Adaptive Chinese Pinyin IME for Most Similar Representation

DONGSHENG JIANG, XINYU CHENG, AND TIANYI HAN
State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
Key Laboratory of Intelligent Medical Image Analysis and Precise Diagnosis of Guizhou Province, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
Corresponding author: Xinyu Cheng (xycheng@gzu.edu.cn)

This work was supported by the Guizhou Science and Technology Plan Project Qianke-he[2018]5301.

ABSTRACT Many neural-network approaches are used for Pinyin-to-character (P2C) conversion in Chinese input method engines (IMEs). However, in previous research, the conversion effectiveness of neural network P2C models relies on adequate training data. Unfortunately, neural networks cannot maintain high performance with conversions across users and domains. In this study, we propose a method for improving the efficiency of model conversion and tracking user behavior based on dynamic storage and representations that can be updated using historical information from user input. Our experimental results show that our technique tracks user behavior and has strong domain adaptability without requiring additional training. For the cross-domain datasets Touchpal, cMedQA1.0, CAIL2019, compared with the direct use of neural network, its indicators, Top-1 MIU-Acc, CA and KySS, are improved by at least 20.0%, 8.1%, 18.3%, respectively, and the results are close to the in-domain training of the model. Furthermore, compared with the traditional methods On-OMWA and Google IME, this method improves at least 7.8%, 2.0%, 11.9% and 3.2%, 0.7%, 13.9% in Top-1 MIU-Acc, CA and Kyss, respectively. This demonstrates that the proposed method is superior to existing models in terms of conversion accuracy and generality, and can point a new path for P2C platforms.

INDEX TERMS Chinese input method engine, domain adaptability, neural network representation, user experience.

I. INTRODUCTION
As the most mainstream Chinese input method at present, Chinese Pinyin input method has attracted lots of interests in both academia and industry. However, typing tens of thousands of Chinese characters into an electronic device using a 26-key Latin-style keyboard is not an easy task [1]. The Chinese Pinyin input method engine (IME) provides a solution by utilizing the fact that Pinyin comprises Latin letters and splitting their sequences into multiple syllables, which can then be mapped to different Chinese characters.

Pinyin-to-character (P2C) IMEs aim to provide high-precision conversion between Pinyin syllables and Chinese characters sequences. However, the P2C process faces a well-known challenge in that there is considerable ambiguity in the mapping of syllables to characters. Although there are approximately 6,000 most-commonly used Chinese characters, there exist only approximately 400 corresponding Pinyin syllables [2]. Thus, an average Pinyin syllable may correspond to dozens of Chinese characters. In practice, Pinyin input methods can be used to improve input efficiency by decoding longer Pinyin sequences. When a given Pinyin sequence lengthens, the corresponding list of legal character sequences is significantly reduced [3]. As summarized in Table 1, entering different numbers of syllables into an IME results in different character sequences. For instance, entering the six Pinyin syllables, “zi’ran’yu’yan’chu’li,” into an IME results in the output sequence, “自然语言处理,” a reasonable inference from the input syllables. This feature is accommodating for efficient and correct decoding, as each of the six syllables can be mapped to at least a dozen different Chinese characters.
Many neural-network approaches have been applied to P2C, and many experiments have shown that the conversion performance of such networks exceeds that of conventional statistical models [4], [5], [6], [7]. However, such methods can only achieve superior performance after training on specific domain data, and the model performance rapidly decays when P2C conversion is performed on different domain data. They do not consider differences in data distributions among different domains and users, within which the expected output of a given Pinyin input can differ. For example, the term “dian’xian’” in Pinyin often corresponds to the Chinese “电线” as it relates to news and user chats, but it corresponds to “癫病” in the field of medicine. The inability of a trained neural-network model to adaptively process different domain data and new data generated by users in real time is a significant drawback. Therefore, to continuously train the neural network model online [3] is a new idea, but maintaining an online trained neural network for each user to track user performance inevitably consumes enormous computing resources.

We propose a representation-based solution for making IMEs adaptive to different domains and user data by storing and updating representations while keeping the neural network close to the training domain and reducing its reliance on model training data. To implement a P2C algorithm using representations, we must first pretrain a neural network to implement the P2C algorithm on a single corpus. For this purpose, we chose the transformer model [8] as our base model. The self-attentive mechanism of this model encodes Pinyin syllables and Chinese characters from text into representations containing semantic information. A pretrained transformer model is then used to encode parallel corpora from the same domains or different ones into representations, storing Pinyin syllables, Chinese characters, and generated source and target representations in a manner suitable for data storage. By retrieving the most similar source and target representations during the P2C process, the characters corresponding to the target representations are converted into probability distributions and weighted with the transformer’s output to address domain adaptation and user behavior tracking problems. The primary contributions of this paper are as follows:

1) A P2C algorithm that uses the most similar representation to produce a model that requires no additional training and adapts to other domain data by storing and updating the representation. This approach yields performance close to that of an in-domain trained model.

2) A method for storing and updating representations that allows effective tracking of user behavior, which in turn enables timely adaptations with improvements to the user experience.

3) A representation method that carries semantic information while effectively combining historical user input information with neural network models, resulting in improved model performance.

### II. RELATED WORK

#### A. STATISTICAL MODEL-BASED PINYIN IMEs

The advent of more sophisticated statistical machine translation methods has led to Pinyin IME research based on a variety of methods. For example, Pinyin and output Chinese are treated as separate languages, and a machine translation framework is applied to effectively combine the features of source and target sequences to complete conversion [9], [10], [11]. However, Pinyin IMEs differ from machine translation methods in that IMEs must constantly acquire new knowledge from users, adapt to user habits, and reduce the number of keystrokes as much as possible. Therefore, in practice, computer-assisted functions are embedded [12] to provide more intractability to users while tracking their behaviors online [13].

