With increased training iteration times, the performance of the HTM model also generally improves; the optimal number of training iteration times is 250, and more training times will result in overfitting. The parameters of the HTM model are shown in Table 1 (the meanings of the parameters are given in detail in Appendix A). After the training, we obtained an HTM model that had learned the naming rule and combination patterns of different elements of the geoscientific dataset.

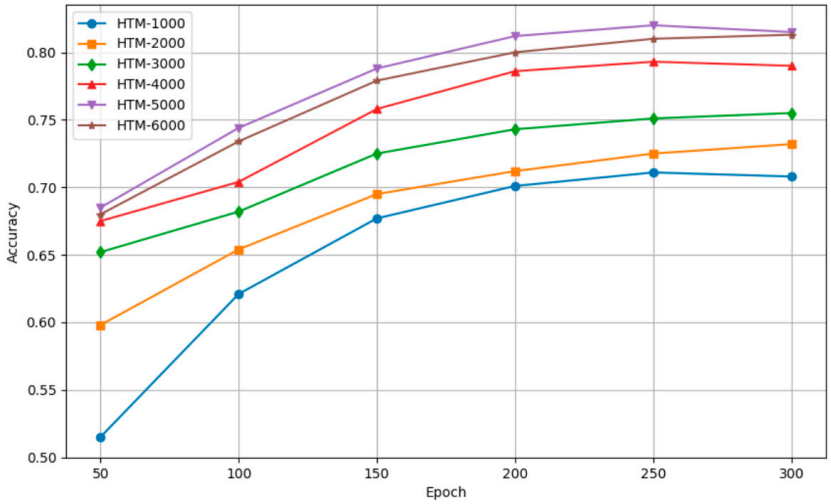


**Figure 6.** Prediction accuracy with different model sizes and training iteration times.

**Table 1.** Parameters of trained HTM model.

| Parameter Name | Value | Parameter Name | Value |
|---|---|---|---|
| Number of columns | 5000 | Number of cells per column | 6 |
| Input dimensions | 22,476 | Potential radius | 6 |
| Number of active columns | 48 | Connected threshold for synaptic permanence | 0.7 |
| Initial synaptic permanence | 0.1 | Dendritic segment activation threshold | 4 |
| Synaptic permanence increment | 0.1 | Synaptic permanence decrement | 0.1 |

## 5. Evaluation

In this research, the word-encoding method is the basis of the experiment. Whether or not the binary vector can represent the semantic features of the word has an important influence on the accuracy of extracting geoscientific dataset names. Therefore, we firstly evaluated the precision of the word-encoding method. Then we used the trained HTM model to extract geoscientific dataset names and evaluated its accuracy by comparing it with existing methods.

### *5.1. Evaluation of Word-Encoding Method Precision*

To evaluate the precision of the proposed word-encoding method, we compared it with the semantic folding method [68], which is the main text encoding method used for the HTM model. The key to the quality of encoding methods is in their ability to convey the semantics of words, so we can use the encoding binary vectors' cosine similarity to indicate the semantic similarity of word pairs. Because we extracted names in the geoscientific domain, we select the Geo-Terminology Relatedness Dataset (GTRD) [69] which contains 66 pairs of geographic terms and corresponding semantic similarities, as the benchmark.

First, we encoded the words in the GTRD using our method and the semantic folding method. Then, we calculated the cosine similarity of the encoders for the word pairs in GTRD separately. The result is shown in Table 2. Subsequently, we calculated the Pearson correlation coefficient between the cosine similarity and the standard similarity of word pairs in the benchmark dataset. Finally, to determine whether the correlation was statistically significant, we applied the $p$-value test [70] to the Pearson correlation coefficient. The result is shown in Table 3.

**Table 2.** Cosine similarity of two encoding methods for word pairs in GTRD.

| ID | Word One | Word Two | Standard Similarity | Semantic Folding Similarity | Our Encoding Method Similarity |
|---|---|---|---|---|---|
| 1 | waterway transportation | waterway transportation | 1 | 1 | 1 |
| 2 | oasis city | oasis city | 1 | 1 | 1 |
| 3 | port city | estuary city | 0.91 | 0.62 | 0.67 |
| 4 | tropical rainforest climate | equatorial rain climate | 0.87 | 0.53 | 0.33 |
| 5 | city | cities and towns | 0.86 | 0.58 | 0.50 |
| 6 | transportation | communication and transportation | 0.83 | 1 | 0.71 |
| 7 | nearshore environment | coastal environment | 0.78 | 0.88 | 0.75 |
| 8 | nearshore environment | sublittoral environment | 0.78 | 0.96 | 0.50 |
| 9 | plateau permafrost | frozen ground | 0.78 | 0.22 | 0.58 |
| 10 | cold wave | cold air mass | 0.77 | 0.64 | 0 |
| 11 | iron and steel industry | metallurgical industry | 0.73 | 0.56 | 0.72 |
| 12 | geographical environment | environment | 0.71 | 1 | 0.71 |
| 13 | highway transport | transport | 0.71 | 1 | 0.71 |
| 14 | semi-arid climate | steppe climate | 0.71 | 0.67 | 0.45 |
| 15 | climate | weather | 0.69 | 0.23 | 0 |
| 16 | milk industry | food industry | 0.68 | 0.7 | 0.75 |
| 17 | cultural landscape | landscape | 0.67 | 1 | 0.71 |
| 18 | processing industry | light industry | 0.66 | 0.66 | 0.58 |
| 19 | cold wave | disastrous weather | 0.65 | 0.28 | 0.03 |
| 20 | coal industry | heavy industry | 0.63 | 0.61 | 0.58 |
| 21 | farming industry | industry | 0.61 | 1 | 0.53 |
| 22 | gray desert soil | brown desert soil | 0.60 | 0.82 | 0.67 |
| 23 | marine environment | geographical environment | 0.60 | 0.61 | 0.50 |
| 24 | eco-environment | water environment | 0.59 | 0.61 | 0.58 |
| 25 | tropical soil | subtropical soil | 0.57 | 0.84 | 0.89 |
| | ... | ... | ... | ... | ... |
| 64 | desert climate | internal water transport | 0.03 | 0.26 | 0.01 |
| 65 | polar climate | mining industry | 0.02 | 0.25 | 0 |
| 66 | desert climate | labor-intensive industry | 0 | 0.2 | 0 |

**Table 3.** Pearson correlation coefficient and *p*-value test between proposed encoding method and semantic folding method on GTRD.

| Statistical Indicator | Proposed Encoding Method | Semantic Folding Method |
|---|---|---|
| Pearson correlation coefficient | 0.69 | 0.62 |
| *p*-value | $2.15 \times 10^{-10}$ | $3.61 \times 10^{-8}$ |

From Table 3, it is evident that on the benchmark GTRD dataset, our proposed encoding method outperforms the semantic folding approach in terms of accuracy.

