Rethinking Input Methods As Back-Transliteration and the Quest for Linguistic Equity

Anonymous ACL submission

Abstract

Back-transliteration serves a dual purpose: processing informal Romanized text and powering phonetic Input Methods (IMEs). This survey bridges the fragmented research landscape by introducing a "maturity spectrum" that contrasts the sophisticated engines of CJK languages with the foundational models of Arabic and Indic scripts. We systematically analyze the convergent technological evolution—from statistical models to Large Language Models (LLMs). We demonstrate that despite the rise of LLMs, specialized back-transliteration remains essential for latency-sensitive input. The paper concludes with a technical roadmap for building the next generation of universal input systems.

1 Introduction

Text input is a foundational layer of digital interaction, yet for users of non-Latin scripts, the standard QWERTY keyboard presents a persistent barrier. A widely adopted solution is phonetic input, where users type the sounds of their language using Latin letters, and a system converts this input into native characters—a process termed back-transliteration. This computational task powers two critical applications: enabling real-time, interactive input methods for languages such as Chinese and Japanese, and processing the vast volume of informally Romanized text—like Arabic Arabizi or Romanized Indian languages—that pervades digital communication.

Despite this shared core, research is siloed and marked by stark inequity. For a handful of high-resource languages, back-transliteration has evolved into mature IME research, featuring sophisticated disambiguation, personalization, and integration with large language models. In contrast, for most of the world's languages—including many in the Arabic, Indic, and indigenous families—work remains foundational, focusing on converting noisy, non-standard input amid severe data constraints. This technological gap reinforces digital marginalization, limiting both individual participation and the online vitality of low-resource languages.

This paper bridges these divides by proposing a unified framework for back-transliteration-based input methods. We analyze the field through two complementary lenses: a convergent technological evolution that traces a shared journey from rule-based and statistical models to neural and generative paradigms; and a technological maturity spectrum that contrasts the state of the art in high-resource languages with the foundational needs of low-resource ones. Systematically, we examine core challenges—ambiguity resolution, adaptation under resource constraints, interaction optimization, and system efficiency—and present cross-linguistic case studies spanning Arabic, Indic, Japanese, and Chinese systems. Finally, we assess the dual role of LLMs in this domain and outline a concrete, equity-focused research agenda aimed at democratizing intelligent text input and fostering a linguistically inclusive digital future.

2 The Unified Framework and Core Challenges

2.1 Formalization

The typical workflow of a back-transliteration-based input method is as follows: A user types phonetic input using a standard keyboard. The input method engine processes this sequence through multiple stages: mapping the phonetic string to candidate native script units, generating initial candidates, leveraging context for disambiguation, ranking the candidates, and finally outputting the most appropriate native script sequence. While real-world systems often incorporate additional functionalities such as error tolerance, personalization, and multimodal interaction, the most fundamental and universal computational task remains the

conversion from phonetic representation to native script.

This core conversion task can be formalized as a supervised sequence-to-sequence transformation. Given a romanized sequence composed of Latin characters, $\mathbf{p} = (p_1, p_2, ..., p_m)$, the goal is to find the most probable target native script sequence $\mathbf{c} = (c_1, c_2, ..., c_n)$. This process can be modeled as:

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}} P(\mathbf{c}|\mathbf{p}) \quad (1)$$

where the probability $P(\mathbf{c}|\mathbf{p})$ is estimated by a system incorporating both language and transliteration models. This definition distinguishes it from general machine translation (natural language to natural language) and pure transliteration (character-to-character), emphasizing that its input is systematic phonetic encoding and its output comprises semantic-laden textual units.

2.2 Three Tasks Back-transliteration-based IME

The research landscape of back-transliteration-based input methods is structured around three interrelated core task families. The Core Conversion Task constitutes the fundamental mapping of standard romanized sequences to native characters, forming the foundation of all systems. However, real-world input is fraught with uncertainty, giving rise to the Error-Tolerant Input Task, which handles spelling mistakes, dialectal variations, and informal abbreviations, requiring strong noise robustness. Finally, the Personalization Task focuses on dynamically adapting the conversion model using historical user behavior data to align with individual linguistic habits and specialized domains, thereby reducing cognitive load and enhancing input efficiency. These tasks face a series of common, fundamental challenges across different script systems. We will explain in detail in the next section.

3 Core Technical Challenges and Methodological Evolution

Despite being developed for distinct writing systems, solutions for back-transliteration-based input methods have followed a remarkably convergent evolutionary path, collectively undergoing several paradigm shifts driven by the need to solve a set of interconnected, language-agnostic technical challenges. In summary, the evolution of input method technology has roughly gone through the four stages shown in Figure 1: starting with Rule-Based Foundations (1980s–2000s) reliant on handcrafted rules, collocation dictionaries, and finite-state machines; advancing into the Statistical Revolution (1990s–2010s) with N-gram model, Hidden Markov Model (HMM), Support Vector Machine (SVM), and Statistical Machine Translation (SMT); followed by the Neural Leap (2010s–2020s) characterized by Neural Network Language Model (NNLM), Recurrent Neural Networks (RNN) including GRU and LSTM, and the Transformer architecture; and currently entering the Generative Paradigm (2020s–present) with models such as PinyinGPT and GeneInput, which leverage large language models to enable more natural, context-aware, and personalized input experiences.

