



Unsettled Scripts: Script Invention and the Bind of Digital Recognition

ANUSHAH HOSSAIN 

SPECIAL COLLECTION:
THE DIGITAL LIVES
OF GREATER SOUTH
LANGUAGES

ARTICLES – DIGITAL
MODERN LANGUAGES



ABSTRACT

This article investigates the invention of new writing systems, or “neographies”, and their uncertain trajectories in a digital age. It offers a working definition of neographies and surveys key themes in the literature: the motivations for script invention, factors influencing adoption, and the role of digital tools in enabling or impeding success. In particular, it attends to the difficulties of standardizing a script in a technical code when the script is still evolving and expanding, and the ways in which technologists have become involved in decisions about a script’s future. It uses the case study of the Adlam script, tracing its path from grassroots dissemination to Unicode encoding and subsequent refinements. The article concludes by outlining new directions for the study of neographies.

CORRESPONDING AUTHOR: Anushah Hossain

University of California,
Berkeley, US
anushah.h@berkeley.edu

TO CITE THIS ARTICLE:

Hossain, Anushah 2026
Unsettled Scripts: Script
Invention and the Bind of
Digital Recognition. *Modern
Languages Open*, 2026(1):
8 pp. 1–25. DOI: [https://doi.
org/10.3828/mlo.v0i0.590](https://doi.org/10.3828/mlo.v0i0.590)

The Fula language is used by one of the largest ethnic groups in Africa – over fifty million people across twenty countries from Senegal to Sudan. Yet for most of its history, Fula was only ever written in foreign scripts: either Ajami (a modified Arabic) or Latin. For many Fula speakers, these adapted writing systems never comfortably captured the sounds of the language and were marred by inconsistent orthographic practices that complicated communication.

This absence of a native script for Fula bothered Guinean brothers Abdoulaye and Ibrahima Barry. In 1989, at just ten and fourteen years old, they began inventing a script from first principles. After school, they would sit in their room, pronounce sounds aloud in their native tongue, and take turns sketching shapes to represent them. Within a few months they had devised a full character set. Stylized as ADLaM, the script's name derives from its first four letters (A, D, L, M) and stands for *Alkule Dandayde Leñol Mulugol* or, “the alphabet that protects the people from vanishing”.

Over the next few decades, Adlam spread from the brothers' childhood room and into the markets of their small town, and eventually into printed books and dedicated schools. Today, it is available globally – on social media platforms, in online classrooms, and on digital devices, alongside pen and paper. Adlam met a practical need by providing a standardized system that fitted the phonology of Fula. For many Fulani women, who were otherwise excluded from schooling, the network of Adlam after-school programmes has offered a pathway to literacy. The script also alleviated a cultural burden, giving the Fulani something beautiful of their own to take pride in and offering a reclamation of dignity eroded by decades of colonization (Barry et al.).

This article situates Adlam within a larger but understudied phenomenon: the invention of scripts whose designs are deliberately created prior to widespread adoption. About half of the known scripts in the world – at least 150 out of approximately 300 – are neographies, and nearly fifty have been introduced in the twenty-first century (Brookes). These *modern* neographies face distinct challenges in achieving recognition in today's digital infrastructure. Their fates are tightly bound to the implementation of the script in digital fonts, keyboards, operating systems, and the underlying character codes. These supports not only enable smooth participation in global communications networks but also confer public recognition and a measure of legitimacy. As recent inventions, however, their relatively uncertain status can complicate efforts to formalize their digital representations – presenting a challenge for both communities and technologists.

This article brings together the growing body of literature on neographies and outlines the emerging set of issues they face in digital infrastructure. In particular, it examines the dilemmas of standardizing a script in technical code while the script itself is still evolving, and the converse difficulties of building adoption without digital support. Using Adlam as a case study, it interrogates the slippery concept of “script stability” in a digital environment and concludes by identifying new directions for research on neographies.¹

1. CONSTRUCTING THE CATEGORY: WHAT IS A NEOGRAPHY?

“Neography” – meaning new writing – is still an emerging term in academic literature.² While the word is sometimes taken to mean “recent”, “new” here refers to a novel construction whose rules are determined by design. Academics often contrast neographies with so-called “natural” scripts – a heuristic term for writing systems whose origins are obscured by history and which are thought to have evolved organically over long periods, rather than being deliberately designed at a single point in time (Sanders 26). If one imagines a family tree of scripts, neographies might be pictured as fresh saplings planted alongside it, rather than branches growing from

¹ The author currently serves as Director of the Script Encoding Initiative, a research project that supports proposals to the Unicode Standard and collaborates with script communities to develop fonts, keyboards, and documentation. SEI provided funding for the Unicode proposals supporting the Adlam script and continues to support research and projects related to the digital implementation of neographies. This article does not advocate for or against the inclusion of any specific script, but instead offers a critical examination of the tensions surrounding the digitization of neographies today.

² Other terms sometimes appear in the literature: “constructed scripts” is occasionally used in academic settings as a synonym for neographies, while “conscript” is the label adopted in hobbyist communities (for instance, the ConScript Unicode Registry) for artistic or intellectual projects not usually intended for community use.

the trunk; they are new inventions whose emergence can be tied to an inventor, a place, and a time.

In practice, the line between neographies and so-called natural scripts can blur. Cyrillic, for example, was devised in the ninth century by priests drawing on the Greek alphabet. Its deliberate creation at a single point in time aligns it with neographies, yet its subsequent gradual evolution and diversification resembles the trajectory of so-called natural scripts. Depending on one’s emphasis, Cyrillic could be placed in either category.

Several further distinctions are useful in constructing the category of neographies. Where the few academic treatments of “neography” tend to refer to new orthographies or character additions within established scripts, I use the term to highlight cases where wholly new, visually distinctive character sets are created. As [Philippa Steele](#) has argued, visual distinctiveness can occur at multiple levels: orthographies of the same script may be distinguished by diacritics (as in Polish, Vietnamese, or Turkish); distinctiveness can also be enhanced through typeface or stylistic choices; and, at the far end of the spectrum, some newly created writing systems are constructed *ex novo* with visual principles unrelated to earlier scripts. Neographies, as I define them here, sit closer to this last category: they are not merely new orthographies of an existing character set, but rather scripts designed from scratch with forms that set them apart.

At the same time, not all neographies achieve distinctiveness through entirely novel forms. The Cherokee script, for instance, reuses the shapes of Latin letters but assigns them wholly unrelated phonemic values. Other scripts, such as the Lisu script, similarly borrow familiar shapes while giving them new functions. These cases show that visual borrowing does not preclude neographic status: what ultimately defines such scripts is their structural independence from the systems whose forms they resemble.

The neography category also focuses only on scripts intended for community adoption, not those designed primarily as an artistic, intellectual, or recreational endeavour. This measure therefore encompasses scripts such as Hangul, invented by King Sejong in the fifteenth century for Korean, but not Tengwar, J. R. R. Tolkien’s fictional script for Elvish, which was not necessarily invented for community use.

In sum, neographies can be defined as intentionally designed scripts meant for community use, typically distinguished by their independence from earlier scripts and, in most cases, by their visual distinctiveness. A set of examples is shown in [Figure 1](#).

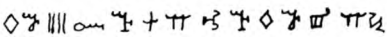
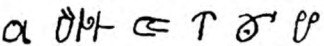
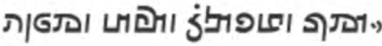
 Vai 1833 West Africa Vai Encoded in Unicode 5.1	 Pahawh Hmong 1959 Southeast Asia & diaspora Hmong Encoded in Unicode 7.0
 Afaka 1910 Suriname Ndyuka Not yet encoded in Unicode	 Tani 2001 Northeast India Tani languages Not yet encoded in Unicode
 Lisu 1915 Southwest China & Northern Myanmar Lisu Encoded in Unicode 5.2	 Ditema tsa Dinoko 2010s Southern Africa Bantu languages Not yet encoded in Unicode
 Ol Chiki 1929 Eastern India Santali Encoded in Unicode 5.1	 Rma 2017 Southwest China Qiang Not yet encoded in Unicode

Figure 1 Examples of neographies, indicating year and place of origin and target language(s).

Neographies are relatively overlooked in scholarly literature – a fringe topic within the already marginalized study of writing. Having not yet coalesced into a field of its own, the extant scholarship on neographies is scattered across Area Studies, Media Studies, Linguistic Anthropology, Classics, and Linguistics. It also raises recurring questions of motivation (“Why invent a new script?”), features (“What script grammars and visual features do we tend to see?”), and adoption (“What has led certain neographies to succeed and others to fail?”). These topics are briefly reviewed below.

1.1 CONTEXT AND MOTIVATIONS BEHIND SCRIPT INVENTION

The invention of a script begins with a fundamental choice: whether to adapt an existing set of characters or to create new forms altogether. As Dimitrios Meletis observes, this is “arguably the most pressing sociocultural question” in the creation of writing systems; very often, the pragmatic decision is to adopt Latin script, given both its global prestige and its ready compatibility with modern technologies (341). This article, however, focuses on the more demanding path of inventing wholly new, distinctive systems for community use.

Across regions, certain common patterns recur. New scripts frequently emerge as markers of identity, signalling affiliation or distance in relation to neighbouring traditions. Even when devised from scratch, scripts are created with a “conscious awareness of their neighbors” (Unseth, “Sociolinguistic Parallels” 33). In this sense, invention is never a neutral act of design but a situated response to political, cultural, and social conditions.

These dynamics, however, play out in different ways across regions. West Africa has been especially notable for the density and diversity of its neographies, many of which were framed as demonstrations of cultural pride and as assertions of independence from Latin and Arabic models. In South Asia, script invention is deeply tied to identity politics and frequently unfolds within competitive “script marketplaces”, where new systems not only vie with dominant scripts, but also compete against other recent inventions for the same language. In Southeast Asia, script invention has often been narrated in terms of cultural loss and restoration, with new systems positioned as a way to reclaim disrupted traditions and assert recognition. Taken together, these three regions, described further below, illustrate both recurring patterns and the distinctive historical conditions that have shaped their emergence.

1.1.1 West Africa

West Africa has long been a focal point for scholarship on indigenous writing. As Piers Kelly writes, “Few regions of the world rival West Africa for the sheer scale, diversity and dynamism of its indigenous writing traditions” (“Invention, Transmission and Evolution” 189). Some of the most prominent neographies to emerge from the region include Vai, N’Ko, and Adlam; lesser-known examples include Mende Kikakui, Loma, Kpelle, Bamum, and Masaba.

As David Dalby’s foundational studies in the 1960s emphasized, script inventors in West Africa often had characteristics in common: they were frequently young men, educated in mission contexts, familiar with Arabic and Latin scripts, and especially aware of the precedent set by the Vai syllabary invented in 1833. More importantly, Dalby highlighted a shared underlying motivation: a desire to demonstrate that Africans could create writing systems of their own, independent of both Latin and Arabic models. Peter Unseth has linked the proliferation of neographies in the mid-twentieth century to a moment of “great political aspiration and social change” between the First World War and decolonization (“Invention of Scripts” 23). Solomana Kante, who invented N’Ko in 1949 for Manding languages, stated that his motivation was “anger about foreigners who consider Africans as devoid of culture because they do not have writing systems of their own” (Pasch 84). Other script inventors were also motivated by a desire to correct the belief that Africans did not have writing of their own, such as Frédéric Bruly Bouabré, the inventor of the Bété script (Yeo and Cao).

Script invention in West Africa, and across the continent, continues today. Alongside the creation of wholly new systems, some efforts rework earlier traditions – for instance, modernized versions of the ancient Nsibidi or Adinkra systems (Yousuf et al.).

1.1.2 South Asia

Like West Africa, South Asia has produced a number of modern neographies. Some arose from the desire to capture phonological details more precisely, as in the case of the Wancho script, invented in 2001 in Arunachal Pradesh in India. One user explained, “we cannot write in English or Hindi our own names but in this Wancho script we can write our name in the perfect way” ([Simpson 70–71](#)).