#### B. NEURAL-NETWORK-BASED PINYIN IMEs

Neural-network models outperform conventional statistical models and provide additional reference methods for IME research [8], [14], [15]. Chen et al. [4] were the first to use a neural network to integrate a back-off n-gram language model into a Pinyin input method. Following the successful application of attention mechanisms [8], [14], researchers began to use neural machine translation frameworks to translate from Pinyin to Chinese [3], [5], [6]. To minimize user tapping and improve their experiences, neural-network-based IMEs must take full advantage of the contextual information of user input to predict behaviors [5] while learning new vocabularies and tracking adaptations to user input behaviors [3].

Notably, although current neural-network IMEs have achieved satisfactory results in terms of accuracy on a single domain, under the current research approach, it is difficult to adapt an offline neural-network model to new environments. Although the method proposed by Zhang et al. [3] can learn new vocabulary while tracking user behavior, it must perform

| Number of syllables | Pinyin syllable sequence conversion |
|---------------------|------------------------------------|
| 1                   | zi ran yu yan chu li               |
| 2                   | zi’ran yu’yan chu’li              |
| 6                   | zi’ran’yu’yan’chu’li              |

**TABLE 1.** Effect of syllable count on Chinese character sequences.
online training of the neural network. Our approach is more flexible in that it better adapts to user habits than existing neural-network IMEs and does not require further training. As a user continues to generate new data, the proposed approach learns their input online. By creating a datastore, our approach more efficiently utilizes historical user input information to supplement and improve model output.

C. NEURAL-NETWORK-BASED PINYIN IMEs

Domain adaptation is an ongoing topic in the use of deep learning to mitigate distributional differences between domains [16], [17], [18], [19]. In recent studies, the adaptability and performance of neural networks have been effectively improved using caching and retrieval methods. Assuming that recently appearing words might reappear in the future, many studies have used caching mechanisms to optimize neural language models [20], [21], [22]. In this approach, the cached history is used to match new inputs and change the output distribution of the neurolinguistic model, thus reducing its perplexity. Additionally, the principles of the caching mechanism have been used extensively in neural machine translation tasks [23], [24], [25], [26], [27] through the use of neural networks to encode sentences or words from training sets into representations for caching. The corresponding words are then found by retrieving translated fragments of the input sentences and weighted with the output of the neural machine translation model to improve translation quality. However, caching too many representations increases storage requirements significantly and can reduce retrieval efficiency; therefore, cached representations must be reasonably compressed while main-taining model performance, and efficient retrieval tools must be used to speed up retrieval [25], [26]. Alternatively, a small data cache can be constructed for each token of the source sentence to effectively maintain model performance and improve its decoding rate by constraining the nearest neighbor search during decoding [27].

III. PROPOSED METHOD

Our proposed adaptive Chinese Pinyin IME uses the most similar representations, which differs from conventional neural-network Pinyin IMEs in that the model output is adjusted while maintaining and utilizing data storage to achieve domain and user data adaptation without additional training.

The proposed method is divided into two phases. In the first, a pretrained transformer carries out model representation generation and storage by passing the Pinyin syllables and Chinese characters from a training set through the model to generate representations with semantic information. The Pinyin syllables, source representations, target representations, and characters are stored as key-value pairs, \((K, V)\), in which the Pinyin syllables are used as keys, \(K_i\), and the source representations, target representations, and characters form an ordered set, \(V_i\), as values. As IMEs generate new data during user interactions, it is necessary to limit the number of elements in by updating the ordered set with elements that are sequentially updated when the elements of \(V_i\) reach a specified upper limit, \(L_i\). Additionally, before storing the representations, they must be spatially compressed using product quantization [28] to reduce storage space requirements.

In the second stage, P2C is completed by retrieving the representations from the datastore. Each source Pinyin syllable sequence in the test set is encoded by the transformer model to generate corresponding parallel source representations, with corresponding target characters and representations continuously generated in an autoregressive fashion. Using the data in the datastore, the source and target representations are retrieved for similarity, and the Chinese characters corresponding to the representations are extracted and normalized into a character probability distribution that is weighted to the output of the transformer model to form a final probability distribution. To maximize the efficiency of representation retrieval, inverted indexing is used to speed up the process. Pinyin syllables and Chinese characters that have been translated and confirmed by the user are maintained and updated for user adaptivity.

A. REPRESENTATION GENERATION AND STORAGE DEFINITION

In the first stage (representation generation and storage), the Pinyin syllable lexicon is set to \(P = p_1, p_2, \ldots, p_n\), and the Chinese character lexicon is set to \(C = c_1, c_2, \ldots, c_m\). As shown in Figure 1(a), each sentence pair, \((s, t)\), comprising any sequence of \(p_i\) and \(c_j\), respectively in the training set, generates the corresponding representation, \((h, z)\), after passing through the pretrained transformer model. We define the transformer model encoder and decoder as functions \(f_{enc}(\cdot)\) and \(f_{dec}(\cdot, \cdot)\), which are calculated in the following forms:

\[
h = f_{enc}(s),
\]
\[
z = f_{dec}(h, t).
\]

In Figure 1(b), we describe the representation information corresponding to the same pinyin syllable from different contexts. Although the same pinyin syllable corresponds to many different Chinese characters, in different contexts, the source representation and target representation contained in the same pinyin syllable are different in similarity. Therefore, we can use the difference information to confirm the Chinese characters after Pinyin syllable conversion.