### *5.2. Evaluation of Geoscientific Dataset Name-Extraction Accuracy*

To comprehensively evaluate the ability of HTM model to extract geoscientific dataset names, we compared it with two large language models, GPT-4 and Claude-3, which have excellent performance in named-entity recognition [71,72]. First, we created a baseline based on the geoscientific literature. The Journal of Geographical Sciences is a high-level academic journal in China (https://www.geog.com.cn/EN/home (accessed on 13 January 2024)) that publishes about 200 papers every year. We randomly selected 100 papers published in this journal from 2021 to 2023. From these articles, we extracted the paragraphs introducing the research data. Then, for each of the 100 pieces of text, we invited three experts to find and write down all of the geoscientific dataset names contained within it. A comparison showed that the three experts' results were almost identical. This result is used as the baseline to evaluate the ability of AI models to extract geoscientific dataset names.

The detailed process of extracting geoscientific dataset names is as follows: (1) For each of the 100 pieces of text, the NLPIR segmentation tool is used to split them into word sequences. The starting word of a piece of text is encoded by the method in Section 3.2.2 and the encoder is input to the trained HTM model. After processing by the classifier in the HTM model, the predicted encoder for the next word is output. Subsequently, the encoder for the actual next word is generated using the method described in Section 3.2.2. The similarity between the two encoders is calculated using the cosine similarity formula. If the similarity is greater than 0.4, the combination of the two adjacent words is deemed to conform to the naming pattern of geoscientific datasets, and they are part of a geoscientific dataset name. All words in the piece of text are dealt in the same way. If more than three consecutive words conformed to the pattern of scientific data naming, then the combination of these words is regarded as the name of a geoscientific dataset. (2) For GPT-4, we firstly use it directly to extract the geoscientific dataset names for every 100 pieces of text. The prompt for the zero-shot learning (ZSL) task is as follows: Please extract the names of the geoscientific datasets contained in the following paragraphs: {paragraphs containing geoscientific dataset names}. We then teach GPT-4 all 12,642 dataset names and use it again to extract the names of geoscientific datasets for every 100 pieces of text. In this context, for GPT-4, the prompt for the few-shot learning (FSL) task is as follows: The train-GeoscientificDatasetNames.txt file includes 12,642 geoscientific dataset names. Please learn the characteristics of these geoscientific dataset names. Now, based on the characteristics of the geoscientific dataset names mentioned and your knowledge, please extract the names of the geoscientific datasets contained in the following paragraphs: {paragraphs containing geoscientific dataset names}. The process of extracting geoscientific dataset names using GPT-4 is achieved by using the GPT-4 dialog window on the official website of OpenAI (https://openai.com/ (accessed on 25 March 2024)). (3) For Claude-3 (https://claude.ai (accessed on 5 June 2024)), we conduct similar experiments, which are tested on GPT-4.

To display our extraction results in an intuitive and representative way, we show the extraction results of two random pieces of text in Tables A2 and A3 in Appendix A. By comparing with the benchmark dataset names, it can be observed that our method can relatively accurately locate the positions of geoscientific dataset names and has relatively little redundancy when extracting specific geoscientific dataset names. However, when utilizing GPT-4 and Claude-3 for extracting geoscientific dataset names, issues such as redundancy or the omission of parts of words can occur. For instance, there are five correct geoscientific dataset names in Table A2, and our method can basically extract these names accurately. In contrast, the zero-shot learning (ZSL) method based on GPT-4 and Claude-3 can lose parts of the correct words of geoscientific dataset names. Both ZSL methods cannot correctly recognize "The spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m", and the GPT-4-based ZSL method cannot correctly recognize "1:1,000,000 soil type map". Additionally, the GPT-4-based ZSL and FSL methods can produce redundancy in recognizing geoscientific dataset names. For example, the correct geoscientific dataset name in Table A2 is "The Land Use and Cover Change (LUCC) dataset", but the recognized result is "The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences". The correct geoscientific dataset name in Table A3 is "The Fourth Forest Resources Inventory of Guangdong Province", but the recognized result is "The Fourth Forest Resources Inventory of Guangdong Province (partial pilot cities from 2013 to 2016, province-wide from 2017 to 2018)". Moreover, as shown in Tables A2 and A3, the ZSL method based on Claude-3 can extract more geoscientific datasets compared to the method based on GPT-4. However, it also demonstrates problems with redundancy and omission. GPT-4- and Claude-3-based methods extract more redundant or omissive results because the two methods may not understand the naming rules and combination patterns of various elements of geoscientific data. For example, the two methods will extract geoscientific

names containing the word "data" or "dataset," such as "NDVI data" and "precipitation dataset", which are not regarded as geoscientific dataset names because they do not include temporal or spatial coverage elements.

With all three methods, the extraction result contains three kinds of items: exact geoscientific dataset names, partially correct items (i.e., geoscientific dataset names with redundancy), and completely incorrect items. To quantitatively describe the extraction accuracy, we computed the precision, recall, and F1-score of the three methods.