3.1 Ambiguity Resolution

The most fundamental and universal obstacle in back-transliteration is resolving the inherent ambiguity arising from the many-to-one mapping between phonetic Latin strings and native script candidates. This challenge, manifesting across all writing systems but with varying complexity due to homophony density and script structure, has been the primary driver of methodological innovation.

Early systems across language communities tackled this problem by encoding explicit linguistic knowledge. For Chinese, Xiaolong et al. (2004) applied rough set theory to systematically mine Pinyin-to-character conversion rules from large-scale corpora, constructing a Linguistic Information Table to encode contextual features and extracting high-confidence rules—an approach that demonstrated the feasibility of automating rule extraction but remained brittle. The Japanese community pursued a similar path with manually constructed resources; Koyama et al. (1998) built a collocation dictionary of 135,000 entries, significantly improving accuracy for short sentence processing. A more sophisticated, syntax-aware approach was pioneered by Abe et al. (1986), who constructed a morpheme network to handle segmentation and homonym ambiguities in non-segmented Japanese input, employing heuristic search and case grammar analysis to achieve a notable 90.5% translation accuracy and laying a theoretical foundation for deep linguistic disambiguation. For languages like Urdu, where vowel omission in the script creates specific ambiguities, rule-based finite-state transducers (Bögel, 2012) provided effective solutions without probabilistic models.

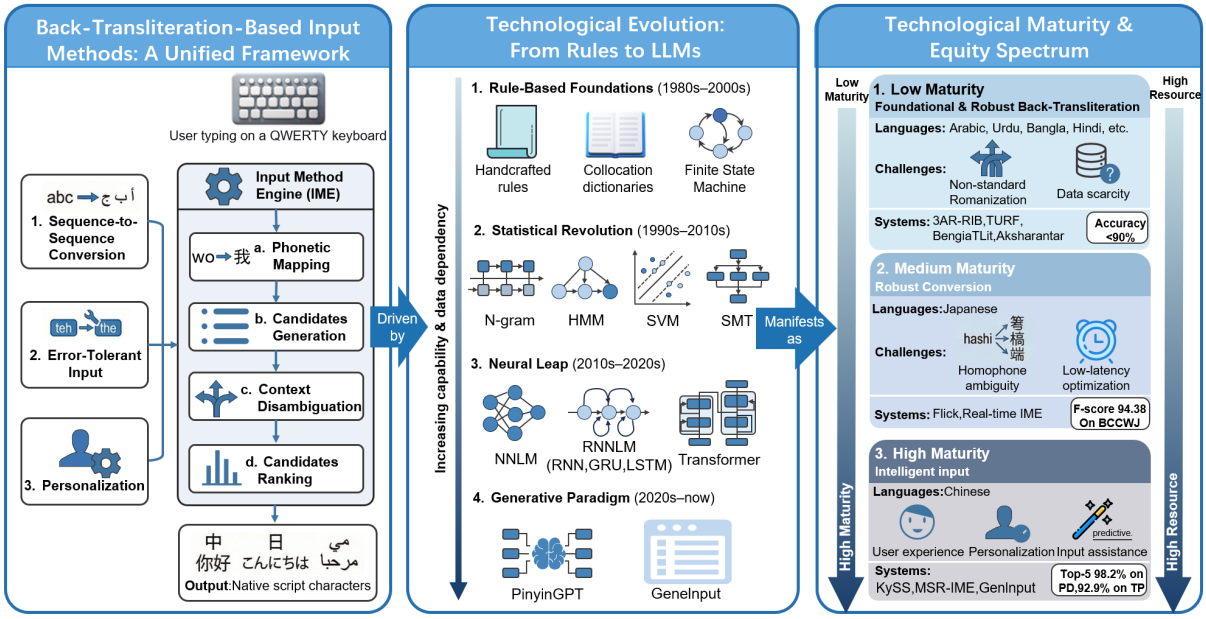

Figure 1: Overview of back-transliteration-based input method.

The shift towards data-driven paradigms enabled a more scalable and robust approach to modeling ambiguity. The core idea of probabilistic modeling was embraced nearly simultaneously. Chen and Lee (2000) pioneered the use of n-gram language models within a Bayesian framework for Chinese Pinyin input, while Mori (1999) formalized Japanese Kana-Kanji conversion as a maximum a posteriori estimation problem. A significant conceptual leap was the reconceptualization of the task as statistical machine translation (Yang et al., 2012), framing the conversion as a translation between phonetic and character sequences. This era also saw the rise of discriminative learning frameworks. Jiang et al. (2007) utilized Support Vector Machines to integrate long-distance contextual features for Chinese, and Tokunaga et al. (2011) applied structured SVMs to Japanese conversion, the latter achieving an F-score of 92.3 on the BCCWJ corpus and demonstrating the superiority of discriminative models in data-rich scenarios. The statistical paradigm culminated in unified joint models, such as the graph model by Jia and Zhao (2014) that integrated segmentation, error correction, and conversion into a single optimization problem, establishing a core architectural principle.