At the same time, script invention in South Asia is deeply tied to struggles over identity and recognition. Scripts not only preserve languages against pressures from dominant others – such as the role Wancho plays in resisting the imposition of Assamese, Hindi, and English – but also act as visible markers of community distinctiveness. As Carmen Brandt observes, “script in South Asia is an important identity marker for many speech communities”, helping to stabilize and elevate minoritized groups ([79](#)). Ol Chiki, invented in 1925 by Raghunath Murmu for Santali, exemplifies this trajectory. Initially little used, it gained prominence as an emblem of Adivasi identity during the Jharkhand self-determination movement ([Choksi, *Graphic Politics*](#)).

Script competition is another enduring feature of the region’s bustling “script marketplace”. The past century has seen both movements to revive historical systems (e.g. Meetei Mayek for Manipuri) and campaigns to establish new ones (e.g. Ol Chiki for Santali). In such situations, different regions or communities might favour different scripts. For example, Sunuwar is written in two competing neographies: Jenticha (renamed Sunuwar and now encoded in the Unicode Standard) and Tikamuli, with reports suggesting that the former is more common in India and the latter in Nepal ([Holloway](#)). Similar divides are reported between Tolong Siki and Kurukh Banna for the Kurukh language, with the states of Jharkhand preferring the former and Odisha the latter ([Mandal](#)). In other cases, no consensus has emerged, as with the three rival scripts for Badugu in Tamil Nadu ([SEI Team, “Interview with Jan Kučera”](#)).

Not all inventions originate within communities themselves. In India, some scripts have been devised by outside inventors. For example, Prasanna Sree, a professor of English Literature at Andhra University, has created at least eighteen scripts for tribal languages, while Ganesh Devy, a prominent scholar of English Literature and founder of the Bhasha Research and Publication Center, has designed writing systems for eleven oral languages ([Brandt 92](#)). It remains unclear, however, how widely these scripts have been adopted.

1.1.3 Southeast Asia

Similar questions of recognition and contestation have also shaped script invention in Southeast Asia, a fertile though comparatively understudied region. Scholars have highlighted the region’s “orthographic plurality”, where both reforms of existing systems and wholly new scripts have emerged.³ Catalogues of contemporary scripts in the region attest to ongoing invention, including neographies such as Chak, Chong, and Danu ([Mitchell and Tep](#)). Locally created scripts such as Pahawh Hmong stand alongside missionary-designed projects such as the Pollard script, developed primarily for literacy and evangelism.

Piers Kelly situates these developments within a wider debate about the role of writing in upland Southeast Asian societies. He notes that communities in the region often narrate the loss of writing through stories of manuscript destruction, through fires, floods, or theft by enemies. In these accounts, invention becomes a way to restore a valued but disrupted tradition. Kelly contrasts this with James C. Scott’s argument that oral cultures have historically avoided writing in order to remain illegible to state power, suggesting instead that invention in Southeast Asia served as a means of asserting visibility and authority ([“Art of Not Being Legible”](#)).

Taken together, these regional case studies highlight the varied political, cultural, and linguistic contexts that have fostered script invention.

1.2 FEATURES OF NEOGRAPHIES

Scholars have examined not only how neographies came about, but also the internal features that characterize them. A first area of attention is the basic structure of neographies – whether they take the form of syllabaries, alphabets, or other models. In South and Southeast Asia,

³ See examples in the “Orthographic Plurality: Case Studies from Mainland Southeast Asia’s Borderlands” panels at the 7th Conference of the Asian Borderlands Research Network ([Imamura](#)).

many new inventions seem to depart from the alphasyllabic structure of Brahmi-descended scripts and instead follow alphabetic or syllabic models – for example, Coorgi-Cox, Kharpa, Tani, Chong, and Danu. While syllabaries are common in West Africa – due in part to the influence of the Vai script – alphabetic systems such as Adlam and featural systems such as Ditema tsa Dinoko demonstrate that there is no single model.⁴

Researchers have also emphasized the visual and cultural sources that script inventors draw on, from art forms to local annotation systems. Piers Kelly has argued that West African neographies are frequently informed by indigenous pictorial culture and annotation systems as much as by linguistic concerns ([“Invention, Transmission and Evolution”](#)). Logan Simpson similarly notes that neographies around the world often draw on traditional art forms, transforming them into repositories of endangered knowledge. The script Ditema Tsa Dinoko, for instance, incorporates patterns from southern African mural painting; Chiso draws from Indian *rangolis* and *alpanas*; and Avoiuli references sand-drawing practices from Vanuatu. Neographies can thus function as symbolic containers for cultural traditions, even when such associations are made retrospectively.

These visual qualities can, however, pose challenges. Questions of legitimacy often arise in specific contexts – especially in digitization or classification debates – where such scripts might be dismissed as decorative rather than communicative. African scripts, in particular, have frequently been caught in this tension between being treated as graphic art and being recognized as writing systems ([Hales](#); [Mafundikwa](#); [Mazzaferro](#); [Yeo and Cao](#)). In short, the features of neographies both reveal the range of their designs and hint at some of the challenges of having them recognized as writing systems.

1.3 FACTORS EXPLAINING NEOGRAPHY SUCCESS

Determining what makes a neography successful is not straightforward, since success can be defined in multiple ways: by the number of users, official recognition by a government or representative community body, or functional use in specific domains such as education, religion, or media. Each of these measures captures a different facet of a script’s reach and vitality. As Peter Unseth notes, the most successful West African scripts for ethnic revival are those that have fulfilled all three dimensions of language planning: status (social prestige and visibility), corpus (a growing body of texts), and acquisition (organized systems of teaching and learning) ([“Invention of Scripts”](#) 24–25).⁵

The measure of success also varies with the scope of a script’s intended user base. Adlam and N’Ko, for example, serve large, mobile populations; the migration of Mande speakers outside of West Africa enabled N’Ko literacy to spread globally, as diasporic users became de facto ambassadors and helped the “internationalization of the movement” ([Wyrod](#) 40). By contrast, scripts such as Masaba or Garay were designed for small, localized communities. Limited reach here does not necessarily signal failure, but reflects the intended scope of the invention.

Social infrastructure is often decisive in determining whether functional success takes hold. Formal institutions – schools, churches, and community associations – have long provided the backbone for literacy movements. The Winden Jangen organization created by the Barry brothers taught Adlam to tens of thousands of learners through in-person classes and, more recently, virtual instruction ([Barry et al.](#)). Government-organized workshops in Arunachal Pradesh helped the Wancho script gain wider acceptance ([Simpson](#)). Informal infrastructures matter as well: WhatsApp classrooms, YouTube tutorials, and social media groups now play an essential role in dissemination and maintenance.

State recognition can also elevate a script’s status. In South Asia, official acknowledgement often translates into concrete measures such as state approval for teaching in schools, the production of textbooks and other printed materials, and the formal sanction of a script as the designated medium for a language. The recognition of scripts such as Tolong Siki in

⁴ Further, the similarities between Vai and Cherokee have been widely noted, and scholars continue to debate the extent of any historical connection ([Tuchscherer and Hair](#)).

⁵ At the same time, Unseth notes that some scripts have achieved symbolic success without sustained functional use. The Bamum script, for instance, is widely recognized as a cultural emblem even though its everyday use is limited ([Unseth](#), [“Invention of Scripts”](#) 25).

Jharkhand illustrates how institutional endorsement can secure a place for new scripts in highly competitive environments.

Linguistic fit can also contribute to a script's vitality, since communities are more likely to adopt a system that they feel captures their language well. But fit alone rarely guarantees success; without the social, institutional, and digital infrastructures outlined above, even scripts that are highly regarded linguistically can struggle to gain traction.

Digital incorporation has become both a marker and facilitator of success. Logan Simpson's review of modern neographies explicitly presents inclusion in the Unicode character code as a measure of script status (58). Nishaant Choksi likewise highlights how Unicode accelerated the spread of Ol Chiki in eastern India: as Ol Chiki fonts adapted for Unicode became available in the 2000s, they quickly circulated in Santali-speaking regions and enabled a shift from manual to digital printing at relatively low cost (*"From Language to Script"* 1546). Peter Unseth observed that "failed" scripts tended to remain only in handwritten form (*"Invention of Scripts"* 28), while Helma Pasch noted that the availability of digital fonts for African scripts has bolstered their vitality (90). Jannis Kostelnik's study of Minim Dag Noore underscores the practical hurdles inventors face in achieving such support: creating fonts, engaging "regulatory authorities" such as the Unicode Consortium, and hiring professional developers (77).

These cases highlight that neographies require substantial investment to succeed; some scholars have suggested that "the option of creating new scripts from scratch is nowadays often not even considered given that new shapes are not immediately suitable for technology as they are not yet encoded in Unicode" (Meletis 344). At the same time, this constraint can also be understood as a compressed version of the centuries-long investments from which established scripts have long benefited.

In short, the literature suggests that successful neographies share similar characteristics: consistent use by their intended communities, support from social and educational infrastructures, endorsement by local or state institutions, and incorporation into digital communication systems.

This review does not claim to be an exhaustive representation of research on neographies, but calls attention to the dominant lines of questioning, broad differences in regional dynamics, and avenues for further exploration. Some aspects remain relatively under-explored, such as practical design concerns including legibility, ease of handwriting, or psycholinguistic factors, as well as the role of language policy in shaping the conditions for script invention and uptake. The following sections expand on issues of digital recognition, particularly with respect to recently invented neographies.

2. SEALING THE SCRIPT: CHALLENGES IN THE DIGITAL REPRESENTATION OF UNSETTLED SCRIPTS

When neography advocates – whether inventors or supporters – seek digital representation, they often confront the fact that their systems are still unsettled. Unsettledness can take more than one form. It can be social: arising in contexts with weak or hostile language policy, where official recognition is unlikely; in crowded script marketplaces, where multiple systems compete for community acceptance; or in situations where new systems struggle even to be acknowledged as scripts at all, dismissed instead as graphic or artistic practices, a distinction with real consequences for their eligibility to be incorporated into technical infrastructure. The issue can also be structural: character sets revised, glyphs reshaped, and features such as directionality, punctuation, or spacing altered in response to use. In both senses, neographies raise the question of when a script is "settled enough" to be fixed in a technical system.

Of course, some degree of script change is observed across most of the world's scripts, whether because of adaptation to different languages, the influence of novel writing implements and print materials, reform or rationalization by linguists or policymakers, or otherwise. But for "natural" scripts and older neographies, such processes often unfolded over decades or centuries before the advent of digital representation. By the time they were encoded, they had already accumulated social proof and relative stability.

By contrast, many recent neographies face technical standardization while they are still evolving. For technologists, this presents the problem of when to engage with what appears to be a moving target. For advocates, it creates the burden of proving that their script is stable enough to warrant investment in digital infrastructure. The following sections discuss how digital representations of scripts work today, what options are available to neographies, and what particular concerns technologists have with respect to the adoption and stability of these scripts.

2.1 THE CENTRALITY OF THE UNICODE STANDARD

At the heart of today's digital communications paradigm is the Unicode Standard. Unicode functions like an international Morse code for computers: it assigns unique "code points" – sequences of 0s and 1s – to each unit of a writing system, enabling consistent text storage, transmission, and rendering across platforms. For script communities seeking widespread usability, Unicode encoding is not the final goal, but is better thought of as an important inflection point: a prerequisite for digital support on an established, internationally recognized scale.

Unicode originated in the 1980s, when a wide variety of character-encoding schemes were in use, often tied to specific operating systems or constrained within national boundaries. With the rise of international commerce and globalization came pressure to ensure interoperability between systems, so that text could move across borders and devices without corruption. Just as importantly, vendors sought to reduce the complexity and engineering costs of supporting multiple encodings across products in many markets.