Precision ($P$) represents the rate of correct results for all extracted dataset names, where $T$ is the number of correctly extracted names and $N$ is the number of all extracted names. The formula is as follows:

$$P = \frac{T}{N} \tag{12}$$

Recall ($R$) represents the ratio of correctly extracted geoscientific dataset names to the number of names in the benchmark, where $T$ is the number of correctly extracted names and $S$ is the number of names in the benchmark. The formula is as follows:

$$R = \frac{T}{S} \tag{13}$$

The F1-score is a composite indicator determined by a weighted average of precision and recall, which responds to the performance of the model. The formula is as follows:

$$F1\text{-score} = \frac{2PR}{(P+R)}. \tag{14}$$

The precision, recall, and F1-score of the three methods are shown in Table 4 and Figure 7. Comparison of the five methods on precision, recall, and F1-score.

**Table 4.** Statistics for the extraction results of the three methods for 100 randomly selected pieces of text.

| Statistical Indicator | Proposed Method | GPT-4 Based Zero-Shot Learning (ZSL) Method | GPT-4 Based Few-Shot Learning (FSL) Method | Claude-3 Based Zero-Shot Learning (ZSL) Method | Claude-3 Based Few-Shot Learning (FSL) Method |
|---|---|---|---|---|---|
| Number of benchmark geoscientific dataset names | 530 | 530 | 530 | 530 | 530 |
| Number of extracted geoscientific dataset names | 600 | 478 | 459 | 534 | 520 |
| Number of correctly extracted geoscientific dataset names | 411 | 340 | 345 | 368 | 378 |
| Precision (%) | 68.5 | 71.1 | 75.2 | 69.4 | 71.3 |
| Recall (%) | 77.5 | 64.2 | 65.1 | 68.9 | 72.7 |
| F1-score | 0.727 | 0.675 | 0.698 | 0.691 | 0.720 |

From Table 4 and Figure 7, it can be seen that GPT-4 and Claude-3 have better precision than our method because they extract fewer results. However, our method extracts the most correct geoscientific dataset names and performs better in terms of recall and F1-score. The precision of GPT-4 and Claude-3 is improved after few-shot learning (FSL). Our method can find more geoscientific dataset names, although some of them are not always exact, such as "data of precipitation, temperature and other daily observation data from national meteorological stations on the Qinghai Tibet Plateau", which is a partially correct name (the right name is "data of precipitation, temperature and other daily observation data from national meteorological stations on the Qinghai Tibet Plateau from 2020 to 2020") with a

rare combination of elements of geoscientific data. Overall, if the intention is to extract more correct datasets, researchers can consider using our method.

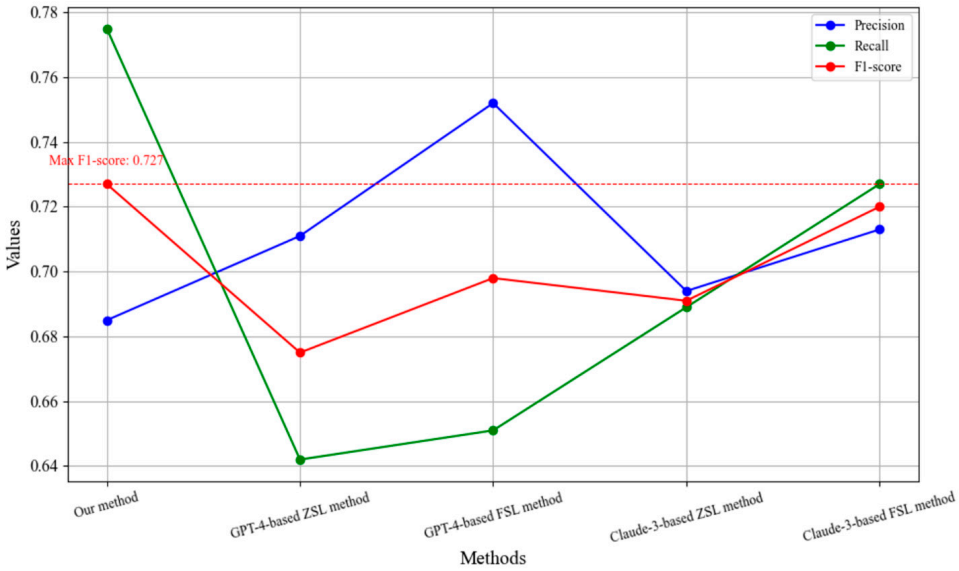


**Figure 7.** Comparison of the five methods on precision, recall, and F1-score.

## 6. Conclusions and Discussion

In this study, we propose an artificial neural network method for extracting geoscientific dataset names. Specifically, a new word-encoding method for the HTM model is proposed. The new word-encoding method uses Unicode to generate a binary vector for characters. The encoder is input into a human brain-inspired neural network (the HTM model) to generate a predictive vector. In our research, we ingeniously transform the predictive vector into the encoder of the next word by using the classifier of backpropagation (BP). By computing the similarity between the encoders of the next predictive word and the next real word, we can determine whether the combination of two adjacent words is part of a geoscientific dataset name and thus extract geoscientific dataset names from unstructured text. Through training with more than 12,000 dataset names, the HTM model learns various naming patterns and can be used to extract geoscientific dataset names from the literature. Finally, we compare our method with two existing methods—GPT-4 and Claude-3. The result shows that our method outperforms the existing methods in recall and F1-score.

Our new artificial neural network method for extracting geoscientific dataset names achieves a higher F1-score on the task of extracting the names of geoscientific datasets, while the large language model methods achieve higher accuracy because they extract fewer results. Our model requires a smaller training corpus and computing power than the large language models. Our method can be utilized not only for extracting knowledge from the literature and reproducing geoscientific studies but also for carrying out scientific data bibliometrics and other tasks related to geospatial information.

However, the method has the following limitations: First, while our encoding method demonstrates higher accuracy in encoding Chinese characters compared to the semantic folding method, the accuracy in encoding English characters needs improvement. Additionally, although the numerical mapping method reduces the encoding length, the length of encoded words remains relatively large, resulting in a longer training time for the HTM model. Second, while our method achieves a higher F1-score in extracting geoscientific dataset names compared to larger language models such as GPT-4, it has lower extraction accuracy. Moreover, as large language models learn more knowledge, their extraction accuracy continues to improve. Therefore, it is necessary to improve our method. In the future, we plan to modify existing word-embedding

techniques such as Word2vec to create a new word-encoding method for the HTM model. This method will be able to not only encode multiple languages but also reduce the encoding length and improve accuracy, thereby further improving the precision of the HTM model in extracting dataset names.