Neural networks, particularly sequence-to-sequence models with attention mechanisms, brought a paradigm shift by enabling end-to-end learning of long-range contextual dependencies critical for disambiguation. Initial work focused on enhancing prediction with Neural Network Language Models (Chen et al., 2015). The field quickly advanced to full encoder-decoder architectures. For Arabic-English transliteration, attention-based models achieved excellent performance (Hadj Ameur et al., 2017). For Chinese, the purely attention-based PERT model (Xiao et al., 2022) was designed explicitly for Pinyin-to-character conversion, outperforming RNN-based models in capturing semantic coherence. Japanese research addressed the unique demands of real-time input with an alignment-based decoding policy for neural Kana-Kanji conversion, achieving a top F-score of 94.58 on BCCWJ (Sarhangzadeh and Watanabe, 2024). The gated-attention sequence-to-sequence model by Huang et al. (2018) further exemplified how neural mechanisms could dynamically incorporate historical typing context for improved Chinese input. For Indic languages, the combination of the massive Aksharantar corpus and the Transformer-based IndicXlit model (Madhani et al., 2023) demonstrates that sophisticated, data-driven disambiguation is achievable even in historically lower-resource contexts, highlighting the transformative role of curated datasets.

The advent of large language models is fundamentally redefining ambiguity resolution, transitioning from disambiguation to intent-aware generation. Early exploration by Tan et al. (2022) adapted Chinese GPT for Pinyin input, revealing both potential and challenges like performance

degradation with abbreviated input (73.2% Top-1 accuracy on PD). The GeneInput framework (Ding et al., 2024) represents a paradigm shift, unifying multiple input modes under a single generative architecture via Full-mode Keystroke-to-Character (FK2C) modeling and reinforcement learning from human feedback, achieving 88.4% Top-1 accuracy on PD and excelling in noisy scenarios (77.0% Top-1 on TP).

3.2 Adaptation under Resource Constraints

For the majority of the world's languages, the pressing research agenda is not refining intelligent disambiguation but achieving robust conversion from the noisy, non-standard Romanized text prevalent in digital communication, all while operating under severe data and tooling constraints. This challenge has necessitated pragmatic, often hybrid, solutions tailored to specific socio-linguistic contexts.

The initial phase for many low-resource languages involved combining basic linguistic rules with minimal available data. For Indian languages, early systems like that of Raj (2014) combined n-gram language identification with rule-based back-transliteration, achieving moderate accuracy but being fundamentally constrained by data scarcity. Similar rule-based efforts were seen for Persian (Maleki and Ahrenberg, 2008), which used syllabification and finite-state methods for converting Romanized Persian (Dabire), though it struggled with loanwords, and for Urdu (Irvine et al., 2012), where a hidden Markov model combined with a dictionary for joint de-Romanization and normalization reduced error rates by over 50% on informal SMS text.

Arabic and its related script communities present a paradigmatic case where technology development has been directly shaped by widespread user adoption of non-standard orthography. The primary challenge here is converting *Arabizi* or *Franco-Arabic*—highly informal Romanized conventions born from early digital constraints. Early work by Chalabi and Gerges (2012) employed a hybrid rule-based and statistical machine translation approach to generate and score candidates, directly tackling the lack of standardization. This was significantly advanced by the 3AR-RIB system (Al-Badrashiny et al., 2014), a purpose-built pipeline of character-level finite-state transducers, morphological analyzers, and language models designed to convert Arabizi into Conventional Orthography for Dialectal Arabic (CODA), addressing the core technical hurdle of vowel recovery and disambiguation within an abjad system.

A decisive turning point, powerfully demonstrated by the progress in Indian languages, has been the recognition that the primary barrier is often resource availability, not model architecture. The field shifted from algorithmic novelty to data-centric strategies. The release of the Dakshina dataset (Roark et al., 2020) first enabled robust, comparable evaluation of transliteration across multiple South Asian languages. This was followed by the monumental Aksharantar corpus—26 million transliteration pairs across 21 languages—and the accompanying IndicXlit model (Madhani et al., 2023), which leveraged the modern Transformer architecture to achieve state-of-the-art performance. This momentum continues with efforts to construct large-scale Romanized datasets from social media for individual languages like Assamese (Baruah et al., 2024) and Bangla (Fahim et al., 2024). This trajectory underscores a critical and replicable model: dedicated, community-driven resource construction can bootstrap low-resource languages from foundational struggles to a position where powerful, modern models can be effectively deployed, thereby breaking the cycle of digital marginalization.

3.3 Optimizing User Interaction

As the core conversion technology matures for a language, the research focus naturally expands from basic accuracy to encompass the entire user experience. This involves optimizing for efficiency, enabling personalization, and integrating intelligent features that transform the input method from a mere transcription tool into a collaborative platform.

Early efforts within the statistical paradigm focused on making models practical and efficient for deployment. A key concern was model size and speed, especially for mobile and embedded devices. Maeta and Mori (2012) introduced a statistical Japanese input method based on a phrase class n-gram model that maintained high conversion accuracy while significantly reducing model size, pursuing the goal of "small yet accurate" language modeling. For mobile-specific deployment, Wu et al. (2013) trained a Japanese IME on large-scale web data, employing an efficient n-pos model combined with a cloud-based 4-gram language model and multi-stage filtering. Beyond efficiency, enhancing utility for specialized users

was also explored; the CoCat system (Huang et al., 2015) integrated knowledge from Statistical Machine Translation directly into the typing process to provide real-time translation assistance for human translators.

The neural era enabled more sophisticated, adaptive, and interactive features. Predictive text capabilities were enhanced by hybrid models that integrated RNN Language Models with traditional n-grams (Ikegami et al., 2017). Neural machine translation was combined with information retrieval to power associative "cloud-based" input in Chinese IMEs, offering context-aware predictions and customizable associations (Huang and Zhao, 2018). A critical breakthrough for real-time usability was the incremental vocabulary selection method (Yao et al., 2019), which dynamically constructed relevant vocabulary subsets during neural decoding to achieve significant speedups, a solution with universal relevance. Personalization became a major theme, with frameworks like the online vocabulary adaptation system by Zhang et al. (2019), which allowed neural IMEs to continuously integrate user-confirmed terms, and MSR-IME (Jiang et al., 2022), which used dynamic representation storage and similarity-based retrieval for improved domain adaptability.