Joe Becker, one of Unicode's co-founders, first articulated the specific formulation of Unicode in a [1988](#) position paper. The design principles outlined there, and refined over the next decade, reflected not only engineering constraints but the need to balance political pressure, industry coordination, and long-term software stability ([Hossain, "Text Standards"](#)). Among these first principles was the central "characters, not glyphs" policy, which prescribed that Unicode would define only the abstract concepts of characters (e.g., the "Latin lowercase letter a") without dictating their visual representations (e.g., "a" vs. "ā"). Unicode thus assumed responsibility for the abstract building blocks of scripts, and not the infinite variations of their visual forms. It also introduced a strict stability policy: once published, the character repertoire, character and script names, and elements of script grammar would never be altered or removed. This principle was designed to build trust across the software industry, assuring developers that referencing a code point or block name would never later break functionality.

By its third edition in 1999, Unicode had begun to prove its staying power. During that period, a diplomatic agreement was reached with the developers of the competing ISO/IEC 10646 standard, leading to synchronized code charts and aligned development timelines ([Adal](#)). Microsoft and other major vendors also began to adopt Unicode, sending a clear signal of confidence to others in the industry. By 2007 Unicode – specifically the UTF-8 encoding – had become the dominant format for webpages and was integrated into all major operating systems ([Davis](#)). Despite its present dominance, this victory was far from inevitable: Unicode began as a fledgling project that made pragmatic decisions to meet the technical and political constraints of its time. Its scope has steadily expanded, from covering only major national and commercially important scripts to now supporting dozens of historical and modern minority scripts.

Publishing a script in the Unicode Standard sends a powerful signal to software vendors to consider studying the script and implementing support for it in digital devices. Vendors must update systems to recognize new Unicode blocks, ensure that usable fonts and keyboard input methods exist, and provide correct text behaviours such as line breaking, cursor movement, and character positioning. Even when operating systems ship fonts and text engines for a script, individual applications may lag behind, or may not implement needed behaviours at all, limiting real-world use.⁶

⁶ The nature of the script itself can compound these delays: systems with complex shaping or directionality are far harder to implement consistently. For example, Figma only introduced right-to-left (RTL) support in 2022, long after major operating systems had supported RTL scripts. Similarly, Slack still does not fully support RTL scripts, even though underlying platforms do. See [Field](#); [Slack](#).

2.2 WHY UNICODE MATTERS FOR NEOGRAPHIES

Though Unicode inclusion does not guarantee universal script availability, it is challenging to use new scripts on and across digital devices – whether for chatting, typesetting, or archiving – without it. One workaround available to users is the Private Use Area (PUA), a designated set of Unicode code points reserved for discretionary usage and testing. With enough coordination, a community can create fonts that map their script to this free space instead of to an already-assigned block. The advantage of PUA points is that users can still reap the benefits of the rest of the Unicode Standard, producing multi-script documents that contain both encoded and unencoded scripts.⁷ The drawback is significant, however: PUA code points lack shaping behaviour and are assumed to run left-to-right, making it difficult to use them for right-to-left scripts or those with complex joining or positioning requirements.

Alternatively, communities may resort to so-called “hacked” fonts. In a hacked font, the designer borrows the code points of an already-encoded script – commonly Latin or Arabic (particularly if joining behaviour or right-to-left directionality are required) – and substitutes the glyphs of the unencoded script in the font file, as in Figure 2. Users can then type on platforms that support Latin or Arabic but see their own letters on screen. This type of “font hack” is helpful for typesetting as it displays the desired content, but it comes at a cost.⁸

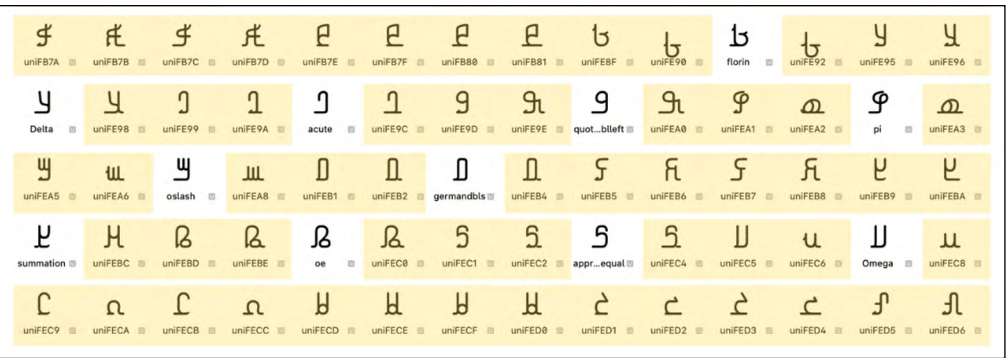


Figure 2 Example of a hacked font, with Adlam glyphs mapped on to Arabic (highlighted in yellow) and other mixed Unicode code points (courtesy of Neil Patel).

Both PUA-based and hacked-font approaches break the fundamental promises of digital text. Text produced in these ways cannot be reliably searched, sorted, or indexed online. It will not be picked up by search engines or surfaced easily on platforms such as YouTube, Facebook, or X/Twitter. It might not display correctly when sent across devices (for example, in a webpage or email), and it cannot be reliably archived with the assumption that it will be preserved with integrity – raising the risk of encountering *mojibake*, or garbled text, when old files are reopened. Every recipient of a document must install the same non-standard font to see it correctly, imposing a great deal of friction and requiring a baseline level of technical literacy among users. Furthermore, these strategies are largely limited to desktop environments.

For mobile phone users – arguably the most critical domain for minority script communities – the situation is even more constrained. Unlike desktop systems, where PUA or hacked fonts can sometimes be installed (albeit with major drawbacks), mobile platforms tightly restrict font customization. Users cannot freely add fonts, since third-party font files can pose serious privacy and security risks (*Indigenous Languages*). Vendors also limit fonts to preserve aesthetic consistency and avoid layout breakage in user interfaces. As a result, users are reliant on the small number of Unicode-compliant fonts preloaded at the platform level. Increasingly, vendors rely on Google’s Noto project, an open source font family that aims to cover all scripts encoded in Unicode. For minority communities, the most likely pathway to mobile support is to have the script encoded in Unicode, integrated into Noto, then loaded by default in the next software update.

7 An example of effective coordinated use of the Unicode Private Use Area (PUA) is the Medieval Unicode Font Initiative (MUIFI), which standardized PUA assignments for medievalist projects. Another long-running example is the ConScript Unicode Registry, an informal volunteer effort to assign PUA code points for constructed scripts (including recreational/artistic inventions).

8 This was common even in the pre-Unicode era as a strategy for making ASCII-based systems work for non-Latin scripts (though again, complex joining and positioning behaviour required further hacks to work correctly with Latin code points).

If there is no Unicode-compliant font available, mobile phone users can turn to third-party keyboard applications, which embed custom fonts within the app itself rather than at the platform level. These allow users to type as they please, screenshot or save the text as an image, and then send the image in their usual messaging application (Figure 3).

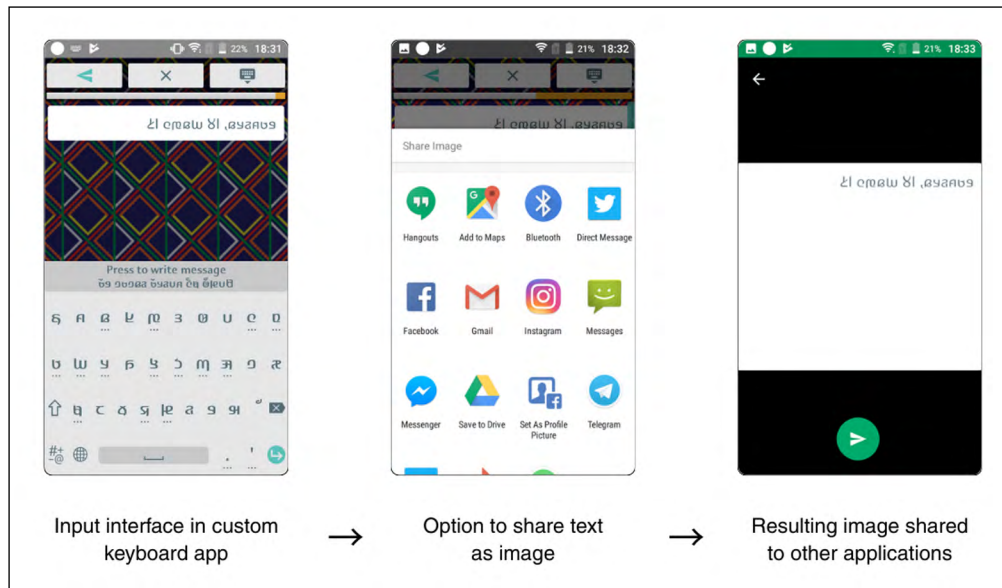


Figure 3 Workflow from the Tappirgal Adlam keyboard application made by JamraPatel.

While effective in limited contexts, such approaches introduce friction and lack interoperability. Some standalone apps (such as dictionaries or news services) can function with embedded fonts, but once text leaves those environments – when copied, shared, or archived – the absence of standardized encoding becomes an obstacle.

In sum, on a practical level, the fate of neographies is closely tied to Unicode encoding; without it, users must resort to burdensome, error-prone, and isolating strategies. On a symbolic level, inclusion in Unicode serves as a claim to legitimacy. Unicode code charts are published publicly and freely, and aspects that might appear mundane – block names, character labels, even the placement of scripts within the code charts – can become highly politicized (Hossain, “ISCI Imperialism”; Kučera). Despite Unicode’s disclaimers that the Standard is “not a language standard” nor “a pedagogical tool”, communities inevitably interpret inclusion as a valuable mark of recognition. Over time, Unicode maintainers have become more cautious, tying decisions more closely to evidence of community usage and stability, in order to minimize risks such as fuelling local conflicts or recognizing scripts before they gain genuine support (SEI Team, “Interview with Jan Kučera”). Still, it is worth recognizing that Unicode inclusion is often associated with an increased sense of pride and formalization, making it a goal for not only practical but symbolic ends.

2.3 HOW NEOGRAPHIES ENTER UNICODE

The process by which a script gets added to the Unicode Standard is far from straightforward, and it is even more complex for neographies whose newness limits the evidence available to prove their use and importance. Unicode acceptance for a new script occurs through a formal submission process. This begins with the preparation of a proposal document that outlines the script’s background, character names and repertoire, and technical properties such as joining behaviour, line breaking, directionality, positioning, and collation. The background section describes when and where the script was developed, its user base, how it has evolved, and whether it has institutional backing. These claims are then supported with an appendix of script samples, ideally demonstrating usage across a variety of media and contexts (D. Anderson).

Proposals are first directed to the Script Encoding Working Group (SEW, formerly known as the Script Ad Hoc), which meets monthly to review submissions. SEW participants include representatives from member institutions of the Unicode Consortium – primarily US-based

software companies – as well as independent experts from different countries. Participants typically have backgrounds in linguistics or computer science, often specializing in particular families of scripts (e.g., Indic, Arabic) and in different aspects of the text stack (e.g., text shaping, keyboards, or fonts). In important ways, the working group is a credentialed, experienced, and thoughtful adjudicator, representing a niche and deep area of expertise. Still, it is not unsusceptible to bias or ignorance, making decisions from a distance over a wide variety of complex cases.

Reviewers provide feedback and usually prompt several rounds of revision to clarify the encoding model or strengthen the case for usage (Hossain and Bashir). Once finalized, SEW recommends the proposal for inclusion during quarterly meetings of the Unicode Technical Committee (UTC), which coordinates the Standard’s overall development. Approved proposals then move into the production pipeline, including work in groups such as Properties and Algorithms, Editorial, and the Common Locale Data Repository (CLDR), where additional data and implementation details are added. In parallel, the proposal is reviewed by ISO/IEC 10646 subcommittees to maintain alignment between the synchronized international standards. This review phase may include further input from national standards body representatives and other experts (Unicode, “Submitting Character Proposals”).