**Author Contributions:** Conceptualization, Zugang Chen, Xinqian Wu, and Kai Wu; methodology, Zugang Chen, Xinqian Wu, and Kai Wu; software, Zugang Chen, Kai Wu, and Hang Feng; validation, Kai Wu; formal analysis, Zugang Chen, Xinqian Wu, and Kai Wu; resources, Guoqing Li, Zugang Chen, and Xinqian Wu; data curation, Kai Wu and Haodong Wang; writing—original draft preparation, Kai Wu; writing—review and editing, Zugang Chen, Xinqian Wu, and Kai Wu; visualization, Kai Wu; supervision, Zugang Chen and Xinqian Wu; project administration, Zugang Chen and Xinqian Wu; funding acquisition, Zugang Chen, Jing Li, and Shaohua Wang. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (Grant No. 42201505), the Natural Science Foundation of Hainan Province of China (Grant No. 622QN352), and the National Key Research and Development Program of China (Grant No. 2021YFF070420304).

**Data Availability Statement:** The datasets and code can be found at https://github.com/WkStatisticsRoad/HTMTextExtract (accessed on 8 June 2024).

**Acknowledgments:** We thank the anonymous reviewers and all of the editors who participated in the revision process.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

Table A1 lists the main parameter settings for the hierarchical temporal memory (HTM) model utilized in this study. The parameters are input dimensions, number of columns, number of cells per column, potential radius, number of active columns, connected threshold for synaptic permanence, initial synaptic permanence, dendritic segment activation threshold, and synaptic permanence increment and decrement. In the experiment of extracting geoscientific dataset names, these parameters were employed to configure the HTM model. The specific explanation and setting of each parameter are as follows:

**Table A1.** Parameters of created HTM model.

| Parameter Name | Value | Parameter Name | Value |
|---|---|---|---|
| Number of columns | 5000 | Number of cells per column | 6 |
| Input dimensions | 22,476 | Potential radius | 6 |
| Number of active columns | 48 | Connected threshold for synaptic permanence | 0.7 |
| Initial synaptic permanence | 0.1 | Dendritic segment activation threshold | 4 |
| Synaptic permanence increment | 0.1 | Synaptic permanence decrement | 0.1 |

Input dimensions: This parameter represents the dimensionality of the input data, i.e., the number of input bits processed by the HTM model. We encoded 3596 commonly used Chinese characters and set the number of continuously active bits to 151, resulting in a character encoding length of 3746. In this experiment, the maximum character length of a word was set to six, resulting in a word-encoding length of 22,476, which was the input dimension for the HTM model.

Number of columns: This parameter determines the number of columns in the spatial pooler of the HTM model. Having more columns allows the HTM model to handle more features or patterns simultaneously, but also increases computational complexity. In this experiment, after tuning the parameters, the number of columns was set to 5000.

Number of cells per column: Each column in the spatial pooler contains a set of cells. This parameter specifies how many cells are contained in each column. Typically, the number of cells per column is not very large, usually ranging from a few to several dozen, as more cells will increase computational cost. In this experiment, based on experience and parameter optimization, the number of cells per column was set to six.

Potential radius: The potential radius defines the neighborhood size within which a cell in the spatial pooler can form connections. A larger potential radius allows cells to form connections with a broader area, but an excessively large potential radius can lead to overfitting. In this experiment, based on experience and parameter optimization, the potential radius was set to six.

Number of active columns: The number of active columns refers to the maximum number of columns that can remain active within a local inhibition area. By controlling the number of active columns, the sparsity and representation capacity of the HTM model can be adjusted. In this experiment, based on experience and parameter optimization, the number of active columns was set to 48.

Connected threshold for synaptic permanence: If the connected threshold for synaptic permanence (values in [0, 1]) is set to 0.7, initially 70% of the potential synapses are connected. If the permanence value for a synapse is greater than this threshold, it is considered connected. In this experiment, based on experience and parameter optimization, the connected threshold was set to 0.7.

Initial synaptic permanence: This parameter indicates the initial permanence value for a newly formed synapse. It is usually set to a low value to ensure that the initial connections are neither too strong nor too weak. In this experiment, the initial synaptic permanence was set to 0.1.

Dendritic segment activation threshold: This parameter indicates how many cells need to be active within a time step in order to activate the entire column. Its value is related to the number of cells in the column, and the threshold cannot exceed this number. In this experiment, based on experience and parameter optimization, the dendritic segment activation threshold was set to four.

Synaptic permanence increment: This parameter indicates how much a synapse is strengthened when the cell is activated. It is typically set to a positive value of less than 1 to ensure that synaptic connections gradually strengthen during learning. In this experiment, the synaptic permanence increment was set to 0.1.

Synaptic permanence decrement: This parameter indicates how much a synapse is weakened when the cell is inactivated. It is typically set to a positive value of less than one to ensure that inactive synaptic connections gradually weaken during learning. In this experiment, the synaptic permanence decrement was set to 0.1.

In summary, Table A1 describes the key parameters of the HTM model, including input dimensions, number of cells per column, and synaptic connectivity thresholds. These parameters are crucial in controlling the model's ability to encode input patterns, perform pattern recognition, and adapt to changes in the input data. Understanding and appropriately setting these parameters are essential for the effective application of the HTM model.