The generative paradigm, coupled with advanced evaluation, is pushing the boundary towards understanding and anticipating user intent. Frameworks like GeneInput (Ding et al., 2024) move beyond transcription, unifying fragmented input modes into a coherent, intent-aware generative process. This shift is paralleled by the development of user-centric evaluation metrics that better reflect real-world utility. The Keystroke Score framework (Jia and Zhao, 2013), pioneered for Chinese, quantifies the actual number of keystrokes a user must expend, shifting the evaluation focus from pure character accuracy to tangible user effort reduction—a metric of universal importance for assessing practical input efficiency. Together, these advances chart a course towards input methods that function as proactive cognitive assistants, capable of managing complex tasks and adapting seamlessly to individual user needs and contexts.

3.4 System Architecture and Performance Optimization

Beyond algorithmic accuracy and user-centric features, the practical success of an input method hinges on its system architecture and computational performance. The perennial challenge of balancing sophisticated functionality with the stringent latency, memory, and power constraints of real-world devices—particularly mobiles—has driven continuous innovation across technological paradigms.

The core problem is the efficiency-accuracy trade-off. Early statistical approaches addressed this through model compression and streamlined design, such as the phrase class n-gram model for Japanese that maintained accuracy with a reduced footprint (Maeta and Mori, 2012), and mobile-oriented systems that combined lean local models with cloud-based resources (Wu et al., 2013). The shift to neural models exacerbated computational demands, spurring focused work on efficient inference. The incremental vocabulary selection method (Yao et al., 2019) enabled real-time neural IMEs by dynamically pruning the search space, while alignment-based decoding policies (Sarhangzadeh and Watanabe, 2024) minimized latency for neural Japanese conversion.

In the generative LLM era, this architectural challenge intensifies. Models like GeneInput (Ding et al., 2024) offer unprecedented capability but at high computational cost. The central task is now to bridge this capability-efficiency chasm through aggressive research in model compression, efficient inference algorithms, and hardware-software co-design. The goal is democratizing access: ensuring that advanced, intelligent input can run fluently on the affordable devices used by billions worldwide, thereby preventing the LLM era from widening existing technological divides.

4 Cross Language Studies: A Spectrum of Technological Maturity

While a convergent technological evolution exists, the real-world development of back-transliteration systems is highly uneven across languages. We analyze this disparity along a maturity spectrum and conduct a simple test experiment to prove it.

4.1 Spectrum of Technological Maturity

The development of back-transliteration systems is highly uneven across languages.

For low-resource languages, research focuses on converting noisy Romanized text (e.g., Arabizi) under data scarcity. Progress has evolved from rule-based hybrids to data-driven pipelines, with large-scale dataset creation enabling modern Transformer-based models.

For high-resource languages, foundational conversion is solved. Research advances real-time efficiency and intelligent interaction. Japanese input epitomizes engineering refinement through optimized neural decoding. Chinese leads the generative shift with frameworks like GeneInput, moving from transcription to intent-aware generation.

This spectrum highlights a global technological inequity, where high-resource languages explore cognitive input while low-resource languages still struggle with basic conversion.

4.2 LLM Performance on Multilingual Back-Transliteration

We independently constructed a small multilingual dataset to test the performance of different large models on back-transliteration task. The performance of large language models on multilingual back-transliteration is evaluated using two relatively common metrics, as shown in Table 1. The first metric is the Character Error Rate (CER), which measures the character-level discrepancy between the predicted sequence and the ground truth. It is formally defined as follows:

$$\text{CER} = \frac{S + D + I}{N} \times 100\%, \tag{2}$$

where S, D, and I represent the counts of character substitutions, deletions, and insertions, respectively, and N is the total number of characters in the reference. A lower CER indicates higher accuracy in the fundamental grapheme-to-grapheme conversion.

The second metric, Sentence Accuracy (SA), provides a more stringent, holistic assessment. It measures the proportion of test instances for which the entire output sentence is perfectly correct:

$$\text{SA} = \frac{\text{Number of perfectly correct sentences}}{\text{Total number of sentences}}. \tag{3}$$

While CER reflects fine-grained transcription fidelity, SA directly correlates with practical usability by indicating how often a user obtains a fully correct result without any post-editing.

Through this table, we can catch a glimpse of the technological inequity. A very obvious phenomenon is that the CER rate of the same model varies greatly between different languages, and for the same language, the CER rate also differs significantly between different models. Moreover, sentence accuracy remains relatively low (mostly below 0.3), while Chinese achieves significantly higher accuracy (mostly above 0.35). This disparity illustrates the substantial gap between low- and high-resource languages in back-transliteration performance. Figure 2 illustrates this more intuitively. Among model families, the Qwen3 series shows no consistent improvement with increased parameters—for example, the 14B variant often matches or outperforms the 32B model (e.g., Arabic CER: 0.81 vs. 0.86)—suggesting that factors beyond parameter count, such as training data and architectural optimizations, play crucial roles. In contrast, the GPT series demonstrates steady improvement from GPT-3.5 to GPT-5.2, particularly in sentence accuracy (e.g., Chinese SA: 0.66 vs. 0.55 for GPT-4), indicating enhanced contextual understanding. Gemini3-pro-preview delivers the strongest overall performance, achieving nearly the lowest CER and highest SA across all languages. These observations confirm that while LLMs offer potential, their direct application remains uneven and involves efficiency–usability trade-offs, underscoring the need for continued development of specialized and equitable input technologies.