From Unicode’s perspective, the complexity of this process reflects real implementation constraints. Adding a script, or even a single character, touches many interconnected data files across the Standard. Reversals once a script enters the production pipeline are particularly difficult, since changes ripple across many interconnected parts of the Standard. Multiple layers of review also help ensure that a script is actually implementable in operating systems and fonts; in some cases, proposals have raised challenges precisely because their behaviours would be prohibitively difficult to support in practice.⁹

For neographies, these complex processes pose significant obstacles. Inclusion requires that a compelling case be made to somewhat opaque institutions, in highly technical terms, and usually over several years. While the Unicode Consortium does not require official affiliation to participate, the technical specificity and relative obscurity of its operations can be daunting to outsiders. Its meetings occur in English and typically on US Pacific Time. The process takes several years at minimum and strongly favours scripts that already enjoy institutional backing, such as from governments, universities, or publishers (Unicode, “Submitting Character Proposals”). Many neographies emerge in precisely the opposite context: in politically marginalized communities without strong state recognition or formal language-planning bodies.

In such cases, the Unicode Consortium itself can become a key source of legitimacy. Writing in 2002, anthropologist David K. Jordan observed: “for most modern languages, governments define the standard form of an orthography, the characters which Unicode must accommodate. For dead languages, scholars do.” By contrast, he warned, “unorthodox, unstable, or unofficial” scripts risked being relegated to “corners as curiosities that cannot be seriously used in a world in which literacy has become intimately bound to electronic information flow” (111–12, 121). Jordan’s framing reflected a moment when national encodings loomed large, and when state or scholarly endorsement was often decisive.

These concerns remain salient today, though Unicode’s criteria have shifted. Rather than deferring only to state or scholarly authorities, decisions are now more explicitly tied to evidence of broad community usage and script stability. Yet institutional authority still matters: Unicode’s proposal guidelines encourage citing recognized bodies or named experts, and “unofficial” scripts often face higher hurdles in proving their legitimacy (Unicode, “Submitting Character Proposals”).

Since Jordan’s article, however, the Unicode Standard has continued to expand, and has evidently taken several calculated risks in accepting “unauthorized” scripts. Today the Unicode Standard is on Version 17.0 (released September 2025). The Standard now includes

⁹ This difficulty is not unique to neographies. For example, Egyptian hieroglyphs were encoded in Unicode in 2009, but because the Standard could not yet represent complex sign clustering, Egyptologists often relied on images rather than encoded text. Only with advances in shaping technology did accurate rendering become feasible, paving the way for current work on Mayan hieroglyphs.

172 scripts – up from 23 in the first release – with a median of around four added per update (Unicode, “Supported Scripts”). Depending on one’s definition, close to fifty of the encoded scripts could be considered neographies, including several invented as recently as 2015. Yet recent shifts in SEW practice suggest a tightening of standards and growing hesitation about encoding neographies (SEI Team, “Interview with Jan Kučera”). This shift warrants further discussion of what exactly motivates Unicode’s caution, and research into the consequences of these decisions.

2.4 THE CHICKEN-AND-EGG PROBLEM

Having outlined how neographies interact with Unicode in general, this section turns to the specific thresholds a script must cross to achieve encoding. Two criteria in particular – adoption and stability – are central to Unicode’s decision making, yet they pose paradoxical difficulties for newly invented scripts.

Adoption means proposal authors must provide substantial evidence that a script is actively used in its intended community, and that such usage extends beyond its inventor. Unicode reviewers look for signs that a script has a future: that it will persist and grow, not vanish with its originator.

What counts as valid evidence of adoption, however, is not always clear. Unicode guidance privileges print as the most persuasive medium – published books, periodicals, or other durable texts. Use in schools can also provide convincing evidence. Yet many neographies today circulate primarily through newer channels such as WhatsApp classrooms, Facebook groups, or YouTube tutorials (SEI Team, “Interview with Oreen Yousuf”). See Figure 4 for an example. These forms of activity are often inaccessible to reviewers – whether because they are unindexed, private, or ephemeral – and Unicode has no clear framework for weighing their significance. A central question remains: to what extent do metrics such as video views or group membership meaningfully demonstrate a script’s spread within a community?



Figure 4 Example of a teacher using an emoji as a pointer to virtually teach the Minim Dag Noore script. The lesson is screen recorded, posted on YouTube, and then reviewed in WhatsApp classrooms dedicated to the script (Yousuf 77).

Neography advocates, meanwhile, often view Unicode inclusion as necessary to catalyse broader adoption. As discussed in earlier sections, there is no seamless way to bring a script into everyday use on mobile or desktop platforms without Unicode support. Communities that lack encoding must rely on labour-intensive workarounds – handwriting large corpora of material, or developing non-standard fonts and keyboards. The underlying assumption is that the absence of Unicode support hinders language planning at scale, limiting a script’s potential for sustained use.

Alongside adoption, proposal authors must also demonstrate stability. In its narrowest sense, stability refers to consistent mappings of phonemes to graphemes – the information codified in a Unicode chart. More broadly, it encompasses the full range of script features formalized in the Standard and implemented across software: directionality, spacing, joining and positioning, glyph shapes, and character names. While inventors might attempt to prescribe these elements, once a script circulates more widely – an indicator of its “success” – usage patterns inevitably reshape its forms. Complex scripts may simplify over time; letterforms may shift when realized across different media and contexts.

Because stability is best demonstrated through extended and widespread use, Unicode typically waits until such evidence accumulates before encoding a script. Yet this raises deeper questions. How long must a script circulate before it is deemed “stable”? And can Unicode present itself as a neutral observer of stability when its own decision to encode might itself help fix a script’s features? The history of print offers a useful parallel: mechanization has often been credited with giving a “new fixity” or “regulation and fixation” to language (B. Anderson, 44; McLuhan 229). By extension, one might ask whether Unicode plays a similar role in stabilizing scripts, rather than merely recording stability after the fact.

These tensions are often described as a “chicken-and-egg problem” (Gnanadesikan 310). The very infrastructures that communities look to for recognition and legitimacy often require prior evidence of legitimacy and usage to justify representation. Encoding, in this sense, is not widely seen as a final validation of success but as a prerequisite for it.

The “chicken-and-egg” framework thus highlights both the expectations placed on neography communities and the questions that remain under-explored. To what extent can language planning proceed without large-scale digital support? How do different political or cultural contexts alter this calculus? While such questions call for broader comparative research, the following section turns to a detailed case study of Adlam to illustrate the close and, arguably, unavoidable entanglement of neography communities and technologists in the process of standardization.

3. ADLAM: A STUDY OF STABILITY

By most measures, Adlam is the most successful of recent neographies. It has become the benchmark against which other modern script inventions are compared, particularly those emerging from Africa. This comparison, however, can be unfair. Adlam represents an exceptional case, the product of decades of movement building and grassroots dedication that set a uniquely high bar. Moreover, its trajectory after Unicode encoding has at times been interpreted by reviewers as a sign of “instability”, given the refinements that have continued in its digital representation. That perception risks creating a burden for other neographies, which might be judged against Adlam’s perceived weaknesses as much as its undeniable achievements.

This section aims to present a balanced view of Adlam’s journey from 1989 to the present. The first part traces its evolution across nearly three decades, following the script through phases of invention and grassroots dissemination, movement building and suppression, and eventual international recognition. This overview, drawn largely from secondary accounts, sketches how Adlam grew from a local initiative into a transnational movement (see Table 1 for key milestones). The second part turns to the years since Unicode encoding, examining how the script’s digital representation has raised new questions of stability and standardization, based on closer analysis of primary materials.

1989	Adlam script devised by Ibrahima and Abdoulaye Barry
2000	Winden Jangen organization founded in Conakry, Guinea
2002	Ibrahima imprisoned for three months; starts <i>Karambol</i> newspaper upon release
2003–07	Barry brothers move to the United States
2008	Barry brothers commission the creation of three non-Unicode fonts (Pulaar, Fuuta, and Aissata) and a keyboard
2012–13	Barry brothers connect with Michael Everson via connections at calligraphy conferences First Unicode proposal for Adlam submitted
2014	Revised Unicode proposal accepted
2016	Adlam code block published in Unicode 9.0 Feature article on Adlam published in the <i>The Atlantic</i>
2017–19	System-level shaping and input support becomes available in Windows, Apple, and Android Proposal accepted to revise character shapes in Unicode
2024	Proposal accepted to revise character shapes and provide new character name annotations in Unicode
2025	New Adlam characters requested to support additional languages

Table 1 Timeline of developments of the Adlam script.

3.1 FROM INVENTION TO INTERNATIONAL RECOGNITION

After inventing Adlam in 1989, Ibrahima and Abdoulaye Barry spent the next decade spreading the script through grassroots networks. They first taught the script to their sister, then began teaching it in local markets, asking each learner to pass on the knowledge by teaching three others. Realizing the need for reading materials, they began by laboriously handwriting folk tales, then later practical references on subjects ranging from mathematics to infant care (Bach; Hasson; Waddell). Ibrahima later recalled the sheer intensity of those years: “I used to spend sixteen hours a day writing, because I used to do everything by hand” (Microsoft Unlocked).

During this period the script itself also evolved. The brothers realized that students advanced more quickly with unjoined letters, so they produced primers in simplified forms while reserving cursive writing for advanced learners (Barry et al.).

A breakthrough came in 1993, when Abdoulaye, still a high school student, persuaded a radio host in Conakry to let him present the script live on air. The broadcast introduced Adlam to a national audience (Waddell). In the years that followed, the mobility of Fulani farmers and traders across markets and borders helped carry the script beyond its initial circles, so that by the mid-1990s the brothers could walk into new towns and encounter strangers already reading and writing their alphabet (Waddell).

In the early 2000s, while studying at the University of Conakry, the brothers founded a student group called Winden Jangen – Fulani for “reading and writing” – to expand their network of learners. With the help of classmates, they transcribed more and more books by hand, sustained by small supports such as a gift of carbon-copy paper from a relative (Waddell).

In 2002, however, Adlam’s growing visibility provoked state suspicion. During his final year of university, Ibrahima was imprisoned for three months after military officers raided a Winden Jangen meeting. No charges were filed, but the brothers believed their literacy campaign had unsettled officials (Bach; Waddell). On his release, Ibrahima redoubled his efforts, producing a Fulani-language newspaper by hand and distributing editions through Winden Jangen, now with access to a photocopier and financial support from a local Guinean businessman (Hasson; Waddell). During this same period, Abdoulaye left for Portland, Oregon, where he was later joined by Ibrahima in 2007.

Once resettled in the United States, they turned to digital tools. Pooling their savings, they hired a Seattle font company in 2008 to develop three Adlam fonts and a keyboard (Barry et al.; Waddell). As Abdoulaye recalled, “We knew that the only way to spread Adlam faster and

publish more books was to have it in computers” (Waddell). These fonts relied on borrowed Arabic code points – the kind of font hack described in the previous section. They enabled cursive joining behaviour but carried major limitations: casing was not possible, and digital documents could only be read if recipients had installed the same non-standard font (Neil Patel, pers. comm., 19 August 2025). Even so, the fonts marked an important breakthrough, making it possible to typeset Adlam on personal computers and produce printed materials more easily.

The Barrys purchased computers, loaded them with the new fonts, and in 2008 Abdoulaye brought the machines back to Guinea, demonstrating how they could be used for books and newspapers (Barry et al.). Ibrahima also demonstrated the digital tools to the Fulani community in New York; New York steadily became a major learning centre for Adlam and the eventual headquarters for the Winden Jangen non-profit (Barry et al.; Hasson). This digital phase was a milestone, but it also highlighted the next challenge: “People immediately asked us how they could use it on their cell phones”, Abdoulaye noted (Barry et al.).

The brothers began making inquiries about getting Adlam encoded in Unicode but received no response (Barry et al.).