**Table A2.** Results of five methods for extracting geoscientific dataset names.

| Extraction Method | Names of Benchmark Geoscientific Datasets | Results of Five Recognition Methods |
| --- | --- | --- |
| Proposed method | The Land Use and Cover Change (LUCC) dataset, The China Annual Land Cover Dataset (CLCD), The regional data of China from GlobeLand30, The spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m, 1:1,000,000 soil type map | The data sources for this article are as follows: (1) Ecosystem-type data. **The Land Use and Cover Change (LUCC) dataset** from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; **the China Annual Land Cover Dataset (CLCD)** based on satellite data from Google Earth Engine (GEE); and the **regional data of** China **from GlobeLand30**. Based on the aforementioned land use and cover types, they were transformed into ecosystem types such as cropland, forest, grassland, wetland, and settlement, obtaining the **spatial distribution data of the Qinghai-Tibet Plateau ecosystems** for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data. The soil property tables and spatial distribution data attached to the **1:1,000,000 soil type map** come from the Western Environmental and Ecological Science Data Center, and the soil erodibility factor was estimated using the Nomograph method. |
| GPT-4-based zero-shot learning (ZSL) method | | The data sources for this article are as follows: (1) Ecosystem-type data. **The Land Use and Cover Change (LUCC) dataset** from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; **the China Annual Land Cover Dataset (CLCD)** based on satellite data from Google Earth Engine (GEE); and **the regional data of China from GlobeLand30**. Based on the aforementioned land use and cover types, they were transformed into ecosystem types such as cropland, forest, grassland, wetland, and settlement, obtaining the spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data. The soil property tables and spatial distribution data attached to the 1:1,000,000 soil type map come from the Western Environmental and Ecological Science Data Center, and the soil erodibility factor was estimated using the Nomograph method. |
| GPT-4-based few-shot learning (FSL) method | | The data sources for this article are as follows: (1) Ecosystem-type data. **The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; the China Annual Land Cover Dataset (CLCD) based on satellite data from Google Earth Engine (GEE); and the regional data of China from GlobeLand30**. Based on the aforementioned land use and cover types, they were transformed into ecosystem types such as cropland, forest, grassland, wetland, and settlement, obtaining the spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data. The soil property tables and spatial distribution data attached to the **1:1,000,000 soil type map come from the Western Environmental and Ecological Science Data Center**, and the soil erodibility factor was estimated using the Nomograph method. |
| Proposed method | The Land Use and Cover Change (LUCC) dataset, The China Annual Land Cover Dataset (CLCD), The regional data of China from GlobeLand30, The spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m, 1:1,000,000 soil type map | Data sources for this article: (1) Ecosystem-type data: **The Land Use and Cover Change (LUCC) dataset** from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; **the China Annual Land Cover Dataset (CLCD)** based on satellite data from Google Earth Engine (GEE); and the **regional data of** China **from GlobeLand30**. Based on land use and cover types, they were transformed into ecosystem types including cropland, forest, grassland, wetland, and settlement, to obtain **spatial distribution data of the Qinghai-Tibet Plateau ecosystems** for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data: Soil property tables and spatial distribution data attached to **1:1,000,000 soil type map** come from Western Environmental and Ecological Science Data Center, and soil erodibility factor was estimated using the nomograph method. |
| Claude-3-based zero-shot learning (ZSL) method | | Data sources for this article: (1) Ecosystem-type data: **The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences**; the **China Annual Land Cover Dataset (CLCD) based on satellite data from Google Earth Engine (GEE)**; and **the regional data of China from GlobeLand30**. Based on land use and cover types, they were transformed into ecosystem types including cropland, forest, grassland, wetland, and settlement to obtain spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data: The soil property tables and spatial distribution data attached to **1:1,000,000 soil type map come from the Western Environmental and Ecological Science Data Center**, and the soil erodibility factor was estimated using the nomograph method. |
| Claude-3-based few-shot learning (FSL) method | | Data sources for this article: (1) Ecosystem-type data: **The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences**; the **China Annual Land Cover Dataset (CLCD) based on satellite data from Google Earth Engine (GEE)**; and **the regional data of China from GlobeLand30**. Based on land use and cover types, they were transformed into ecosystem types including cropland, forest, grassland, wetland, and settlement, to obtain spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data: **The soil property tables and spatial distribution data** attached to the **1:1,000,000 soil type map** come **from the Western Environmental and Ecological Science Data Center**, and soil erodibility factor was estimated using the nomograph method. |

**Table A3.** Results of five methods for extracting geoscientific dataset names.

| Extraction Method | Names of Benchmark Geoscientific Datasets | Results of Five Recognition Methods |
| --- | --- | --- |
| Proposed method | The Fourth Forest Resources Inventory of Guangdong Province, The second National Soil Survey Data for Guangdong (1979–1985), Harmonized World Soil Database, The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018) | **The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) is sourced from Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 $km^2$ each. **The second National Soil Survey data for Guangdong** (1979–1985), with information such as soil organic carbon content and soil bulk density, is sourced from the **Harmonized World Soil Database** by the UN Food and Agriculture Organization, with a spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of** China's **high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced from the Soil Data Center of the National Earth System Science Data Center (http://soil.geodata.cn/ (accessed on 12 April 2022)), with a spatial resolution of 1 km × 1 km. |
| GPT-4-based zero-shot learning (ZSL) method | | **The Fourth Forest Resources Inventory of Guangdong Province (partial pilot cities 2013–2016, province-wide 2017–2018)** was sourced from the Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 $km^2$ each. **The second National Soil Survey data for Guangdong (1979–1985)**, with information including soil organic carbon content and soil bulk density, sourced from the Harmonized World Soil Database by the UN Food and Agriculture Organization, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced from the Soil Data Center of the National Earth System Science Data Center (http://soil.geodata.cn/ (accessed on 12 April 2022)), with a spatial resolution of 1 km × 1 km. |
| GPT-4-based few-shot learning (FSL) method | | **The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from the Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 $km^2$ each. **The second National Soil Survey data for Guangdong** (1979–1985), with information including soil organic carbon content and bulk density, sourced from **the Harmonized World Soil Database by the UN Food and Agriculture Organization**, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of China's high-resolution national soil information grid for Guangdong** (2010–2018) also provides information on soil organic carbon content and bulk density, sourced from the **Soil Data Center of the National Earth System Science Data Center** (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km. |
| Proposed method | The Fourth Forest Resources Inventory of Guangdong Province, The second National Soil Survey Data for Guangdong (1979–1985), Harmonized World Soil Database, The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018) | **The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 $km^2$ each. **The second National Soil Survey data for Guangdong** (1979–1985), with information including soil organic carbon content and bulk density, sourced from the **Harmonized World Soil Database** by the UN Food and Agriculture Organization, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of** China's **high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced from the Soil Data Center of the National Earth System Science Data Center (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km. |
| Claude-3-based zero-shot learning (ZSL) method | | The **Fourth Forest Resources Inventory of Guangdong Province (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from Guangdong Provincial Department of Natural Resources** in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 $km^2$ each. The **second National Soil Survey data for Guangdong (1979–1985)**, with information including soil organic carbon content and bulk density, sourced from **the Harmonized World Soil Database by the UN Food and Agriculture Organization**, with spatial resolution of 0.25 km × 0.25 km. The basic attribute dataset of **China's high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced **from the Soil Data Center of the National Earth System Science Data Center** (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km. |
| Claude-3-based few-shot learning (FSL) method | | **The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 $km^2$ each. **The second National Soil Survey data for Guangdong (1979–1985)**, with information including soil organic carbon content and bulk density, **sourced from the Harmonized World Soil Database by the UN Food and Agriculture Organization**, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, **sourced from the Soil Data Center of the National Earth System Science Data Center** (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km. |