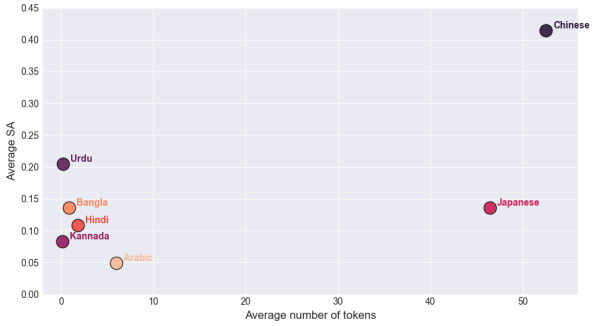

Figure 2: Scatter plot of sentence accuracy for each language. The X-axis represents the average number of tokens estimated based on Common Crawl, and the Y-axis represents the average sentence accuracy.

5 Resources, Evaluation, and Future Directions: A Roadmap for Equity

The cross-linguistic analysis conducted under the unified framework of back-transliteration has yielded one unambiguous conclusion: the primary challenge is no longer a lack of technical paradigms, but a crisis of equitable distribution and adaptation of these paradigms. This chapter synthesizes the insights from the survey to outline a concerted research agenda aimed explicitly at bridging this technological divide. We critically examine the transformative yet constrained role of Large Lan-

Model	Arabic		Bangla		Hindi		Japanese		Kannada		Urdu		Chinese	
	CER	SA	CER	SA	CER	SA	CER	SA	CER	SA	CER	SA	CER	SA
Open-Source														
Qwen3-32B	0.86	0.03	0.18	0.05	0.10	0.06	0.77	0.04	0.57	0.00	0.22	0.07	0.46	0.09
Qwen3-14B	0.81	0.06	0.15	0.05	0.12	0.03	0.92	0.03	0.57	0.01	0.16	0.09	0.48	0.12
Qwen3-8B	0.84	0.01	0.20	0.02	0.12	0.05	0.97	0.06	0.77	0.00	0.19	0.05	0.63	0.06
Deepseek-v3.2	0.70	0.08	0.09	0.08	0.07	0.07	0.77	0.07	0.19	0.14	0.08	0.23	0.15	0.44
GLM-4.6	0.74	0.06	0.08	0.19	0.07	0.14	0.41	0.10	0.13	0.10	0.08	0.14	0.19	0.38
Closed-Source														
GPT5.2-chat	0.76	0.05	0.16	0.19	0.24	0.14	0.60	0.15	0.73	0.11	0.13	0.29	0.19	0.66
GPT4	0.70	0.04	0.06	0.20	0.06	0.12	0.95	0.13	0.22	0.05	0.05	0.31	0.14	0.55
GPT3.5	0.71	0.05	0.08	0.09	0.06	0.11	0.35	0.12	0.23	0.02	0.06	0.22	0.21	0.37
Doubao-seed-1.6	0.77	0.04	0.07	0.11	0.06	0.11	0.55	0.10	0.26	0.03	0.08	0.13	0.50	0.54
Claude-opus-4.5	0.81	0.06	0.05	0.27	0.06	0.19	0.27	0.22	0.19	0.17	0.05	0.38	0.11	0.62
Gemini3-pro-preview	0.69	0.06	0.04	0.25	0.04	0.18	0.14	0.48	0.05	0.28	0.05	0.34	0.22	0.74

Table 1: Performance of Large Language Models on Multilingual Back-Transliteration (Character Error Rate & Sentence Accuracy). Models are grouped by open-source and closed-source.

Feature	Traditional IME (Statistical/Rule)	Generative IME (LLM-based)	Impact on Low-Income Users
Model Size	< 50 MB	> 7 GB (quantized 7B model)	**Storage Barrier:** Requires high-end phones
RAM Usage	< 100 MB	> 4 GB	**Hardware Barrier:** Apps may crash on budget devices.
Latency	< 10ms (On-device)	> 500ms (or network-dependent)	**Usability Barrier:** Typing flow is broken.
Energy	Negligible	High battery drain	**Utility Barrier:** Impractical for daily use.

Table 2: Comparison of Traditional IMEs and Generative IMEs in terms of resource requirements and implications for low-income users.

guage Models in this domain, and finally present a set of concrete, actionable initiatives designed to catalyze progress toward linguistic equity.

5.1 The Promise and Pitfalls of LLMs in Transliteration

The advent of Large Language Models has undoubtedly introduced a transformative force across NLP, prompting a critical re-evaluation of their role in back-transliteration. Their remarkable capacities in contextual understanding, world knowledge encoding, and few-shot learning present significant potential. Early explorations, such as adapting Chinese GPT for Pinyin input (Tan et al., 2022), have demonstrated that generative models can handle the conversion task within a unified sequence-to-sequence framework, showing promise in capturing semantic intent beyond mere phonemic mapping.

However, the direct application of monolithic, general-purpose LLMs as core engines for real-time input methods faces fundamental and likely insurmountable constraints rooted in the unique requirements of the domain. The foremost constraint is latency. Text input is a highly interactive task demanding millisecond-level response times to maintain a user's cognitive flow and typing rhythm. The autoregressive decoding nature and massive parameter count of state-of-the-art LLMs inherently conflict with this stringent latency requirement, even with advanced inference optimizations.