Meanwhile, Ibrahima turned to improving the letterforms themselves. He believed it was vital that Adlam be not only functional but beautiful (Barry and Barry). Dissatisfied with shapes in the early fonts, he enrolled in calligraphy classes at Portland Community College in 2012 with instructor Rebecca Wild. Impressed by his work, Wild helped him secure a scholarship to a local calligraphy conference that same year. There Ibrahima met calligrapher Randall M. Hasson, who invited him to co-present at a Colorado Springs conference in 2013. Their presentation met with a standing ovation and introduced Adlam to a larger lettering audience (Bach).

Through these networks the brothers were introduced to Michael Everson, an experienced Unicode proposal author with a long track record of successful submissions (Barry et al.). With funding from SEI, Everson submitted the first formal Unicode proposal for Adlam in November 2013. The Unicode Technical Committee approved the script in October 2014 after revisions and further review, and two years later it appeared in Unicode 9.0.

Although Adlam was added to Unicode in 2016, the brothers were initially discouraged to find that the script could not yet be typed on most devices (Barry et al.; Waddell). They had assumed Unicode inclusion was the final step, not realizing it marked the beginning of a longer process requiring vendors to integrate fonts, keyboards, and shaping support. Journalist Kaveh Waddell underscored this uncertainty in a 2016 article in *The Atlantic*, which both introduced Adlam to a broad new readership and highlighted the obstacles ahead.¹⁰ Among the article’s readers was Craig Cornelius, a software internationalization engineer at Google who was already involved in supporting other minority scripts. Cornelius reached out to the Barrys, championed Adlam within Google, and helped secure its integration into the Noto font project and Android, along with a functioning keyboard (Barry et al.).

During the same period, Adlam found allies at Microsoft. Program Manager Andrew Glass, a long-time participant in Unicode meetings, became an early advocate after meeting the brothers at a UTC meeting (Bach). Glass prototyped Adlam keyboard layouts during Microsoft’s annual employee hackathon. These efforts, combined with internal advocacy led by Program Manager Judy Safran-Aasen, led Microsoft to enlist the Maine-based design studio JamraPatel to add Adlam to the Ebrima font, which already supported several African scripts. With the 2019 update to Windows 10, Adlam became available across Windows and Office products (Bach).

These industry partnerships built on, rather than replaced, the grassroots infrastructure already in place. Winden Jangen continued to expand across West Africa and the diaspora, so that by the time Google and Microsoft added Adlam support, a transnational network of teachers and

¹⁰ Waddell, in turn, reportedly learned of the script through Hasson’s earlier blog posts on Adlam, published in 2014 (Barry et al.). The original links to Hasson’s posts are now defunct but are archived via the Internet Archive’s WayBack Machine, starting with Part One available here: <https://web.archive.org/web/20160407035403/http://blog.randallmhasson.com/2014/04/the-adlam-story-how-alphabet-changes.html> (accessed 27 October 2025).

learners was already actively sustaining the script (Barry et al.). The convergence of grassroots activism and corporate backing marked a turning point: Adlam was no longer confined to handwriting or hacked fonts but had entered mainstream digital platforms.

3.2 POST-ENCODING REFINEMENT AND EXPANSION

While the previous section traced Adlam’s trajectory from grassroots invention to recognition by major technology companies, this section turns to a parallel story. Alongside that public narrative ran a less visible process within Unicode itself, where the script was scrutinized, revised, and debated. The timelines overlap: as Adlam was gaining broader visibility and support, Unicode reviewers were still questioning its readiness, and community feedback was prompting refinements.

Adlam’s initial proposal, submitted by Michael Everson in 2012, was judged to provide only limited evidence: handwritten pedagogical examples produced by the Barry brothers and a single clipping from an undated periodical (see excerpts from Everson, “Preliminary Proposal” in Figure 5). Reviewers in SEW noted that “more script samples are needed” that showed “consistency”, and believed the “current set of figures suggest[ed] that the script” was “still in development and ... not yet ready for encoding in a standard” (D. Anderson et al., “UTC #137”).

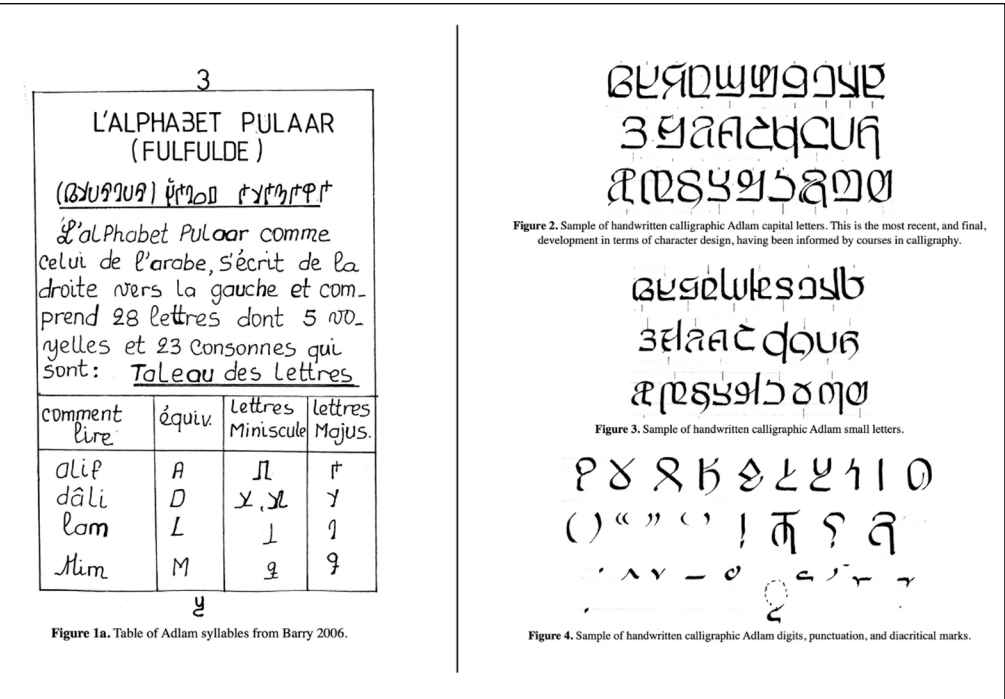


Figure 5 Sample evidence provided in initial Adlam proposal to Unicode in 2012.

In 2014 a revised proposal offered a more expansive case: population figures for Fula speakers, clarification of character behaviour, samples from classrooms and print materials, and eight letters of support from ministries, NGOs, and Fula diaspora organizations (excerpts in Figure 6). These letters attested to over two hundred Adlam learning centres in Conakry alone, with use extending even into smaller towns where people employed the script for everyday communication and record keeping (Everson, “Revised Proposal”).

Even then, Unicode’s questions persisted. Reviewers struggled to parse how the joining behaviour worked, which elements of the cursive style were optional (in part due to compromises made in the hacked Arabic font), and how upper-case and lower-case forms functioned. The proposal was accepted, but a reviewer noted that “No matter how many examples from the proposal we look at now, I have to say that there are other experts in UTC still having concerns about the script” (Umamaheswaran and Ksar).

These lines of questioning reflect the broader concerns Unicode standards makers often raise with “newer” scripts. Adlam did not struggle to demonstrate adoption, effectively sidestepping that half of the usual chicken-and-egg problem. The Barrys’ efforts to handwrite materials, commission non-standard fonts and keyboards, purchase devices for community use, and



Figure 13. A classroom in Nigeria where children are taught Adlam.



Figure 14. Some teachers of Adlam in Guinea.

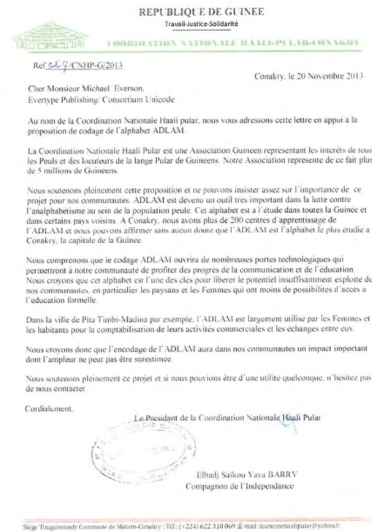


Figure 22a. Letter of support from the Coordination Nationale Haali Pular. This is the largest Fulani organization in the word, which co-ordinates the ethnic group relations across borders. Their headquarters are based in Guinea.

Figure 6 Sample evidence provided in revised Adlam proposal to Unicode in 2014.

build a lively non-profit organization around the script provided clear evidence of a user base. While Adlam was never adopted as the official script of any government administrative unit, several ministries nonetheless endorsed its teaching and recognized it as a valuable tool for reducing illiteracy among Fulani speakers (a notable reversal from the state suspicion that had once led to Ibrahima Barry's imprisonment) (Everson, "Revised Proposal").

The greater uncertainty lay in stability. Reviewers pointed to cursive behaviour, the treatment of upper- and lower-case forms, and the precise shapes of certain characters as features that did not yet appear fully settled. By the time of encoding, Adlam had been in circulation for twenty-five years, but it had also undergone repeated refinements during that period – enough to leave some experts questioning whether its forms were firmly fixed.

Unicode version 9.0, containing the Adlam block, was published in 2016 – two years after the proposal was accepted. By then, however, new refinements to the script's letterforms had already emerged. Teachers involved in literacy campaigns flagged ambiguities and legibility issues: capital and lower-case forms were sometimes hard to distinguish and some characters required dramatic revision – most often "supplementary letters" used in loanwords for non-Fula languages (see examples in Figure 7).

Name (Pronunciation)	Font	Capital	Small	Notes
BA (BA)		1E904	1E926	The proposed replacement eliminates the ascenders in the medial and final lowercase letters to prevent diacritic collision.
	PULAAR	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
	NOTO	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
	UNICODE	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
E (E)		1E909	1E92B	The design in the proposed replacement font is a stylistic variation of the Unicode font.
	PULAAR	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
	NOTO	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
	UNICODE	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
ZAL (ZA)		1E91F	1E941	The changes in the proposed font are quite dramatic. This new form make the letter more distinctive. This letter is a supplementary letter which is not part of the standard Pular orthography.
	PULAAR	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
	NOTO	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
	UNICODE	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	
	PROPOSED	𞤀𞤁𞤂 𞤀	𞤀𞤁𞤂 𞤀	

Figure 7 Excerpts from 2019 request to revise Adlam Unicode glyphs, comparing proposed shapes to existing fonts and Unicode Standard.

To address these concerns, the design studio JamraPatel worked closely with the Adlam community to adapt feedback into digital typefaces. For those involved, fonts were not only tools for typesetting but also carried pedagogical weight: they served as models for handwriting practice in a still nascent literacy movement (Neil Patel, pers. comm., 22 July 2025). With few alternatives available – minority script users often had access to only a handful of typefaces, compared to the thousands available for Latin – these designs had to be both versatile and maximally legible.

In 2019 a new proposal brought these lessons back to Unicode. Authored jointly by Neil Patel and Mark Jamra (JamraPatel), Craig Cornelius (Google), and the Barry brothers, it argued for replacing the Unicode chart font with updated designs, offering JamraPatel’s Ebrima and Kigelia fonts as new reference models (Patel, Jamra, Cornelius et al.). The rationale was straightforward: Unicode’s code charts often serve as the primary model for future font designers, so outdated glyphs risked perpetuating forms that the community had already moved beyond.

This update was possible, despite Unicode’s stability policy, because it did not alter the underlying character specifications that constitute the Standard. Unicode has long maintained its “characters, not glyphs” policy: it encodes abstract characters, not their visual forms. By that logic, changing the font used in the code charts is not considered a substantive revision. Yet in practice the charts exert far more authority than Unicode acknowledges. For many type designers – especially those approaching a script for the first time – the chart is the de facto reference model.¹¹ Outdated glyphs risk propagating into future fonts, even if the community itself has moved on. The Adlam case thus highlights a tension: while Unicode insists that the charts are not prescriptive, they nonetheless shape typographic practice in consequential ways.

Unicode experts agreed to replace the glyphs, noting that Adlam was “a relatively young script that does not have much digital data yet” and that it was preferable for the charts to reflect community-backed forms (D. Anderson et al., “UTC #159”). At the same time, they voiced unease about the potential for ongoing revisions, questioning how much further the script was expected to evolve.