To display our extraction results in a more intuitive and representative way, we randomly selected the extraction results from two paragraphs. Tables A2 and A3 in Appendix A show part of the recognition results of the three methods (Represent the recognized results in bold black text). From these tables, it can be observed that the proposed method can relatively accurately locate the positions of geoscientific dataset names and has relatively low redundancy when identifying specific geoscientific dataset names. A detailed analysis can be found in Section 5.2.

## References


1. Li, J.; Zhou, C. Analysis on the Characteristics of Geospatial Data. *Sci. Geogr. Sin.* **1999**, *19*, 158–162. [CrossRef]
2. Lu, M.; Appel, M.; Pebesma, E. Multidimensional Arrays for Analysing Geoscientific Data. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 313. [CrossRef]
3. Buttlar, J.v.; Zscheischler, J.; Mahecha, M.D. An extended approach for spatiotemporal gapfilling: Dealing with large and systematic gaps in geoscientific datasets. *Nonlin. Process. Geophys.* **2014**, *21*, 203–215. [CrossRef]
4. Sun, K.; Zhu, Y.; Pan, P.; Hou, Z.; Wang, D.; Li, W.; Song, J. Geospatial data ontology: The semantic foundation of geospatial data integration and sharing. *Big Earth Data* **2019**, *3*, 269–296. [CrossRef]
5. Kostoff, R.N. Role of Technical Literature in Science and Technology Development and Exploitation. *J. Inf. Sci.* **2003**, *29*, 223–228. [CrossRef]
6. Ning, B.; Zhao, Y. To Embrace Open Science More Closely. *Innovation* **2020**, *1*, 100012. [CrossRef] [PubMed]
7. Morse, P.; Reading, A.; Lueg, C. Animated analysis of geoscientific datasets: An interactive graphical application. *Comput. Geosci.* **2017**, *109*, 87–94. [CrossRef]
8. Konkol, M.; Kray, C.; Pfeiffer, M. Computational reproducibility in geoscientific papers: Insights from a series of studies with geoscientists and a reproduction study. *Int. J. Geogr. Inf. Sci.* **2019**, *33*, 408–429. [CrossRef]
9. Gil, Y.; David, C.H.; Demir, I.; Essawy, B.T.; Fulweiler, R.W.; Goodall, J.L.; Karlstrom, L.; Lee, H.; Mills, H.J.; Oh, J.H.; et al. Toward the Geoscience Paper of the Future: Best practices for document ing and sharing research from data to software to provenance. *Earth Space Sci.* **2016**, *3*, 388–415. [CrossRef]
10. Zhang, S.; Xu, H.; Jia, Y.; Wen, Y.; Wang, D.; Fu, L.; Wang, X.; Zhou, C. GeoDeepShovel: A platform for building scientific database from geoscience literature with AI assistance. *Geosci. Data J.* **2023**, *10*, 519–537. [CrossRef]
11. Tao, L.; Xie, Z.; Xu, D.; Ma, K.; Qiu, Q.; Pan, S.; Huang, B. Geographic Named Entity Recognition by Employing Natural Language Processing and an Improved BERT Model. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 598. [CrossRef]
12. Chung, C.-J.F.; Fabbri, A.G. The representation of geoscience information for data integration. *Nonrenew. Resour.* **1993**, *2*, 122–139. [CrossRef]
13. Arias, A.; Dini, I.; Casini, M.; Fiordelisi, A.; Perticone, I.; Pisano, A. Geoscientific Feature Update of the Larderello-Travale Geothermal System (Italy) for a Regional Numerical Modeling. In Proceedings of the World Geothermal Congress 2010, Bali, Indonesia, 25–30 April 2010.
14. Färber, M.; Albers, A.; Schüber, F. Identifying Used Methods and Datasets in Scientific Publications. In Proceedings of the SDU@AAAI Workshop on Scientific Document Understanding, Online, 19 February 2021.
15. Heddes, J.; Meerdink, P.; Pieters, M.; Marx, M. The Automatic Detection of Dataset Names in Scientific Articles. *Data* **2021**, *6*, 84. [CrossRef]
16. George, D.; Hawkins, J. A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005.
17. George, D.; Hawkins, J. Towards a Mathematical Theory of Cortical Micro-circuits. *PLoS Comput. Biol.* **2009**, *5*, e1000532. [CrossRef] [PubMed]
18. Klukas, M.; Lewis, M.; Fiete, I. Efficient and flexible representation of higher-dimensional cognitive variables with grid cells. *PLOS Comput. Biol.* **2020**, *16*, e1007796. [CrossRef]
19. Cao, Q.; Wang, S.; Chen, Z.; Li, G.; Li, J. The Method of Extracting Names of Geo-science Data based on Regular Expressions. *J. Geo-Inf. Sci.* **2023**, *25*, 1601–1610. [CrossRef]
20. Afzal, M.T.; Maurer, H.A.; Balke, W.-T.; Kulathuramaiyer, N. Rule based Autonomous Citation Mining with TIERL. *J. Digit. Inf. Manag.* **2010**, *8*, 196–204.
21. Fries, J.A.; Varma, P.; Chen, V.S.; Xiao, K.; Tejeda, H.; Saha, P.; Dunnmon, J.A.; Chubb, H.; Maskatia, S.A.; Fiterau, M.; et al. Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences. *Nat. Commun.* **2019**, *10*, 3111. [CrossRef] [PubMed]
22. Soni, A.; Viswanathan, D.; Pachaiyappan, N.; Natarajan, S. A Comparison of Weak Supervision methods for Knowledge Base Construction. In Proceedings of the AKBC@NAACL-HLT, San Diego, CA, USA, 12–17 June 2016.
23. Zech, J.R.; Pain, M.; Titano, J.J.; Badgeley, M.A.; Schefflein, J.; Su, A.; Costa, A.B.; Bederson, J.B.; Lehár, J.; Oermann, E.K. Natural Language-based Machine Learning Models for the Annotation of Clinical Radiology Reports. *Radiology* **2018**, *287*, 570–580. [CrossRef]