Closely related is the constraint of on-device deployment. Input methods are predominantly used on personal devices—smartphones, tablets, and laptops—with limited computational power, memory, and battery life. Deploying a model with billions of parameters locally is often impractical, while reliance on cloud-based APIs introduces unacceptable latency, privacy risks, and dependency on network connectivity. This creates a critical accessibility issue: if advanced "cognitive input" powered by large LLMs becomes the new standard, it risks exacerbating the digital divide, favoring only users with high-end devices and robust internet access, thereby widening the very equity gap this survey critiques.

As illustrated in Table 2, traditional IMEs are lightweight, efficient, and capable of running locally with minimal resource consumption. In con-

trast, LLM-based generative IMEs demand substantially more storage, memory, and computational power, often exceeding the capabilities of affordable mobile devices commonly used in low-income regions. This disparity exacerbates the digital divide, as users with limited access to high-end hardware are unable to benefit from advanced input features such as context-aware predictions, personalized adaptations, and robust error tolerance.

Therefore, the path forward lies not in replacing specialized, efficient back-transliteration engines with monolithic LLMs, but in strategically leveraging LLMs as enablers within a hybrid ecosystem. One promising direction is using LLMs as "teacher models" for knowledge distillation. The rich contextual and semantic knowledge of a large LLM can be distilled into a much smaller, specialized "student" model that retains much of the performance while meeting the efficiency and latency demands for on-device deployment. Another crucial role is data synthesis. For low-resource languages, LLMs can be prompted to generate high-quality, diverse synthetic training data—both standard Romanization pairs and challenging noisy variants—to bootstrap and enhance the training of traditional, efficient models where real-world data is scarce (Madhani et al., 2023). In this symbiotic relationship, LLMs act as powerful upstream tools for knowledge creation and transfer, while the core input experience remains powered by efficient, dedicated, and equitable models.

5.2 Future Research Directions: Concrete Pathways Toward Equitable Input

To translate the diagnosed inequities into actionable progress, we propose three concrete research initiatives. Each addresses a critical gap identified in our survey and is designed as a focused, community-driven call to action.

A Universal Transliteration Evaluation Protocol. The evaluation metrics for input methods across different languages are quite fragmented. These metrics include character and sentence accuracy, Top-k, f-score, and non-standard accuracy. Chinese Pinyin input methods also have a unique KySS evaluation standard. We advocate for a community-developed Universal Transliteration Evaluation Protocol(UTEP). It not only uses unified multidimensional metrics to evaluate the core accuracy of input methods but also quantifies the number of keystrokes by users to assess input efficiency. This protocol aims to create a common benchmark, shifting research incentives from narrow academic accuracy toward practical, user-centric performance for all languages.

Sentence-Level Modeling for Morphologically Rich Languages. For agglutinative languages and many low-resource languages, word-level conversion is fundamentally inadequate due to long-range grammatical dependencies. Building on Japanese advances from *bunsetsu* modeling (Kato et al., 2010) to neural aligners (Sarhangzadeh and Watanabe, 2024), we propose a targeted effort to develop and benchmark architectures for sentence-level and discourse-aware back-transliteration. This includes exploring syntactic-graph-enhanced attention and non-autoregressive decoders for morphological chains. For low-resource scenarios, this research can be accelerated using LLMs to generate context-rich synthetic training data, distilling this knowledge into efficient student models.

Time-Bound Low-Resource Language Technology Sprints. Inspired by the transformative impact of datasets like Dakshina (Roark et al., 2020) and Aksharantar (Madhani et al., 2023), we propose coordinated "Technology Sprints." Each sprint would target a specific language family to deliver, within 12-18 months, an open-source "technology pack": a curated benchmark dataset, an efficient pre-trained conversion model, a full UTEP evaluation report, and a deployable input method prototype. This model demonstrates that rapid, significant advancement is achievable through focused, replicable community action.

6 Conclusion

This survey has established a unified framework for back-transliteration-based input methods, connecting previously fragmented research across languages and technologies. We have demonstrated a convergent evolution from rule-based and statistical systems to neural and generative paradigms, while also diagnosing a profound technological inequity reflected in a maturity spectrum from low-resource to high-resource languages.

In the era of large language models, specialized and efficient back-transliteration remains essential to ensure equitable, real-time input for all languages. Addressing this disparity is both a technical necessity and an ethical imperative. We call for concerted community efforts to bridge the gap, advancing toward a linguistically fair digital future.

Limitations

While this work highlights critical inequities in back-transliteration, our study has limitations. First, the multilingual dataset used for LLM probing is small-scale and may not fully capture the noisy, code-mixed nature of real-world user input. Second, our evaluation relies on static intrinsic metrics (CER/Accuracy) rather than interactive human-subject studies or on-device latency benchmarking, which are crucial for assessing the practical usability of Input Methods. Third, our LLM experiments are limited to zero/few-shot settings without exploring fine-tuning or addressing potential data contamination in closed-source models. Finally, our linguistic analysis primarily focuses on Asian and Middle Eastern scripts, leaving the unique challenges of African and indigenous American languages for future exploration.

References

Masahiro Abe, Yoshimitsu Ooshima, Katsuhiko Yuura, and Nobuyuki Takeichi. 1986. A kana-kanji translation system for non-segmented input sentences based on syntactic and semantic analysis. In *Coling 1986 Volume 1: The 11th International Conference on Computational Linguistics*.

Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash, and Owen Rambow. 2014. Automatic transliteration of Romanized dialectal Arabic. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 30–38, Ann Arbor, Michigan. Association for Computational Linguistics.

Hemanta Baruah, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2024. AssameseBackTranslit: Back transliteration of Romanized Assamese social media text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1627–1637, Torino, Italia. ELRA and ICCL.

Tina Bögel. 2012. Urdu - Roman transliteration via finite state transducers. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 25–29, Donostia–San Sebastián. Association for Computational Linguistics.

Achraf Chalabi and Hany Gerges. 2012. Romanized Arabic transliteration. In *Proceedings of the Second Workshop on Advances in Text Input Methods*, pages 89–96, Mumbai, India. The COLING 2012 Organizing Committee.

Shenyuan Chen, Hai Zhao, and Rui Wang. 2015. Neural network language model for Chinese Pinyin input method engine. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 455–461, Shanghai, China.

Zheng Chen and Kai-Fu Lee. 2000. A new statistical approach to Chinese Pinyin input. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 241–247, Hong Kong. Association for Computational Linguistics.

Keyu Ding, Yongcan Wang, Zihang Xu, Zhenzhen Jia, and Enhong Chen. 2024. Generative input: Towards next-generation input methods paradigm. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3658–3669, Bangkok, Thailand. Association for Computational Linguistics.

Md Fahim, Fariha Tanjim Shifat, Fabiha Haider, Deeparghya Dutta Barua, MD Sakib Ul Rahman Sourove, Md Farhan Ishmam, and Md Farhad Alam Bhuiyan. 2024. BanglaTLit: A benchmark dataset for back-transliteration of Romanized Bangla. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14656–14672, Miami, Florida, USA. Association for Computational Linguistics.

Mohamed Seghir Hadj Ameur, Farid Meziane, and Ahmed Guessoum. 2017. Arabic machine transliteration using an attention-based encoder-decoder model. *Procedia Computer Science*, 117:287–297. Arabic Computational Linguistics.

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: integrating machine translation effectively and imperceptibly. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1163–1169. AAAI Press.

Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon IME: Neural-based Chinese Pinyin aided input method with customizable association. In *Proceedings of ACL 2018, System Demonstrations*, pages 140–145, Melbourne, Australia. Association for Computational Linguistics.

Yafang Huang and Hai Zhao. 2018. Chinese Pinyin aided IME, input what you have not keystroked yet. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2923–2929, Brussels, Belgium. Association for Computational Linguistics.

Yukino Ikegami, Yoshitaka Sakurai, Ernesto Damiani, Rainer Knauf, and Setsuo Tsuruta. 2017. Flick: Japanese input method editor using n-gram and recurrent neural network language model based predictive text input. In *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 50–55.

Ann Irvine, Jonathan Weese, and Chris Callison-Burch. 2012. Processing informal, Romanized Pakistani text messages. In *Proceedings of the Second Workshop on Language in Social Media*, pages 75–78, Montréal, Canada. Association for Computational Linguistics.

Zhongye Jia and Hai Zhao. 2013. KySS 1.0: a framework for automatic evaluation of Chinese input method engines. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1195–1201, Nagoya, Japan. Asian Federation of Natural Language Processing.

Zhongye Jia and Hai Zhao. 2014. A joint graph model for Pinyin-to-Chinese conversion with typo correction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1512–1523, Baltimore, Maryland. Association for Computational Linguistics.

Dongsheng Jiang, Xinyu Cheng, and Tianyi Han. 2022. Adaptive chinese pinyin ime for most similar representation. *IEEE Access*, 10:119533–119545.

Wei Jiang, Yi Guan, Xiao Long Wang, and Bing Quan Liu. 2007. Pinyin to character conversion model based on support vector machines. *Journal of Chinese information processing*, 21(2):100–105.

Shozo Kato, Chikahiro Araki, Shinichiro Hashimukai, Yasuhiro Ogoshi, Mikio Mori, and Shuji Taniguchi. 2010. Kana-to-kanji conversion method using markov chain model of words in bunsetsu. In *2010 4th International Universal Communication Symposium*, pages 154–160.

Yasuo Koyama, Masako Yasutake, Kenji Yoshimura, and Kosho Shudo. 1998. Japanese kana-to-kanji conversion using large scale collocation data. In *Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation*, pages 295–302, Singapore. Chinese and Oriental Languages Information Processing Society.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul NC, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023. Aksharantar: Open indic-language transliteration datasets and models for the next billion users. *Preprint*, arXiv:2205.03018.

Hirokuni Maeta and Shinsuke Mori. 2012. Statistical input method based on a phrase class n-gram model. In *Proceedings of the Second Workshop on Advances in Text Input Methods*, pages 1–14, Mumbai, India. The COLING 2012 Organizing Committee.

Jalal Maleki and Lars Ahrenberg. 2008. Converting Romanized Persian to the Arabic writing systems. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Shinsuke Mori. 1999. Kakuritsuteki model ni yoru kana kanji henkan [kana-kanji conversion by probabilistic model]. *Journal of Information Processing Society of Japan*, 40(7):2953–2964. In Japanese.

Abhinav Raj. 2014. Word level language identification and back-transliteration. In *Proceedings of the 6th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '14, page 74–79, New York, NY, USA. Association for Computing Machinery.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: the Dakshina dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2413–2423, Marseille, France. European Language Resources Association.