Furthermore, we define the key value, \((K, V)\), for the sentence pairs and representations, \((s, t)\) and \((h, z)\), respectively, to be stored sequentially in the datastore. Using the syllable, \(p_i\), in the source Pinyin, \(s\), as the key, \(K_i\), the corresponding ordered set, \(V_i\), is obtained from the datastore, \((K, V)\), through key value mapping. The source representation, \(h_i\), corresponding to the syllable, \(p_i\), and the target representation, \(z_i\), and target character, \(c_i\), are stored in the corresponding ordered set, \(V_i\), at the \(j\)th location, where \(j\) is the capacity of the \(V_i\) modulo the number of elements stored. The data are
stored in the following form:
\[
(K, \mathcal{V}) = \{p_i, \{h_{ij}, z_{ij}, c_{ij}\}\}. \tag{3}
\]

**B. REPRESENTATION QUANTIFICATION AND STORAGE**

Product quantization can be used as a coding method to effectively reduce storage overhead. The proposed method implements quantization by first dividing each input vector, \( \mathbf{u} \in \mathbb{R}^D \), into \( m \) equally sized segments of tandem subvectors, each having the same dimension, \( D^r = D/m \):
\[
\mathbf{u} = [\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^m], \quad \mathbf{u}^i \in \mathbb{R}^{D^r}. \tag{4}
\]

Using a product quantizer, \( q \), with \( m \) sub-quantizers, \( q_1, q_2, \ldots, q_m \), the vector, \( \mathbf{u} \), is then quantified with the quantization objective:
\[
\min_{q_1, q_2, \ldots, q_m} \sum_{\mathbf{u}} \sum_i ||\mathbf{u}^i - q_i(\mathbf{u}^i)||^2, \tag{5}
\]
where \( q_i \) is the low-complexity quantizer of the \( i \)-th subvector, which relates the index set, \( I^i \), to the codebook, \( O^i \). The Cartesian product of the index set, \( I \), and the codebook, \( O \), is defined as
\[
I = I^1 \times \ldots \times I^m, \tag{6}
\]
\[
O = O^1 \times \ldots \times O^m. \tag{7}
\]

Each \( \mathbf{u} \) is mapped to the nearest Cartesian product, \( O \), and identified by the index set, \( I \). Assuming that each codebook, \( O^i \), comprises a finite number of codewords, \( w \), the \( w^m \) \( D \)-dimensional codewords within the Cartesian product space, \( O \), can be represented by vectors with dimensions of only \( w \times m D^r \), which effectively reduces the storage space requirement. We experimentally determined that quantizing each representation effectively reduced the storage space from 2,048 to 128 B, a reduction to 1/16th of the original size.

Although the quantization of representations can effectively reduce storage space, storing representations without limits will also consume a significant amount of space and may even increase retrieval time during subsequent retrieval. Therefore, the representation storage process must limit the capacity of the ordered set, \( \mathcal{V}_i \), corresponding to syllable \( i \) to \( L_i \). Because each syllable has a different probability of occurring in daily use, the 400+ syllables in common use can be divided into high- and low-frequencies, and the capacity, \( \mathcal{V}_i \), corresponding to the ordered set, \( \mathcal{L}_i \), for syllable \( i \) can be specified separately. The choice of capacity, \( \mathcal{L}_i \), will be elaborated in subsequent sections.

**C. REPRESENTATION-BASED P2C TASK**

Figure 2 shows the second stage, in which the datastore is used to complete the P2C task. For a given test case source Pinyin syllable sequence, \( \mathbf{x} = (x_1, \ldots, x_l) \), producing a generated target character sequence, \( \mathbf{y}(1_{r-1}) \), the transformer model generates the following representation:
\[
\mathbf{h}_r = f_{\text{enc}}(\mathbf{x}), \tag{8}
\]
\[
\mathbf{z}_r = f_{\text{dec}}(\mathbf{h}_r, \mathbf{y}(1_{r-1})). \tag{10}
\]
where \( \mathbf{h}_r \) is the source representation sequence for the test case, \( \mathbf{h}_r \) is the representation corresponding to the \( r \)-th syllable of the Pinyin syllable sequence, \( \mathbf{x} \). \( \mathbf{z}_r \) is the target representation generated by the model at step \( r \), and \( p(\mathbf{y}_r) \) is the character probability distribution for the target sequence output at step \( p(\mathbf{y}_r) \).

The ordered set, \( \mathcal{V}_i \), corresponding to syllable \( i \) is first determined by the key mapping, \( p_i = x_r \). Because the transformer can compute in parallel, the sequence of representations, \( \mathbf{h} \), generated on the encoding side can be computed simultaneously with \( \mathbf{h}_{ij} \). Here, we choose cosine similarity as the retrieval metric and determine the similarity between the \( r \)-th source representation, \( \mathbf{h}_r \), in the sequence, \( \mathbf{h} \), and \( \mathbf{h}_{ij} \) in the ordered ensemble, \( \mathcal{V}_i \),

\[\text{FIGURE 1. Representation generation process of sentence pairs and the internal details of different representations of the same syllables.}\]
The first $k_1$ source representations, $\hat{h}_r$, are most similar to $\{h_{11}, h_{12}, \ldots, h_{1k_1}\}$ and their corresponding target representations, $\{z'_{11}, z'_{12}, \ldots, z'_{1k_1}\}$, and the target character, $c'_{11}, c'_{12}, \ldots, c'_{1k_1}$. On the decoding side, the cosine similarity of the target representation, $\hat{z}_r$ to $z'_{ij}$ is then determined, and the $k_2$ target representations with the highest similarity scores, $\{z''_{11}, z''_{12}, \ldots, z''_{1k_2}\}$, and their corresponding target characters, $\{c''_{11}, c''_{12}, \ldots, c''_{1k_2}\}$, are defined:

$$\text{score}(z'_{ij}, c'_{ij}) = \text{top}_k(\text{sim}(\hat{h}_r, h_{ij})),$$
(11)

$$\text{score}(c''_{ij}) = \text{top}_k(\text{sim}((\hat{z}_r, z''_{ij})).$$
(12)