Concerns about instability resurfaced in 2024, when another, more modest proposal requested further glyph refinements, this time emphasizing simplification and consistency (Patel, “Evolution of Adlam Characters”; Patel, Jamra, Barry et al.). This request was accompanied by another proposal to add character name annotations (Patel, “Proposal for Adlam Names List”). The user community had recently learned that name annotations could be added to the Unicode core specification, which would appear alongside the permanent character names. As simpler character names had become common in usage,¹² annotations would allow these common names to appear in the Unicode core specification without altering the permanent ones, bridging practice and standard. SEW reviewers pointed to precedent in scripts such as Balinese and Old South Arabian, and, along with the second glyph revision, approved the request (Kučera et al.). See iterations of the published Adlam code chart in Figure 8 and annotations to the names list in Figure 9.

More recently, another proposal has appeared on the docket for Unicode consideration to extend the Adlam block by thirty-four new characters. These additions are intended to support transcription of languages beyond Fula in multilingual regions where Adlam is taught, along with several new punctuation marks. The rationale is explicitly tied to digital practice:

As the first display face ADLaM Display [recently published in Microsoft systems] opened the door to considering how the script can better represent Fulani visual identity. To this end the community would like to use their own punctuation rather than symbols adopted from other scripts. (Patel et al., “Proposal for Adlam Extensions”)

¹¹ See the discussion of “privileged glyphs” in Haralambous 119.

¹² The proposal explained that “within Adlam education circles the long names for letters are not commonly used. This change was made to make it easier for children to learn and recite their alphabet. Adding informative short names facilitates collaboration with Adlam community members as they are more likely to know the short names” (Patel, “Proposal for Adlam Names List” 1).

Adlam					
Unicode 9.0 (2016)					
1E90	1E91	1E92	1E93	1E94	1E95
0	𐎀	𐎁	𐎂	𐎃	𐎄
1	𐎅	𐎆	𐎇	𐎈	𐎉
2	𐎊	𐎋	𐎌	𐎍	𐎎
3	𐎏	𐎐	𐎑	𐎒	𐎓
4	𐎔	𐎕	𐎖	𐎗	𐎘
5	𐎙	𐎚	𐎛	𐎜	𐎝
6	𐎞	𐎟	𐎠	𐎡	𐎢
7	𐎣	𐎤	𐎥	𐎦	𐎧
8	𐎨	𐎩	𐎪	𐎫	𐎬
9	𐎭	𐎮	𐎯	𐎰	𐎱
A	𐎲	𐎳	𐎴	𐎵	𐎶
B	𐎷	𐎸	𐎹	𐎺	𐎻
C	𐎼	𐎽	𐎾	𐎿	𐏀
D	𐏁	𐏂	𐏃	𐏄	𐏅
E	𐏆	𐏇	𐏈	𐏉	𐏊
F	𐏋	𐏌	𐏍	𐏎	𐏏
Unicode 13.0 (2020)					
1E90	1E91	1E92	1E93	1E94	1E95
0	𐎀	𐎁	𐎂	𐎃	𐎄
1	𐎅	𐎆	𐎇	𐎈	𐎉
2	𐎊	𐎋	𐎌	𐎍	𐎎
3	𐎏	𐎐	𐎑	𐎒	𐎓
4	𐎔	𐎕	𐎖	𐎗	𐎘
5	𐎙	𐎚	𐎛	𐎜	𐎝
6	𐎞	𐎟	𐎠	𐎡	𐎢
7	𐎣	𐎤	𐎥	𐎦	𐎧
8	𐎨	𐎩	𐎪	𐎫	𐎬
9	𐎭	𐎮	𐎯	𐎰	𐎱
A	𐎲	𐎳	𐎴	𐎵	𐎶
B	𐎷	𐎸	𐎹	𐎺	𐎻
C	𐎼	𐎽	𐎾	𐎿	𐏀
D	𐏁	𐏂	𐏃	𐏄	𐏅
E	𐏆	𐏇	𐏈	𐏉	𐏊
F	𐏋	𐏌	𐏍	𐏎	𐏏
Unicode 17.0 (2025)*					
1E90	1E91	1E92	1E93	1E94	1E95
0	𐎀	𐎁	𐎂	𐎃	𐎄
1	𐎅	𐎆	𐎇	𐎈	𐎉
2	𐎊	𐎋	𐎌	𐎍	𐎎
3	𐎏	𐎐	𐎑	𐎒	𐎓
4	𐎔	𐎕	𐎖	𐎗	𐎘
5	𐎙	𐎚	𐎛	𐎜	𐎝
6	𐎞	𐎟	𐎠	𐎡	𐎢
7	𐎣	𐎤	𐎥	𐎦	𐎧
8	𐎨	𐎩	𐎪	𐎫	𐎬
9	𐎭	𐎮	𐎯	𐎰	𐎱
A	𐎲	𐎳	𐎴	𐎵	𐎶
B	𐎷	𐎸	𐎹	𐎺	𐎻
C	𐎼	𐎽	𐎾	𐎿	𐏀
D	𐏁	𐏂	𐏃	𐏄	𐏅
E	𐏆	𐏇	𐏈	𐏉	𐏊
F	𐏋	𐏌	𐏍	𐏎	𐏏

Figure 8 The Adlam code chart as published in [Versions 9.0, 13.0, and 17.0](#) of the Unicode Standard (reproduced with permission from the Unicode Consortium).

Whether these new characters are approved remains to be seen.

From the vantage point of Unicode reviewers, Adlam can still appear mercurial. Continued requests for glyph updates, name annotations, and extensions arrive at the SEW docket, reinforcing a perception that the script is unsettled. For those unfamiliar with the script’s long history, this ongoing stream of revisions can reinforce the impression that Adlam – and perhaps other newer scripts – is a “moving target” whose stability remains in doubt.

Yet with respect to its fundamental structure, this impression is misleading. The core repertoire and script grammar first encoded in Unicode have held firm, despite earlier fears to the contrary. What has changed, primarily, are letterforms and typographic conventions, shaped by the unusually high stakes of minority script digitization. So few font options exist, particularly on mobile devices, that the default designs take on disproportionate authority, guiding not only digital use but handwriting practices as well.

In addition to these visual refinements, the repertoire itself is not shifting but gradually expanding. Adlam is attracting interest from Hausa, Serer, Kanuri, and Mande speakers who regularly interact with Fulani communities. Unicode encoding played a key role in signalling that Adlam was a legitimate, durable system and in making it accessible to other language communities. People could now encounter Adlam on Facebook and see it working on phones and computers. Some had previously resisted learning it, seeing little value in a script confined to handwriting, but Unicode inclusion “inform[ed] people that the script is real and worth considering” (Neil Patel, pers. comm., 7 August 2025).

In this sense, Adlam illustrates a feedback loop between technical and script standardization. Years of grassroots practice laid the foundation for Unicode recognition, but once encoded, the script’s expanded visibility generated new demands – refinements to glyphs, name annotations,

and proposed extensions – that continue to reshape it. This dynamic echoes the “chicken-and-egg” paradox outlined earlier: adoption and stability are not sequential thresholds but mutually reinforcing processes. For newer neographies in particular, Unicode does not simply *reflect* stability but helps to constitute it, however unintentionally, by participating in the ongoing process of script formation.

1E900	Adlam	1E95F
Capital letters		
1E900	𞤀 ADLAM CAPITAL LETTER ALIF = a	1E929 𞤩 ADLAM SMALL LETTER BHE
1E901	𞤁 ADLAM CAPITAL LETTER DAALI = da	1E92A 𞤪 ADLAM SMALL LETTER RA
1E902	𞤂 ADLAM CAPITAL LETTER LAAM = la	1E92B 𞤫 ADLAM SMALL LETTER E
1E903	𞤃 ADLAM CAPITAL LETTER MIIM = ma	1E92C 𞤬 ADLAM SMALL LETTER FA
1E904	𞤄 ADLAM CAPITAL LETTER BA	1E92D 𞤭 ADLAM SMALL LETTER I
1E905	𞤅 ADLAM CAPITAL LETTER SINNYIYHE = sa	1E92E 𞤮 ADLAM SMALL LETTER O
1E906	𞤆 ADLAM CAPITAL LETTER PE = pa	1E92F 𞤯 ADLAM SMALL LETTER DHA
1E907	𞤇 ADLAM CAPITAL LETTER BHE = bha	1E930 𞤰 ADLAM SMALL LETTER YHE
1E908	𞤈 ADLAM CAPITAL LETTER RA	1E931 𞤱 ADLAM SMALL LETTER WAW
1E909	𞤉 ADLAM CAPITAL LETTER E	1E932 𞤲 ADLAM SMALL LETTER NUN
1E90A	𞤊 ADLAM CAPITAL LETTER FA	1E933 𞤳 ADLAM SMALL LETTER KAF
1E90B	𞤋 ADLAM CAPITAL LETTER I	1E934 𞤴 ADLAM SMALL LETTER YA
1E90C	𞤌 ADLAM CAPITAL LETTER O	1E935 𞤵 ADLAM SMALL LETTER U
1E90D	𞤍 ADLAM CAPITAL LETTER DHA	1E936 𞤶 ADLAM SMALL LETTER MIIM
1E90E	𞤎 ADLAM CAPITAL LETTER YHE = yha	1E937 𞤷 ADLAM SMALL LETTER CHI
1E90F	𞤏 ADLAM CAPITAL LETTER WAW = wa	1E938 𞤸 ADLAM SMALL LETTER HA
1E910	𞤐 ADLAM CAPITAL LETTER NUN = na	1E939 𞤹 ADLAM SMALL LETTER QAAF
1E911	𞤑 ADLAM CAPITAL LETTER KAF = ka	1E93A 𞤺 ADLAM SMALL LETTER GA
1E912	𞤒 ADLAM CAPITAL LETTER YA	1E93B 𞤻 ADLAM SMALL LETTER NYA
1E913	𞤓 ADLAM CAPITAL LETTER U	1E93C 𞤼 ADLAM SMALL LETTER TU
1E914	𞤔 ADLAM CAPITAL LETTER MIIM = ja	1E93D 𞤽 ADLAM SMALL LETTER NHA
1E915	𞤕 ADLAM CAPITAL LETTER CHI = tcha	Supplementary small letters
1E916	𞤖 ADLAM CAPITAL LETTER HA	1E93E 𞤾 ADLAM SMALL LETTER VA
1E917	𞤗 ADLAM CAPITAL LETTER QAAF = qa	1E93F 𞤿 ADLAM SMALL LETTER KHA
1E918	𞤘 ADLAM CAPITAL LETTER GA	1E940 𞥀 ADLAM SMALL LETTER GBE
1E919	𞤙 ADLAM CAPITAL LETTER NYA	1E941 𞥁 ADLAM SMALL LETTER ZAL
1E91A	𞤚 ADLAM CAPITAL LETTER TU = ta	1E942 𞥂 ADLAM SMALL LETTER KPO
1E91B	𞤛 ADLAM CAPITAL LETTER NHA	1E943 𞥃 ADLAM SMALL LETTER SHA
Supplementary capital letters		
1E91C	𞤜 ADLAM CAPITAL LETTER VA	Diacritical marks
1E91D	𞤝 ADLAM CAPITAL LETTER KHA	1E944 𞥄 ADLAM ALIF LENGTHENER
1E91E	𞤞 ADLAM CAPITAL LETTER GBE = gba	1E945 𞥅 ADLAM VOWEL LENGTHENER
1E91F	𞤟 ADLAM CAPITAL LETTER ZAL = za	1E946 𞥆 ADLAM GEMINATION MARK
1E920	𞤠 ADLAM CAPITAL LETTER KPO = kpa	1E947 𞥇 ADLAM HAMZA
1E921	𞤡 ADLAM CAPITAL LETTER SHA	1E948 𞥈 ADLAM CONSONANT MODIFIER
Small letters		
1E922	𞤢 ADLAM SMALL LETTER ALIF	1E949 𞥉 ADLAM GEMINATE CONSONANT MODIFIER
1E923	𞤣 ADLAM SMALL LETTER DAALI	1E94A 𞥊 ADLAM NUKTA
1E924	𞤤 ADLAM SMALL LETTER LAAM	• called hoortobbhere when drawn above and lestobbhere when drawn below the base character
1E925	𞤥 ADLAM SMALL LETTER MIIM	Modifier letter
1E926	𞤦 ADLAM SMALL LETTER BA	1E94B 𞥋 ADLAM NASALIZATION MARK
1E927	𞤧 ADLAM SMALL LETTER SINNYIYHE	• indicates preceding nasal consonant prenasalizes subsequent consonant letter
1E928	𞤨 ADLAM SMALL LETTER PE	Digits
		1E950 𞥌 ADLAM DIGIT ZERO
		1E951 𞥍 ADLAM DIGIT ONE
		1E952 𞥎 ADLAM DIGIT TWO
		1E953 𞥏 ADLAM DIGIT THREE
		1E954 𞥐 ADLAM DIGIT FOUR
		1E955 𞥑 ADLAM DIGIT FIVE
		1E956 𞥒 ADLAM DIGIT SIX
		1E957 𞥓 ADLAM DIGIT SEVEN
		1E958 𞥔 ADLAM DIGIT EIGHT
		1E959 𞥕 ADLAM DIGIT NINE
		Punctuation
		1E95E 𞥖 ADLAM INITIAL EXCLAMATION MARK
		1E95F 𞥗 ADLAM INITIAL QUESTION MARK