Armin Sarhangzadeh and Taro Watanabe. 2024. Alignment-based decoding policy for low-latency and anticipation-free neural Japanese input method editors. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8043–8054, Bangkok, Thailand. Association for Computational Linguistics.

Minghuan Tan, Yong Dai, Duyu Tang, Zhangyin Feng, Guoping Huang, Jing Jiang, Jiwei Li, and Shuming Shi. 2022. Exploring and adapting Chinese GPT to Pinyin input method. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1899–1909, Dublin, Ireland. Association for Computational Linguistics.

Hiroyuki Tokunaga, Daisuke Okanohara, and Shinsuke Mori. 2011. Discriminative method for Japanese kana-kanji input method. In *Proceedings of the Workshop on Advances in Text Input Methods (WTIM 2011)*, pages 10–18, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Xianchao Wu, Rixin Xiao, and Xiaoxin Chen. 2013. Using the web to train a mobile device oriented Japanese input method editor. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1209–1215, Nagoya, Japan. Asian Federation of Natural Language Processing.

Jinghui Xiao, Qun Liu, Xin Jiang, Yuanfeng Xiong, Haiteng Wu, and Zhe Zhang. 2022. Pert: A new solution to pinyin to character conversion task. *Preprint*, arXiv:2205.11737.

W. Xiaolong, Chen Qingcai, and D.S. Yeung. 2004. Mining pinyin-to-character conversion rules from large-scale corpus: a rough set approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2):834–844.

Shaohua Yang, Hai Zhao, and Bao-liang Lu. 2012. Towards a semantic annotation of English television news - building and evaluating a constraint grammar FrameNet. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 333–342, Bali, Indonesia. Faculty of Computer Science, Universitas Indonesia.

Jiali Yao, Raphael Shu, Xinjian Li, Katsutoshi Ohtsuki, and Hideki Nakayama. 2019. Enabling real-time neural IME with incremental vocabulary selection. In *Proceedings of the 2019 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers), pages 1–8, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhuosheng Zhang, Yafang Huang, and Hai Zhao. 2019. Open vocabulary learning for neural Chinese Pinyin IME. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1584–1594, Florence, Italy. Association for Computational Linguistics.

A: Representative Works of Chinese and Japanese Input Methods

We have found that in recent years, research on Chinese Pinyin input methods is interconnected and evaluated on the same benchmark. We have compiled the experimental results of a series of representative methods, as shown in the table 3.

Method	PD		TP	
	Top-1	Top-5	Top-1	Top-5
Google IME	70.9	78.3	57.5	63.8
Cocat	61.4	73.1	-	-
Aided-IME	71.0	80.8	-	-
Moon-IME	70.5	79.8	-	-
On-P2C	71.3	80.5	71.9	89.7
Pinyin-GPT	73.2	84.1	-	-
MSR-IME	90.6	97.8	64.9	85.2
GeneInput	88.4	96.2	77.0	92.9

Table 3: Performance comparison of representative Chinese Pinyin Input Methods on PD and TP datasets

On the standard dataset PD, MSR-IME demonstrates outstanding performance, achieving a Top-1 accuracy of 90.6% and a Top-5 accuracy of 97.8%, indicating the powerful performance of an end-to-end model based on a specially optimized transformer architecture. In contrast, methods based on the simple NNLM architecture (Cocat) and those based on the RNNLM architecture (Aided-IME, Moon-IME, On-P2C) lag behind, reflecting their limitations in model capacity and contextual modeling. Although generative approaches like Pinyin-GPT and GeneInput do not surpass MSR-IME on PD, their Top-1 and Top-5 accuracy are significantly higher than other methods, demonstrating the competitiveness of generative models.

On the TP dataset, which contains real-world noisy input, the advantages of generative methods become more pronounced. GeneInput leads significantly with a Top-1 accuracy of 77.0%, and its Top-5 accuracy reaches 92.9%, highlighting the strong adaptability of generative architectures in noise tolerance and contextual inference. Among traditional methods, only On-P2C and MSR-IME provide complete results on TP, and both underperform compared to GeneInput. Notably, MSR-IME's Top-1 accuracy drops from 90.6% on PD to 64.9% on TP, revealing its lack of robustness in non-standard input scenarios.

Overall, this table confirms the excellent performance of the neural network paradigm and the generative paradigm in input method tasks. Generative methods, in particular, excel in noisy environments, demonstrating their potential in intent understanding and error-tolerant input, and pointing the way for the design of future input systems tailored to real-world scenarios.

Similar to the Chinese Pinyin input method, we have organized the representative works of Japanese input methods at each stage on the same benchmark, as shown in the table 4.

Method	Precision	Recall	F-score
Phrase Class	90.25	90.58	90.41
Discriminative	92.2	92.4	92.3
Ensemble	93.7	93.6	93.7
Alignment-Based	94.47	94.74	94.58

Table 4: Evolution of Japanese Input Method performance on the BCCWJ benchmark

Analysis of the Japanese results reveals a clear evolutionary trajectory. From the initial Phrase Class model to the advanced Alignment-Based decoding policy, F-scores progressively improved from 90.41 to 94.58. The transition from statistical phrase modeling to discriminative frameworks and finally to alignment-based neural approaches reflects a continuous optimization of boundary detection and semantic coherence mechanisms. Particularly noteworthy is the Alignment-Based method's ability to significantly improve recall while maintaining high precision, indicating superior handling of complex contextual dependencies in Japanese text composition.