The $\text{score}(c''_{ij})$ is then normalized to a probability using Softmax:

$$\text{Softmax}(\text{score}(c''_{ij})) = \frac{e^{\text{score}(c''_{ij})}}{\sum_j e^{\text{score}(c''_{ij})}}.$$  
(13)

To make the probability distribution smoother, the $\text{score}(c''_{ij})$ is divided by the temperature, $T$, prior to normalization using Softmax. Because many identical Chinese characters appear during the retrieval process, the same target character probabilities are aggregated to form a new target character probability distribution, $p_{\text{MSR}}(y_r)$:

$$p(c''_{ij}) = \text{Softmax}(\text{score}(c''_{ij})/T),$$
(14)

$$p_{\text{MSR}}(y_r) = \sum_{j=1}^{k_1} \mathbb{1}_{y_r = c''_{ij}} p(c''_{ij}).$$
(15)

Although a P2C obtained purely through the retrieval of representations is valid, to make our approach more robust, the output, $p_{\text{Trm}}(y_r)$, of the transformer model is weighted and averaged into $p_{\text{MSR}}(y_r)$, which improves results and gives a final probability distribution of the P2C algorithm based on the most similar representation vector:

$$p(y_r) = (1 - \lambda)p_{\text{Trm}}(y_r) + \lambda p_{\text{MSR}}(y_r).$$
(16)

Following the standard neural translation model process, Pinyin syllable sequences are decoded using a beam search to produce candidate Chinese character sequences for the user in a real environment. Although, in practice, the user may not find an exact match among the candidates, the generated characters can be easily modified via human-computer interaction. Finally, representations are generated from the user-input Pinyin syllable sequences and user-confirmed Chinese character sequences through the transformer model, and the Pinyin syllable, Chinese characters, and representations are updated to the datastore using the flow of Algorithm 1.

**D. RETRIEVAL DETAILS**

Although we impose a limit on the capacity of the ordered set, $(K, V)$, in the datastore, $V_i$, finding an exact match for the source representation of Equation 11 can be very time-consuming when there are many elements in $V_i$. Therefore, Faiss [29] is used to conduct fast similarity searching of representations. Recall that in the process described above, syllables are divided into low- and high-frequency samples. For the ordered set, $V_i$, of low-frequency syllables, a brute-force search is performed to obtain the $k_1$ most similar source representations. For the ordered set, $V_i$, of high-frequency syllables, an inverted index is used with the size of the cluster.

---

**FIGURE 2.** Illustration of P2C algorithm based on the most similar representation.
**Data Preparation:** Datasetore \((K, V)\), the capacity \(\mathcal{L}\) of the Ordered set \(V\) and its storage cursor \(\mathcal{J}\).

**the Real Environment:**
1. user input Pinyin syllable sequence \(x\)
2. the Pinyin syllable sequence \(x\) generates source representations \(\hat{h}\) via IMEs
3. IMEs output candidates \(Y\)
4. user compares the true sequence \(y\) with the candidate \(Y\)
5. if \(y \neq Y\) then
6. user modifies a candidate in \(Y\) to \(y\)
7. end if
8. IMEs generate target representations \(\hat{h}\) through \(y\) and \(\hat{z}\)
9. for all \((x_r, \hat{h}_r, \hat{z}_r, y_r) \in (x, \hat{h}, \hat{z}, y)\) do
10. \(p_i \leftarrow x_r; \ell \leftarrow \mathcal{L}(p_i); j \leftarrow \mathcal{J}(p_i)\)
11. \(f \leftarrow \text{mod}(j + 1, \ell)\)
12. obtain the ordered set \((K, V)\) from datastore \(V_i\) according to the syllable \(p_i\)
13. read the \(j\)th cell \(v_{ij}\) from the ordered set \(V_i\)
14. \(v_{ij} \leftarrow (\hat{h}_r, \hat{z}_r, \hat{y}_r)\)
15. end for

set to \(N_e\) according to its capacity, \(\mathcal{L}_i\):

\[
N_e = \min(4 \times \sqrt{\mathcal{L}_i}, \mathcal{L}_i/30).
\]  
(17)

The source representations are \(k\)-means clustered and stored in the trained clusters. During the retrieval process, the most similar \(k_1\) source representations are queried by retrieving the 32 nearest cluster hearts. The target representations and characters corresponding to the \(k_1\) retrieved source representations are used in Equation 12. Because the number of target representations is already limited in finding the similarities of the source representations, the \(k_3\) most similar target representations are selected by finding the similarities between the target representations of the test case and the \(k_1\) currently retrieved target representations, and their target characters are used in subsequent calculations.

**IV. EXPERIMENT**

In this study, we evaluated the proposed method by assessing it on a corpus of four stylistic domains. It is worth noting that surpassing state-of-the-art accuracy was not our aim. Although the use of representations enhances model adaptability, without additional training, it does not outperform a model trained directly in the domain. Our goals were instead to use an already trained neural network to maintain high performance across different environments and to track user behavior in conjunction with user history information to improve user input efficiency and experience.

**A. PARALLEL CORPUS**

A parallel corpus from four different domains was used for the evaluation: **People’s Daily** (a corpus of news extracted from the People’s Daily at Peking University from 1992 to 1998 [30]); **Touchpal** (a corpus of user chats collected by Touchpal IME [13]); **CAIL2019** (a legal reading comprehension corpus of the 2019 “China Law Research Cup” Judicial Artificial Intelligence Challenge \(^1\)); and **cMedQA 1.0** (a medical text corpus for Chinese community medical question answering [31]). Each corpus varied in topic and style, allowing us to verify the adaptability of our approach to different environments. We split each corpus into a maximum input unit (MIU) [13], defined as the longest continuous sequence of Chinese characters in a sentence segmented by non-Chinese parts. Each MIU was then converted into the corresponding Pinyin syllable sequences using the Pinyin toolkit\(^2\) to form a parallel corpus. From the four corpora a test set of 2,000 MIUs was extracted and the remaining MIUs were used as the training set. Table 2 lists the statistics for the four corpora.