24. Cui, B.-G.; Chen, X. An Improved Hidden Markov Model for Literature Metadata Extraction. In Proceedings of the 6th International Conference on Advanced Intelligent Computing Theories and Applications: Intelligent Computing, Changsha, China, 18 August 2010; pp. 205–212.
25. Zhang, K.; Xu, H.; Tang, J.; Li, J.-Z. Keyword Extraction Using Support Vector Machine. In Proceedings of the Interational Conference on Web-Age Information Management, Hong Kong, China, 17–19 June 2006; pp. 85–96.
26. Kaur, J.; Gupta, V. Effective Approaches for Extraction of Keywords. *Int. J. Comput. Sci.* **2010**, *7*, 144–148.
27. Han, H.; Giles, C.L.; Manavoglu, E.; Zha, H.; Zhang, Z.; Fox, E.A. Automatic document metadata extraction using support vector machines. In Proceedings of the 2003 Joint Conference on Digital Libraries, Houston, TX, USA, 20 June 2003; pp. 37–48.
28. Shinde, P.; Shah, S. A Review of Machine Learning and Deep Learning Applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 25 April 2018; pp. 1–6.
29. Zhao, Z.; Yang, Z.; Luo, L.; Wang, L.; Zhang, Y.; Lin, H.; Wang, J. Disease named entity recognition from biomedical literature using a novel convolutional neural network. *BMC Med. Genom.* **2017**, *10*, 73. [CrossRef]
30. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
31. Delgado, J.; Ebreso, U.; Kumar, Y.; Li, J.J.; Morreale, P. Preliminary Results of Applying Transformers to Geoscience and Earth Science Data. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022; pp. 284–288.
32. Bhattarai, K.; Oh, I.; Sierra, J.; Payne, P.; Abrams, Z.; Lai, A. Leveraging GPT-4 for Identifying Clinical Phenotypes in Electronic Health Records: A Performance Comparison between GPT-4, GPT-3.5-turbo and spaCy's Rule-based & Machine Learning-based methods. *bioRxiv* **2023**. *preprint*. [CrossRef]
33. Yao, R.; Hou, L.; Ye, Y.; Zhang, J.; Wu, J. Method and Dataset Mining in Scientific Papers. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 6260–6262.
34. Kumar, S.; Ghosal, T.; Ekbal, A. DataQuest: An Approach to Automatically Extract Dataset Mentions from Scientific Papers. In Proceedings of the International Conference on Asian Digital Libraries, Hanoi, Vietnam, 30 November 2021; pp. 43–53.
35. Younes, Y.; Scherp, A. Question Answering Versus Named Entity Recognition for Extracting Unknown Datasets. *IEEE Access* **2023**, *11*, 92775–92787. [CrossRef]
36. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3606–3611.
37. Lin, Z.; Deng, C.; Zhou, L.; Zhang, T.; Xu, Y.; Xu, Y.; He, Z.; Shi, Y.; Dai, B.; Song, Y.; et al. GeoGalactica: A Scientific Large Language Model in Geoscience. *arXiv* **2023**, arXiv:2401.00434.
38. Ahsan, M.M.T.; Rahaman, M.S.; Anjum, N. From ChatGPT-3 to GPT-4: A Significant Leap in AI-Driven NLP Tools. *J. Eng. Emerg. Technol.* **2023**, *1*, 50–60. [CrossRef]
39. Hosseini, S.; Najafipour, S.; Cheung, N.-M.; Yin, H.; Kangavari, M.R.; Zhou, X. TEAGS: Time-aware text embedding approach to generate subgraphs. *Data Min. Knowl. Discov.* **2020**, *34*, 1136–1174. [CrossRef]
40. Najafipour, S.; Hosseini, S.; Hua, W.; Kangavari, M.R.; Zhou, X. SoulMate: Short-Text Author Linking Through Multi-Aspect Temporal-Textual Embedding. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 448–461. [CrossRef]
41. Hosseini, S.; Yin, H.; Zhou, X.; Sadiq, S.; Kangavari, M.R.; Cheung, N.-M. Leveraging multi-aspect time-related influence in location recommendation. *World Wide Web* **2019**, *22*, 1001–1028. [CrossRef]
42. Hosseini, S.; Yin, H.; Zhang, M.; Zhou, X.; Sadiq, S. Jointly Modeling Heterogeneous Temporal Properties in Location Recommendation. In *Database Systems for Advanced Applications*; Springer: Cham, Switzerland, 2017; pp. 490–506.
43. Saaki, M.; Hosseini, S.; Rahmani, S.; Kangavari, M.R.; Hua, W.; Zhou, X. Value-Wise ConvNet for Transformer Models: An Infinite Time-Aware Recommender System. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 9932–9945. [CrossRef]
44. Malawade, A.; Costa, N.; Muthirayan, D.; Khargonekar, P.; Al Faruque, M.A. Neuroscience-Inspired Algorithms for the Predictive Maintenance of Manufacturing Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7980–7990. [CrossRef]
45. Zeng, H.; Zhao, X.; Wang, L. Multivariate Time Series Anomaly Detection On Improved HTM Model. In Proceedings of the 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Fuzhou, China, 24–26 September 2021; pp. 759–763.
46. Krestinskaya, O.; Ibrayev, T.; James, A.P. Hierarchical Temporal Memory Features with Memristor Logic Circuits for Pattern Recognition. *IEEE Trans. Comput. -Aided Des. Integr. Circuits Syst.* **2018**, *37*, 1143–1156. [CrossRef]
47. Irmanova, A.; Krestinskaya, O.; James, A.P. Image Based HTM Word Recognizer for Language Processing. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics—Asia (ICCE-Asia), Jeju, Republic of Korea, 24–26 June 2018; pp. 206–212.
48. Almehmadi, A.; Bosakowski, T.; Sedky, M.; Bastaki, B.B. HTM Based Anomaly Detecting Model for Traffic Congestion. In Proceedings of the 2020 4th International Conference on Cloud and Big Data Computing, Virtual, UK, 26–28 August 2020; pp. 97–101.
49. Szoplák, Z.; Andrejková, G. Anomaly Detection in Text Documents using HTM Networks. In Proceedings of the Conference on Theory and Practice of Information Technologies, Muran, Slovakia, 24–28 September 2021; pp. 20–28.