1788

The Unicode Standard, Version 17.0 BETA REVIEW DRAFT, Copyright © 1991-2025 Unicode, Inc. All rights reserved.

1788

The Unicode Standard, Version 17.0 BETA REVIEW DRAFT, Copyright © 1991-2025 Unicode, Inc. All rights reserved.

Figure 9 Forthcoming update to Adlam names list in [Unicode 17.0](#), with name annotations highlighted in yellow (reproduced with permission from the Unicode Consortium).

4. NEW DIRECTIONS FOR NEOGRAPHY RESEARCH

Adlam represents, in many respects, an exceptional case. The script’s inventors combined talent and ingenuity with extraordinary drive over the course of many decades. They were able to cultivate wide networks, collaborating both across international Fulani communities and with sympathetic external actors. The Fulani themselves were a sizeable, mobile, and well-

organized community that provided a fertile ground for adoption. These same characteristics might not hold true for other neographies or allow for straightforward comparison.

At the same time, the Adlam story also exemplifies broader tensions that confront newly invented scripts. These include the enormous effort required to jump-start adoption, the burden of proving script stability, and the precarious economics of sustaining minority script support.

For standards bodies and technologists, the central dilemma is how responsive to be. Should Unicode reviewers postpone inclusion until a script demonstrates stability? If so, by what criteria can stability be measured, and what are the consequences of withholding recognition in the meantime? Should evidence of institutional endorsement be required, even though many neographies emerge outside formal authority structures? Should proposals await the development of established typographic conventions, and if so, who will assume the risks of producing high-quality fonts for scripts still deemed experimental? The Adlam experience demonstrates that Unicode and font designers have sometimes accommodated gradual evolution in a script, but that very accommodation has made some actors more hesitant to repeat the experiment.

As this article has shown, many of the recurrent features of neographies complicate these judgements. They often arise in activist contexts marked by limited, or even antagonistic, state support; they may blur the boundary between functional writing and symbolic artwork; and they frequently serve communities with little global visibility. Setting rigid thresholds risks excluding viable scripts. At the same time, discretionary inclusion positions encoders as arbiters of legitimacy, effectively licensing or foreclosing the spread of particular writing systems. This is, in short, a bind. Neographies place all parties – community advocates, encoders, designers, and vendors – into unsettled terrain.

This case suggests several avenues for further investigation. Researchers might examine how other neographies have navigated the challenge of establishing themselves, the specific role of digital infrastructures in those efforts, and the extent to which geopolitical contexts shape outcomes. The Unicode Document Registry constitutes a rich yet under-utilized archive for such study. Future work should also be attentive to the wider constellation of external stakeholders – type designers, software developers, journalists, intermediaries such as the Script Encoding Initiative, alongside Unicode reviewers – whose interventions have the power to shape the fate of neographies.

There is also value in widening the comparative lens. Scripts undergoing reform or revitalization might face similar tensions to neographies. Are there more productive conceptual categories than “neography” for thinking across unsettled or evolving writing systems? Moreover, the long-term consequences of adoption remain insufficiently understood, even simply from the standpoint of digital inclusion. Full digital recognition is rarely swift: even in the most favourable cases of Unicode inclusion, years can elapse before a script becomes fully supported on mobile platforms, and it can be longer still before such devices are accessible in low-income regions. Communities might thus inhabit a prolonged liminal state, with significant costs for participation in digital life.

For practitioners, one challenge is to devise alternative tools that loosen the dependency on current technical bottlenecks. These might include more accessible mechanisms for deploying experimental scripts on mobile devices, enhancements to the Unicode Private Use Area to allow flexibility in shaping and directionality, or initiatives to educate major vendors about the distinctive needs of neographies.

This article has highlighted neographies as an important site of inquiry for scholars, technologists, and script communities alike. The case of Adlam highlights both the possibilities and the limits of digital recognition, and offers a view into the unsettled processes that, for now, shape how new scripts enter the infrastructures of global communication.

ACKNOWLEDGEMENTS

This piece benefited tremendously from the generous feedback of Raja Adal, Jonathan Breit, Tim Brookes, Peter Constable, Annie Ferguson, Helena Kansa, Jan Kučera, Nunzio Mazzaferro, Neil Patel, Logan Simpson, Pippa Steele, Oreen Yousuf, Isabelle Zaugg, and the members of the VIEWS Project at the University of Cambridge. All remaining errors are my own.

This research was funded in part by the Andrew W. Mellon Foundation and the Internet Society Foundation.

Andrew W. Mellon Foundation

<https://doi.org/10.13039/1000000873>

Internet Society Foundation

<https://doi.org/10.13039/100026870>

AUTHOR AFFILIATIONS

Anushah Hossain  orcid.org/0000-0002-5716-292X

University of California, Berkeley, US

REFERENCES

- Adal, Raja. "Big Tech, the Big State, and the Battle for Encoding All of the World's Languages." *Journal of Asian Studies*, vol. 85, no. 4, forthcoming October 2026.
- Anderson, Benedict. *Imagined Communities: Reflections on the Origin and Spread of Nationalism*. Rev. edn. Verso, 2006.
- Anderson, Debbie. "Draft Template for New Script Proposal." Unicode Document Registry, L2/23-105, 17 April 2023. <https://www.unicode.org/L2/L2023/23105-new-script-template-april2023.pdf>. Accessed 2 October 2025.
- Anderson, Debbie, Ken Whistler, Rick McGowan, et al. "Recommendations to UTC #137 November 2013 on Script Proposals." Unicode Document Registry, L2/13-210, 31 October 2013. <https://www.unicode.org/L2/L2013/13210-script-rec.pdf>. Accessed 2 October 2025.
- Anderson, Debbie, Ken Whistler, Roozbeh Pournader, et al. "Recommendations to UTC #159 April–May 2019 on Script Proposals." Unicode Document Registry, L2/19-173, 29 April 2019. <https://www.unicode.org/L2/L2019/19173-script-adhoc-recs.pdf>. Accessed 2 October 2025.
- Bach, Deborah. "How a New Alphabet is Helping an Ancient People Write its Own Future." *Microsoft Story Labs*, 29 July 2019. <https://web.archive.org/web/20190811043649/https://news.microsoft.com/stories/people/adlam.html>. Accessed 2 October 2025.
- Barry, Abdoulaye, et al. "Inventing the Adlam Script / Designing Type for a Society in Flux." Letterform Lecture, 19 March 2019. <https://www.youtube.com/watch?v=PXN4aciv19U>. Accessed 2 October 2025.
- Barry, Abdoulaye, and Ibrahima Barry. "The ADLaM Alphabet for Our People." Talks at Google. <https://www.youtube.com/watch?v=NPQ7ufjN2i8>. Accessed 2 October 2025.
- Becker, Joseph D. "Unicode 88." 29 August 1988. <https://unicode.org/history/unicode88.pdf>. Accessed 2 October 2025.
- Brandt, Carmen. "Script as a Potential Demarcator and Stabilizer of Languages in South Asia." In *Language Endangerment and Preservation in South Asia*, Language Documentation & Conservation Special Publication, no. 7, January 2014, pp. 78–99.
- Brookes, Tim. "The Red List of Endangered and Emerging Scripts." Unpublished spreadsheet shared via the Endangered Alphabets Project, August 2024.
- Choksi, Nishaant. "From Language to Script: Graphic Practice and the Politics of Authority in Santali–Language Print Media, Eastern India." *Modern Asian Studies*, vol. 51, no. 5, 2017, pp. 1519–60. DOI: <https://doi.org/10.1017/S0026749X16000470>
- Choksi, Nishaant. *Graphic Politics in Eastern India: Script and the Quest for Autonomy*. Bloomsbury Academic, 2022. DOI: <https://doi.org/10.5040/9781350159617>
- Dalby, David. "A Survey of the Indigenous Scripts of Liberia and Sierra Leone: Vai, Mende, Loma, Kpelle and Bassa." *African Language Studies*, vol. 8, 1967, pp. 1–51.
- Dalby, David. "Further Indigenous Scripts of West Africa: Manding Wolof and Fula Alphabets and Yoruba Holy Writing." *African Language Studies*, vol. 10, 1969, pp. 161–81.
- Dalby, David. "The Indigenous Scripts of West Africa and Surinam: Their Inspiration and Design." *African Language Studies*, vol. 9, 1968, pp. 156–97.
- Davis, Mark. "Moving to Unicode 5.1." Official Google blog, 5 May 2008. <https://googleblog.blogspot.com/2008/05/moving-to-unicode-51.html>. Accessed 2 October 2025.
- Everson, Michael. "Preliminary Proposal for Encoding the Adlam Script in the SMP of the UCS." Unicode Document Registry, L2/13-191, 28 October 2012. <https://www.unicode.org/L2/L2013/13191-n4488-adlam.pdf>. Accessed 2 October 2025.