**B. METRICS**

The IMEs were evaluated in terms of three metrics: MIU accuracy (MIU-Acc), Chinese character accuracy (CA) [4], and keystroke score (KySS) [32]. Each IME output a corresponding ranked list of candidate Chinese character sequences, and MIU-Acc was used to calculate scores based on whether the first \(k\) sequences in the list exactly matched the real sequence:

\[
\text{MIU-Acc} = \frac{N_k}{N_m} \times 100\%.
\]  
(18)

where \(N_k\) is the number of times the IME was correctly predicted from the first \(k\) candidates, and \(N_m\) is the total number of MIUs.

CA, a common evaluation metric for Chinese IMEs used to measure the Chinese character accuracy of the first, is calculated as follows:

\[
\text{CA} = \frac{N_c}{N_t} \times 100\%.
\]  
(19)

where \(N_c\) denotes the number of Chinese characters correctly predicted, and \(N_t\) denotes the total number of Chinese characters.

Chinese IMEs provide users with sequences of five candidate characters per page by default, and if users can obtain output directly from the five sequences provided by an IME, the IME is preferred. KySS quantifies user experience in terms of the number of keystrokes:

\[
\text{KySS} = \frac{N_k}{N_a} \times 100\%.
\]  
(20)

where \(N_k\) is the number of keystrokes performed by the user on the next page and candidate keys, apart from the input letter keys. \(N_a\) represents the actual number of keystrokes apart from the input letter keys. In an ideal IME, to enter any MIU Pinyin syllable sequence, the user only needs to

\(^1\)The 2019 “China Law Research Cup” Judicial Artificial Intelligence Challenge: https://github.com/china-ai-law-challenge/CAIL2019

\(^2\)https://github.com/mozillazg/python-pinyin
hit a candidate key on the first page provided by the IME to obtain the required character sequence, in which case KySS equals one.

C. EXPERIMENTAL SETTINGS

IMEs provide users with lists of Chinese character candidates based on the Pinyin syllable sequence entered. Therefore, measuring the performance of an input method is equivalent to evaluating the lists used. In this task, we referred to the typical IME settings and allocated five candidate Chinese character sequences to each page, which allows users to turn pages to find additional candidate Chinese character sequences. The sequences of Chinese characters were determined using a beam-search algorithm.

To pretrain the transformer model, we followed the design of Vaswani et al. [8] and performed pretraining on the People’s Daily training set to use as our base model, Base(PD). Base(PD) was then used to extract representations of the training set from the four corpora and store the representations, Pinyin syllables, and Chinese characters as key-value pairs to construct the datastore, \( \mathcal{K}, \mathcal{V} \). Before constructing the datastore, \( \mathcal{K}, \mathcal{V} \), we counted the occurrences of syllables on the People’s Daily training set. The syllables appearing more than 30K times were identified as high frequency, and those appearing fewer times were identified as low frequency. The capacity, \( \mathcal{V}_i \), of the ordered set, \( \mathcal{L}_i \), corresponding to the high-frequency syllable, \( i \), was then set to 200K, and the capacity, \( \mathcal{V}_i \), of the ordered set, \( \mathcal{L}_i \), corresponding to the low-frequency syllable, \( i \), was set to 30K. The representations were quantified using Faiss and stored at 128 B.

In the representation-based P2C task, we used cosine similarity to measure similarity based on Equations 11 and 12 with the hyperparameters set to \( k_1 = 1,024 \) and \( k_2 = 16 \), respectively. In the normalization and weighted average process, we set the temperature to \( T = 0.05 \) to slow the overfitting of the distribution and set the weight parameter, \( \lambda \), to 0.8.

On hardware setup, we used GPU GeForce RTX 2080 Ti and CPU Intel(R) Xeon(R) Gold 5220 CPU @ 2.20GHz to execute our method.

D. BASELINE COMPARISON

Previous studies have focused on improving IME conversion using trained neural networks [4], [5], [6] while ignoring domain adaptation. Thus, conventional approaches do not consider the ability of an IME to learn by interacting with users online; therefore, they cannot be adequately applied to practical scenarios. By contrast, our approach exploits the efficient transformation capability of neural networks to track user behavior online and uses historical user input to further improve the output. To validate our approach, we implemented the following baselines for comparison:

- **Base(PD)** is a pretrained transformer model that uses the People’s Daily training set. Model training followed the design of Vaswani et al. [8].
- **On-OMWA** is a conventional adaptive algorithm proposed by Zhang et al. [13], which adjusts word likelihood or generates new words by comparing the user’s real input choices to algorithm predictions.
- **Google IME** is a commercial input method that enhances user experience through various components, including optimized language models, high-quality vocabularies, and a large corpus.\(^3\)

We also applied the transformer model to train and test four datasets (i.e., People’s Daily, Touchpal, cMedQA 1.0, and CAIL2019) to obtain the in-domain effect of the model. Although our method is not effective in improving in-domain performance, its adaptability to different domains is further demonstrated.