50. Khan, H.M.; Khan, F.M.; Khan, A.; Asghar, M.Z.; Alghazzawi, D.M. Anomalous Behavior Detection Framework Using HTM-Based Semantic Folding Technique. *Comput. Math. Methods Med.* **2021**, *2021*, 5585238. [CrossRef] [PubMed]
51. Mackenzie, J.; Roddick, J.F.; Zito, R. An Evaluation of HTM and LSTM for Short-Term Arterial Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1847–1857. [CrossRef]
52. Zyarah, A.M.; Kudithipudi, D. Neuromorphic Architecture for the Hierarchical Temporal Memory. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *3*, 4–14. [CrossRef]
53. Hawkins, J.; George, D.; Niemasik, J. Sequence memory for prediction, inference and behaviour. *Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci.* **2009**, *364*, 1203–1209. [CrossRef]
54. Kostavelis, I.; Gasteratos, A. On the optimization of Hierarchical Temporal Memory. *Pattern Recognition Letters* **2012**, *33*, 670–676. [CrossRef]
55. Zhou, L.; Zhang, D. NLPIR: A Theoretical Framework for Applying Natural Language Processing to Information Retrieval. *JASIST* **2003**, *54*, 115–123. [CrossRef]
56. Hawkins, J.; George, D. *Hierarchical Temporal Memory Concepts, Theory, and Terminology*; Numenta: Redwood City, CA, USA, 2006.
57. Hawkins, J.; Ahmad, S. Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex. *Front. Neural Circuits* **2016**, *10*, 23. [CrossRef]
58. Cui, Y.; Ahmad, S.; Hawkins, J. The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding. *Front. Comput. Neurosci.* **2017**, *11*, 111. [CrossRef]
59. Kanerva, P. Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cogn. Comput.* **2009**, *1*, 139–159. [CrossRef]
60. Purdy, S. Encoding data for HTM systems. *arXiv* **2016**, arXiv:1602.05925. [CrossRef]
61. Bettels, J.; Bish, F.A. Unicode: A Universal Character Code. *Digit. Tech. J. Digit. Equip. Corp.* **1993**, *5*, 21–31.
62. Allen, J.D.; Anderson, D.; Becker, J.; Cook, R.; Davis, M.; Edberg, P.; Everson, M.; Freytag, A.; Iancu, L.; Ishida, R.; et al. *The Unicode Standard, Version 7.0*; Unicode: Mountain View, CA, USA, 2014.
63. Cui, Y.; Ahmad, S.; Hawkins, J. Continuous online sequence learning with an unsupervised neural network model. *Neural Comput.* **2016**, *28*, 2474–2504. [CrossRef] [PubMed]
64. Niu, D.; Yang, L.; Cai, T.; Li, L.; Wu, X.; Wang, Z. A New Hierarchical Temporal Memory Algorithm Based on Activation Intensity. *Comput. Intell. Neurosci.* **2022**, *2022*, 6072316. [PubMed]
65. Wielgosz, M.; Pietroń, M. Using Spatial Pooler of Hierarchical Temporal Memory to classify noisy videos with predefined complexity. *Neurocomputing* **2017**, *240*, 84–97. [CrossRef]
66. Wright, L.G.; Onodera, T.; Stein, M.M.; Wang, T.; Schachter, D.T.; Hu, Z.; McMahon, P.L. Deep physical neural networks trained with backpropagation. *Nature* **2021**, *601*, 549–555. [CrossRef]
67. Wen, J.; Zhao Jia, L.; Luo Si, W.; Han, Z. The improvements of BP neural network learning algorithm. In Proceedings of the WCC 2000—ICSP 2000 5th International Conference on Signal Processing Proceedings and 16th World Computer Congress 2000, Beijing, China, 21–25 August 2000; pp. 1647–1649.
68. Webber, F.D.S. Semantic Folding Theory And its Application in Semantic Fingerprinting. *arXiv* **2015**, arXiv:1511.08855. [CrossRef]
69. Chen, Z.; Song, J.; Yang, Y. An Approach to Measuring Semantic Relatedness of Geographic Terminologies Using a Thesaurus and Lexical Database Sources. *ISPRS Int. J. Geo Inf.* **2018**, *7*, 98.
70. Greenland, S.; Senn, S.J.; Rothman, K.J.; Carlin, J.B.; Poole, C.; Goodman, S.N.; Altman, D.G. Statistical tests, P values, confidence intervals, and power: A guide to misinterpretations. *Eur. J. Epidemiol.* **2016**, *31*, 337–350.
71. Wang, S.; Sun, X.; Li, X.; Ouyang, R.; Wu, F.; Zhang, T.; Li, J.; Wang, G. GPT-NER: Named Entity Recognition via Large Language Models. *arXiv* **2023**, arXiv:2304.10428. [CrossRef]
72. Ashok, D.; Lipton, Z.C. PromptNER: Prompting for Named Entity Recognition. *arXiv* **2023**, arXiv:2305.15444. [CrossRef]