- Everson, Michael. "Revised Proposal for Encoding the Adlam Script in the SMP of the UCS." Unicode Document Registry, L2/14-219R, 28 October 2014. <https://www.unicode.org/L2/L2014/14219r-n4628-adlam.pdf>. Accessed 2 October 2025.
- Field, Dylan. "Localization, Languages, and Listening." Figma blog, 29 March 2022. <https://www.figma.com/blog/expanding-figmas-international-presence/>. Accessed 2 October 2025.
- Gnanadesikan, Amalia E. *The Writing Revolution: Cuneiform to the Internet*. Wiley, 2025. DOI: <https://doi.org/10.1002/9781394218226>
- Hales, Kevin. *The Moving Finger: A Rhetorical, Grammatological and Afrinographic Exploration of Nsibidi in Nigeria and Cameroon*. 2015. PhD thesis, Ohio University. https://etd.ohiolink.edu/acprod/odb_etd_etd/r/1501/10?clear=10&p10_accession_num=ohiou1431071905. Accessed 2 October 2025.
- Haralambous, Yannis. "Unicode et typographie: un amour impossible." *Document numérique*, vol. 6, nos. 3-4, 2002, pp. 105-37. DOI: <https://doi.org/10.3166/dn.6.3-4.105-137>
- Hasson, Randall M. *The ADLaM Story – How Alphabet Changes Culture – Part One*. 2 April 2014. <https://web.archive.org/web/20160407035403/http://blog.randallmhasson.com/2014/04/the-adlam-story-how-alphabet-changes.html>. Accessed 2 October 2025.
- Hasson, Randall M. *The ADLaM Story – How Alphabet Changes Culture – Part Three*. 11 April 2014. https://web.archive.org/web/20160407035407/http://blog.randallmhasson.com/2014/04/the-adlam-story-how-alphabet-changes_11.html. Accessed 2 October 2025.
- Hasson, Randall M. *The ADLaM Story – How Alphabet Changes Culture – Part Two*. 6 April 2014. https://web.archive.org/web/20160407035419/http://blog.randallmhasson.com/2014/04/the-adlam-story-how-alphabet-changes_6.html. Accessed 2 October 2025.
- Holloway, Steph. "Tikamuli." *ScriptSource*, 2025. https://www.scriptsources.org/cms/scripts/page.php?item_id=script_detail&key=Qa19. Accessed 2 October 2025.
- Hossain, Anushah. "ISCII Imperialism: The Legacy of a Devanagari-Centric Character Code." *Journal of Asian Studies*, vol. 85, no. 4, forthcoming October 2026.
- Hossain, Anushah. "Text Standards for the 'Rest of World': The Making of the Unicode Standard and the OpenType Format." *IEEE Annals of the History of Computing*, vol. 46, no. 1, 2024, pp. 20-33. DOI: <https://doi.org/10.1109/MAHC.2024.3351948>
- Hossain, Anushah, and Ahad Bashir. "Encoding Expectations: How Long Does It Really Take?" *UTW* 2024. <https://www.youtube.com/watch?v=NISWD708o5s>. Accessed 2 October 2025.
- Imamura, Masao. "Orthographic Plurality: Case Studies from Mainland Southeast Asia's Borderlands." Paper given at the 7th Conference of the Asian Borderlands Research Network, Seoul, South Korea. https://www.iias.asia/sites/default/files/event/2021-11/Accepted_Proposals_ABRN2022.pdf. Accessed 2 October 2025.
- Indigenous Languages: Zero to Digital*. Translation Commons, December 2019. <https://translationcommons.org/zero-to-digital-a-guide-to-bring-your-language-online/>. Accessed 2 October 2025.
- JamraPatel. "Tappirgal Adlam." Apps on Google Play, 23 July 2025. https://play.google.com/store/apps/details?id=com.jamra_patel.tappirgaladlam&hl=en. Accessed 2 October 2025.
- Jordan, D. K. "Languages Left Behind: Keeping Taiwanese off the World Wide Web." *Language Problems & Language Planning*, vol. 26, no. 2, 2002, pp. 111-27. DOI: <https://doi.org/10.1075/lplp.26.2.02jor>
- Kelly, Piers. "The Art of Not Being Legible." *Terrain. Anthropologie & Sciences Humaines*, no. 70, 2018. DOI: <https://doi.org/10.4000/terrain.17103>
- Kelly, Piers. "The Invention, Transmission and Evolution of Writing: Insights from the New Scripts of West Africa." In *Paths into Script Formation in the Ancient Mediterranean*, edited by Silvia Ferrara and Miguel Valério. Edizioni Quasar, 2019, pp. 189-209.
- Kostelnik, Jannis. "Technological Adaptation and the Role of Social Media in the Proliferation of the Newly Created Minim Script from Burkina Faso." *Ahyu: A Journal of Language and Literature*, vol. 6, 2023, pp. 71-87. DOI: <https://doi.org/10.56666/ahyu.v6i.163>
- Kučera, Jan. *A Study of the Scripts of Tamil Epigraphy and Their Development*. 2023. PhD thesis, Charles University.
- Kučera, Jan, Debbie Anderson, Manish Goregaokar, Ken Whistler, Roozbeh Pournader, and Peter Constable. "Recommendations to UTC #182 (January 2025) on Script Proposals." Unicode Document Registry, L2/25-010, 16 January 2025. <https://www.unicode.org/L2/L2025/25010-script-wg-report.pdf>. Accessed 2 October 2025.
- Mafundikwa, Saki. *African Alphabets: The Story of Writing in Afrika*. Mark Batty, 2004.
- Mandal, Biswajit. "Revised Proposal to Encode the Kurukh Banna in UCS." Unicode Document Registry, L2/24-101, 29 February 2024. <https://www.unicode.org/L2/L2024/24101-kurukh-banna.pdf>. Accessed 2 October 2025.
- Mazzaferro, Annunziato. "The West African Forest of Symbols: Reflections on the Representation of Minority Writing Systems." *Progetto Grafico*, no. 41, forthcoming 2025.
- McLuhan, Marshall. *The Gutenberg Galaxy: The Making of Typographic Man*. University of Toronto Press, 1962. <http://archive.org/details/gutenberggalaxym0000mclu>. Accessed 2 October 2025.

- Meletis, Dimitrios. *The Nature of Writing. A Theory of Grapholinguistics*. Fluxus Editions, 2020. DOI: <https://doi.org/10.36824/2020-meletis>
- Microsoft Unlocked. "Can an Alphabet Save a Culture?" 3 July 2023. <https://unlocked.microsoft.com/adlam-can-an-alphabet-save-a-culture/>. Accessed 2 October 2025.
- Mitchell, Ben, and Sovichet Tep. "Writing Systems of Southeast Asia." 2025. <https://ohbendy.github.io/Scripts-of-SEAsia/>. Accessed 2 October 2025.
- Pasch, Helma. "Competing Scripts: The Introduction of the Roman Alphabet in Africa." *International Journal of the Sociology of Language*, vol. 2008, no. 191, 2008. DOI: <https://doi.org/10.1515/IJSL.2008.025>
- Patel, Neil. "Evolution of Adlam Characters Since Encoding (2016–2024)." Unicode Technical Notes, UTN #58, 21 August 2024. <https://www.unicode.org/notes/tn58/UTN58-Evolution-of-Adlam-Characters-Since-Encoding-2.pdf>. Accessed 2 October 2025.
- Patel, Neil. "Proposal for Adlam Names List Modifications." Unicode Document Registry, L2/24-204, 28 August 2024. <https://www.unicode.org/L2/L2024/24204-adlam-nameslist.pdf>. Accessed 2 October 2025.
- Patel, Neil, Ibrahima Barry, et al. "Proposal for Adlam Extensions." 18 June 2025. *Forthcoming in the Unicode Document Registry*. Internal draft, accessed 2 October 2025.
- Patel, Neil, Mark Jamra, Craig Cornelius, et al. "Replacement of Adlam Reference Font in Codesheet to Updated Design." Unicode Document Registry, L2/19-119R, 19 April 2019. <https://www.unicode.org/L2/L2019/19119r-adlam-font-repl.pdf>. Accessed 2 October 2025.
- Patel, Neil, Mark Jamra, Ibrahima Barry, et al. "Replacement of Adlam Reference Font in Codesheet to Updated Design." Unicode Document Registry, L2/24-205, 8 August 2024. <https://www.unicode.org/L2/L2024/24205-adlam-chart-update.pdf>. Accessed 2 October 2025.
- Sanders, Nathan. "A Primer on Constructed Languages." In *Language Invention in Linguistics Pedagogy*, edited by Jeffrey Punske et al. Oxford University Press, 2020, pp. 6–26. DOI: <https://doi.org/10.1093/oso/9780198829874.003.0002>
- SEI Team. "Interview with Jan Kučera: On the Promises and Perils of Script Encoding." Script Encoding Initiative blog, 16 June 2025. <https://sei.berkeley.edu/2025/06/16/interview-with-jan-kucera-on-the-promises-and-perils-of-script-encoding/>. Accessed 2 October 2025.
- SEI Team. "Interview with Oreen Yousuf: On Engaging Communities through Online Fieldwork." Script Encoding Initiative blog, 1 July 2025. <https://sei.berkeley.edu/2025/07/01/oreen-yousuf-interview/>. Accessed 2 October 2025.
- Simpson, Logan. "From Icons to Identities: Analysing Visual Cultural Elements in Emerging Scripts." *Visible Language*, vol. 58, no. 2, 2024, pp. 42–81. DOI: <https://doi.org/10.34314/37ccs174>
- Slack. "Manage Your Language Preferences." *Slack Help Center*, <https://slack.com/help/articles/215058658-Manage-your-language-preferences>. Accessed 31 August 2025.
- Steele, Philippa M. "Thoughts on the Visual-Cultural Fit of Writing Systems, and Its Intersections with Linguistic Fit." In *Grapholinguistics in the 21st Century—From Graphemes to Knowledge*, edited by Yannis Haralambous and S. Elti di Rodeano. Fluxus Editions, forthcoming.
- Tuchscherer, Konrad, and P. E. H. Hair. "Cherokee and West Africa: Examining the Origins of the Vai Script." *History in Africa*, vol. 29, 2002, pp. 427–86. DOI: <https://doi.org/10.2307/3172173>
- Umamaheswaran, V. S., and Mike Ksar. "Unconfirmed Minutes of WG 2 Meeting 63." Unicode Document Registry, L2/16-052, 1 October 2015. <https://www.unicode.org/L2/L2016/16052-n4603.pdf>. Accessed 2 October 2025.
- Unicode. "Submitting Character Proposals." <https://www.unicode.org/pending/proposals.html>. Accessed 18 July 2025.
- Unicode. "Supported Scripts." <https://www.unicode.org/standard/supported.html>. Accessed 18 July 2025.
- Unicode, Inc. *The Unicode Standard, Version 9.0*. 19 July 2016.
- Unicode, Inc. *The Unicode Standard, Version 13.0*. 10 March 2020.
- Unicode, Inc. *The Unicode Standard, Version 17.0 BETA REVIEW DRAFT*. 20 May 2025.
- Unseth, Peter. "Invention of Scripts in West Africa for Ethnic Revitalization." In *Handbook of Language and Ethnic Identity: The Success–Failure Continuum in Language and Ethnic Identity Efforts*, edited by Joshua Fishman and Ofelia Garcia, vol. 2. Oxford University Press, 2011, pp. 23–32.
- Unseth, Peter. "Sociolinguistic Parallels between Choosing Scripts and Languages." *Written Language & Literacy*, vol. 8, no. 1, 2005, pp. 19–42. DOI: <https://doi.org/10.1075/wll.8.1.02uns>
- Waddell, Kaveh. "The Alphabet That Will Save a People From Disappearing." *The Atlantic*, 16 November 2016. <https://www.theatlantic.com/technology/archive/2016/11/the-alphabet-that-will-save-a-people-from-disappearing/506987/>. Accessed 2 October 2025.
- Wyrod, Christopher. "A Social Orthography of Identity: The N'ko Literacy Movement in West Africa." *International Journal of the Sociology of Language*, no. 192, 2008. DOI: <https://doi.org/10.1515/IJSL.2008.033>
- Yeo, Adam, and Fang Cao. "Study on the Application of the Bété Script in Modern Graphic Design." *Art and Design Review*, vol. 9, no. 2, 2021, pp. 156–79. DOI: <https://doi.org/10.4236/adr.2021.92014>

Yousuf, Oreen. "Proposal to Encode the Minim Dag Noore Script in the UCS." Unicode Document Registry, L2/25-136, 28 March 2025. <https://www.unicode.org/L2/L2025/25136-minim-dag-noore.pdf>. Accessed 2 October 2025.

Yousuf, Oreen, et al. "2023 Update on African Scripts." Unicode Document Registry, L2/23-203, 8 September 2023. <https://www.unicode.org/L2/L2023/23203-update-african-scripts.pdf>. Accessed 2 October 2025.

Hossain
Modern Languages Open
DOI: 10.3828/mlo.v0i0.590

25

TO CITE THIS ARTICLE:

Hossain, Anushah 2026
Unsettled Scripts: Script
Invention and the Bind of
Digital Recognition. *Modern
Languages Open*, 2026(1):
8 pp. 1–25. DOI: <https://doi.org/10.3828/mlo.v0i0.590>

Published: 11 March 2026

COPYRIGHT:

© 2026 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Modern Languages Open is a peer-reviewed open access journal published by Liverpool University Press.