E. RESULTS

Table 3 lists the results of our experiments on the four corpora. As we have seen, when the transformer model is trained and tested directly on the domain, it produces a high MIU-Acc score, but using Base(PD) for testing on datasets from other domains results in a far less effective transformation. Our approach augments the adaptability of the model by storing and retrieving the most similar representations from atop Base(PD). Within the domain (i.e., People’s Daily corpus), our approach was slightly more effective than testing directly within the domain as storing and retrieving representations effectively combine historical information to enhance output. When performing outside the domain, the proposed approach did not outperform the model tested directly in the domain, but it still achieved significantly improved model adaptation. Compared with using Base(PD) directly, our Top-1 scores on Touchpal, cMedQA 1.0, and CAIL2019 corpora of MIU-Acc were improved by at least 20.0% and were all close to those of the model tested directly in the domain without

---

\(^3\)Google IME is the only commercial Chinese IME on the market that provides a debuggable application programming interface.
TABLE 3. Model performance comparison (%).

|                  | People's Daily | Touchpal | CAIL2019 | cMedQA 1.0 |
|------------------|---------------|----------|----------|------------|
|                  | Top-1 | Top-5 | Top-1 | Top-5 | Top-1 | Top-5 | Top-1 | Top-5 |
| In-domain       | 90.3 | 97.1 | 70.2 | 88.2 | 87.1 | 95.1 | 92.8 | 98.3 |
| Base(PD)        | 90.3 | 97.1 | 42.5 | 66.3 | 49.5 | 66.7 | 55.9 | 71.2 |
| On-OMWA         | 64.4 | - | 57.1 | - | 71.4 | - | 79.6 | - |
| Google IME      | 82.7 | - | 61.7 | - | 66.0 | - | 85.4 | - |
| Our Method      | 90.6 | 97.8 | 64.9 | 82.5 | 82.4 | 92.5 | 90.5 | 95.7 |

Note: Apart from the direct test results in the domain, the optimal results are shown in bold.

additional training. Compared with the adaptive method, On-OMWA by Zhang et al. [13], our method achieved Top-1 performance on the People’s Daily, Touchpal, CAIL2019, and cMedQA 1.0 corpora; MIU-Acc was better than that of the On-OMWA method by 26.2%, 7.8%, 11.0%, and 10.9%, respectively. To further investigate its competitiveness, we compared our model with Google IME. Our method outperformed it on different datasets and achieved a Top-1 MIU-Acc on CAIL2019, 16.4% higher than that of Google IME.

FIGURE 3. Top-1 MIU-Acc(%) comparison between our method and the other four methods. "ns" means p-value > 0.05, "*" means 0.01 < p-value <= 0.05, "**" means 0.001 < p-value <= 0.01, "***" means p-value <= 0.001.
TABLE 4. Qualitative table.

| Pinyin | Chinese characters | Representation similarity | Probability distribution |
|--------|-------------------|--------------------------|-------------------------|
| er qie ji fa xing dian xian de zhi xi qin shu dian xian fa bing lv ye yu an bi pu tong ren qun gao | 而且继发性 癫痫 的直系亲属癫痫发病 | 0.8824 | 0.0630 |
| sui ran chou dong shi dian xian de zhu yao zheng zhuang zhi yi | 虽然抽动是 癫痫 的主要症状之一 | 0.8776 | 0.0627 |
| yuan fa xing dian xian ji you yi ding yi chuan xing | 原发性 癫痫 就有一定 遗传性 | 0.8750 | 0.0626 |

The final retrieved character probability distribution: p(癫痫) = 1.0
Base(PD) generates the result: 目前至 于电 线仍以 药物控制法 作为 主
The final result generated by our method: 目前 难治性 癫痫 仍 以药物控制法 作为 主

Note: Input examples were taken from the cMedQA 1.0 test set. The transformation process was performed on each example by retrieving the most similar representation, resulting in the cor-rect retrieval of the target character, “癲” (marked in red), as well as the semantic relevance of the local context of “癲.” The similarity of the representation was calculated on the decoder side of the model and normalized to obtain the probability distribution of “癲”.

F. INSTANCE ANALYSIS

We then performed a qualitative analysis of the similarity of P2C algorithm representations. The representations on the cMedQA 1.0 training set were extracted using the Base(PD) model, and Pinyin syllable and corresponding representations and Chinese characters were stored as key-value pairs. The examples were then decoded based on a beam size of one. As seen in Table 4, the characters retrieved by the most similar representations were all mapped to “癲,” and the retrieved sentences presented high representational similarities to the input sentences on the decoding side as well as strong local correlations between retrieved content (i.e., all retrieved sentences contained “癲”). This indicates that the semantic information carried by the representations resolves ambiguities in the Pinyin syllables mapping of Chinese characters. Storing sentence representations not only allows domain adaptation but also makes full use of the associated information to assist user input.

The representations of the input examples are further analyzed in Table 4. Figure 3 shows the cosine similarity of Chinese characters corresponding to the syllable, “xian,” in the input example. We found that, although there are more than 10 Chinese characters corresponding to “xian” in the cMedQA 1.0 corpus, we distinguished the real meaning of the syllable by using the spatial semantic information of the representation. Therefore, at higher cosine similarity of the representation, we can speculate that the Chinese character corresponding to the syllable “xian” may be “癲”. Simultaneously, in the input pinyin syllable sequence, there is a certain correlation between the representations of the character “癲” corresponding to the syllable “xian” and the character “癲” corresponding to “dian”, further indicating that the representation carries the local semantic information of sentences.

divided into 10 groups with 200 test samples each. Pairwise comparisons between our method and In-domain, Base(PD), On-OMWA, Google IME were performed by t-test to determine check for any difference in the Top-1 MIU-Acc metric. In Figure 3(a), our method is not significantly different from the In-domain model in People’s Daily (p-value = 0.876) and CAIL2019 (p-value = 0.059). In Touchpal (p-value < 0.001) and cMedQA 1.0 (p-value = 0.030), our method underperforms the In-domain model, indicating that our method is close to the In-domain model in performance. In Figure 3(b), our method has a significant performance improvement over Base(PD) in Touchpal (p-value<0.001), CAIL2019 (p-value < 0.001) and cMedQA 1.0 (p-value < 0.001), there is no significant difference in People’s Daily (p-value = 0.876), which indicates that our method can effectively improve the domain adaptability of neural networks. In Figure 3(c), all the four groups show strong inter-group differences (all p-values < 0.001), and it can be seen from the inter-group comparison results that our method is stronger than the traditional adaptive method On-OMWA. In Figure 3(d), People’s Daily (p-value = 0.005), CAIL2019 (p-value < 0.001), and cMedQA 1.0 (p-value = 0.003) groups showed strong significant differences, and in Touchpal (p-value = 0.003) value = 0.085) does not show significant difference, but our method is still stronger than Google IME.

VOLUME 10, 2022
G. WEIGHTING PARAMETER EFFECT

We adjusted parameter $\lambda$ in Equation 16 to enhance model robustness by weighting the average of the base model and the retrieval distribution. Figure 5 shows the performance of the proposed method on the four corpora for different values of $\lambda$. On the in-domain data (People’s Daily), changing $\lambda$ did not reduce Base(PD) performance; instead, it was slightly enhanced with the increase of $\lambda$. By contrast, $\lambda$ had a significant impact on Base(PD) performance on out-of-domain data (i.e., Touchpal, cMedQA 1.0, and CAIL2019). When $\lambda = 0.8$, the performance of Top-1 MIU-Acc on the dataset Touchpal, CAIL2019, cMedQA 1.0 is improved by 22.4%, 32.9%, 34.6%, for $\lambda$ greater than 0.8, model performance gradually leveled. Accordingly, $\lambda = 0.8$ is a good setting as approaching a value of one does not significantly improve model performance; it is not useful to rely completely on retrieved characters when the number of representations within the data store is small.
syllables with more- and less-frequent occurrences than the low-frequency syllable capacity were categorized as high- and low-frequency, respectively. Using this approach, the storage capacity was partitioned between high- and low-frequency syllables. It is seen from Table 5 that a strong correlation appeared between the model transformation quality and the number of stored representations. Although setting the ordered set, \( V_i \), of syllable \( i \) to high-capacity storage improved performance, using high-capacity ordered sets increased storage requirements and computational costs of queries. Nevertheless, we were surprised to find that, with significant reductions in storage capacity, \( L_i \) did not effectively reduce model running speed, leading us to speculate that Faiss might not apply to the representation updating process, and that instead, frequent updates to the representations may be the primary factors affecting performance. When the capacity \( L_i \) of both high-frequency and low-frequency syllables is set to 0, the model no longer uses the representation retrieval method and degenerates into a commonly used neural network model. The model conversion speed is increased four-
TABLE 6. User experience with CA and KySS.

|                  | People’s Daily | Touchpal | CAIL2019 | cMedQA 1.0 |
|------------------|----------------|----------|----------|------------|
|                  | CA             | Ky$$S$$  | CA       | Ky$$S$$    | CA         | Ky$$S$$ |
| In-domain        | 0.985          | 0.975    | 0.942    | 0.898      | 0.978      | 0.957   | 0.988 | 0.986 |
| Base(PD)         | 0.985          | 0.975    | 0.847    | 0.659      | 0.866      | 0.694   | 0.863 | 0.734 |
| On-OMWA          | 0.932          | 0.712    | 0.908    | 0.723      | 0.945      | 0.782   | 0.955 | 0.813 |
| Google IME       | 0.961          | 0.879    | 0.919    | 0.703      | 0.914      | 0.757   | 0.959 | 0.902 |
| Our Method       | 0.985          | 0.983    | 0.928    | 0.842      | 0.968      | 0.934   | 0.981 | 0.961 |

Note: Apart from those used for the direct tests in the domain, the best results are shown in bold.

fold, but it’s conversion performance is reduced to minimum. This suggests that our method requires additional storage space, and is thrice more computationally expensive than directly using neural networks, and it would be worthwhile to investigate $L_i$ using the ordered set, $\mathcal{V}_i$, of each syllable, $i$.

J. EFFECT OF ONLINE UPDATED REPRESENTATION
To demonstrate that our method can automatically adapt to domain data, we initialized the ordered set, $\mathcal{V}$, to empty and updated the ordered set, $\mathcal{V}$, online so that it would adapt to changes in data. We then compared the results to those obtained with the conventional adaptive model, ON-OMWA, and the offline model, Base(PD).

As a test corpus for comparison, we extracted 600K MIUs from the cMedQA 1.0 corpus and divided them equally into 300 groups using different methods to measure the highest accuracy score for each. This was feasible because Base(PD) was trained on People’s Daily, and the significant domain difference between cMedQA 1.0 and People’s Daily allowed us to effectively test the adaptability of our method. It is seen from Figure 7 that the adaptability of Base(PD) was effectively improved by updating the representations online, and the proposed method had better adaptability and performance than On-OMWA.

Figure 8 further demonstrates the adaptability of our approach on a joint test corpus obtained by combining 200K MIU extractions from each of the TouchPal, cMedQA 1.0, and CAIL2019 corpora and dividing the results equally into 300 groups on six sections. The highest MIU accuracy for each group was then recorded, with the vertical lines in the figure indicating the joint points between the respective corpora. In the first three parts of the test corpus, our method quickly adapted to changes after learning some sentences, with the degree of adaptation increasing with accumulated learning. In the latter three sections, the ordered set, $\mathcal{V}$, improved performance based on the previous learning passes as it already contained the previously learned domain data distribution. Moreover, the curve coverage region of our method always matched that of the conventional adaptive method, On-OMWA. These results further demonstrate that our method was consistently superior to the conventional adaptive model in terms of adaptability and performance.

K. USER EXPERIENCE
We also measured user experience using the method proposed by Chen [4] and Jia [32] et al. Measurement was performed using Equations 19 and 20, from which a high CA indicator predicts that users have a high probability of obtaining the Chinese character needed from possibly the first set of IME candidates. A high-quality Ky$$S$$ indicates that users can directly obtain a selected sequence on the first page of the input without the need for further page flipping. It is seen from the results in Table 6, which compares CA and Ky$$S$$ performance, that our method is adaptable to various domain data and provides a good user experience.

V. CONCLUSION
In this study, a new neural-network model-based adaptive P2C method for use in Chinese Pinyin IMEs was proposed. The method supplements the conventional model with a representation based on dynamic storage and retrieval to enhance performance and adaptability. The representation can be updated according to user input without further network training. The resulting IME was experimentally shown to be capable of tracking user behavior with high domain adaptability, and it outperformed conventional Pinyin conversion frameworks and commercial IMEs according to many indicators. The Chinese Pinyin input method is the current mainstream tool for using 26-character Latin keyboards to create Chinese characters. Hence, higher conversion accuracy and stronger versatility is desired by users. Our method greatly improves user experience, enhances user input efficiency, and enables the improvement and user friendliness of P2C platforms. Simultaneously, we encourage IMEs in other languages in the world to use the similarity of representations to enhance translation performance and adaptability and improve user experience. However, this method requires additional storage space and takes longer to decode. Thus, it is necessary to build a smaller, more responsive ordered set and adopt a faster retrieval method in a future work to further enhance its practical use.

REFERENCES
[1] Z. Zhang, Y. Huang, and H. Zhao, “Subword-augmented embedding for cloze reading comprehension,” in Proc. 27th Int. Conf. Comput. Linguistics, 2018, pp. 1902–1814.
[2] Z. Jia and H. Zhao, “A joint graph model for Pinyin-to-Chinese conversion with typo correction,” in Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Baltimore, MD, USA, vol. 1, 2014, pp. 1512–1523, doi: 10.3115/v1/P14-1142.

[3] Z. Zhang, Y. Huang, and H. Zhao, “Open vocabulary learning for neural Chinese Pinyin IME,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 1584–1594, doi: 10.18653/v1/P19-1154.

[4] S. Chen, H. Zhao, and R. Wang, “Neural network language model for Chinese Pinyin input method engine,” in Proc. 29th Pacific Asia Conf. Lang., Inf. Comput., Shanghai, China, 2015, pp. 455–461. [Online]. Available: https://aclanthology.org/Y15-1052.

[5] Y. Huang and H. Zhao, “Chinese Pinyin aided IME, input what you have not keystroked yet,” in Proc. Conf. Empirical Methods Naturl Lang. Process., Brussels, Belgium, Oct. 2018, pp. 2923–2929, doi: 10.18653/v1/D18-1321.

[6] Y. Huang, Z. Li, Z. Zhang, and H. Zhao, “Moon IME: Neural-based Chinese Pinyin aided input method with customizable assistance,” in Proc. ACL, Syst. Demonstrations, Melbourne, VIC, Australia, 2018, pp. 140–145, doi: 10.18653/v1/P18-4024.

[7] X. Zhang, Y. Guan, X. L. Wang, and B. Liu, “Pinyin to character conversion model based on support vector machines,” J. Chin. Inf. Process., vol. 21, no. 2, pp. 100–105, 2007.

[8] S. Yang, H. Zhao, and B. L. Lu, “A machine translation approach for Chinese whole-sentence Pinyin-to-character conversion,” in Proc. 26th Pacific Asia Conf. Lang., Inf. Comput., 2012, pp. 333–342.

[9] S. Zha, “Optimized Chinese pronunciation prediction by component-based statistical machine translation,” J. Inf. Process. Syst., vol. 17, no. 1, pp. 203–212, 2021, doi: 10.3745/JIPS.04.0208.

[10] G. Huang, J. Zhang, Y. Zhou, and C. Zong, “A new input method for human translators: Integrating machine translation effectively and imperceptibly,” in Proc. 24th Int. Joint Conf. Artif. Intell. Jun. 2015, pp. 1–7. [Online]. Available: https://www.aaai.org/ojs/index.php/ICAII15/paper/view/10092.

[11] X. Zhang, C. Wei, and H. Zhao, “Tracing a loose wordhood for Chinese input method engine,” 2017, arXiv:1712.04158.

[12] T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in Proc. Conf. Empirical Methods Natural Lang. Process., Lisbon, Portugal, 2015, pp. 1412–1421, doi: 10.18653/v1/D15-1166.

[13] X. Yi, M. Sun, R. Li, and Z. Yang, “Chinese poetry generation with a working memory model,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 4553–4559, doi: 10.24963/ijcai.2018/633.

[14] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and A. Smith, “Don’t stop pretraining: Adapt language models to domains and tasks,” in Proc. 58th Annu. Meetinng Assoc. Comput. Linguistics, 2020, pp. 8342–8360, doi: 10.18653/v1/2020.acl-main.740.

[15] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, “Maximum density divergence for domain adaptation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 11, pp. 3918–3930, Nov. 2021, doi: 10.1109/TPAMI.2020.2991050.

[16] J. Li, M. Jing, H. Su, K. Lu, L. Zhu, and H. T. Shen, “Faster domain adaptation networks,” IEEE Trans. Knowl. Data Eng., early access, Feb. 19, 2021, doi: 10.1109/TKDE.2021.306473.

[17] Z. Li, X. Peng, M. Zhang, R. Wang, and L. Si, “Semi-supervised domain adaptation for dependency parsing,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 2386–2395, doi: 10.18653/v1/P19-1229.

[18] D. Sontner, Z. Loose, L. Müller, and A. Prazák, “Neural network language model with cache,” in Proc. Int. Conf. Text, Speech Dialogue, Berlin, Germany, 2012, pp. 528–534.

[19] E. Grave, A. Joulin, and N. Usunier, “Improving neural language models with a continuous cache,” 2016, arXiv:1612.04426.