An Architectural View of the Typology of Writing Systems

Author information not provided

Abstract

A good typology of writing systems is important not only from a theoretical point of view, but can also help in designing techniques for teaching or learning. Moreover, with the now ubiquitous use of computing devices for writing, the typology has ramifications for computation too. In this paper we propose an architectural view of the typology of writing systems. We suggest that a writing system could be seen as a system composed of modules and their interfaces. We also suggest that there is a common universal inventory of such modules and interfaces, out of which only some are used by a specific writing system. A writing system can be typified by the modules and the interfaces that it uses and their architectural arrangement. We argue that this view of the typology of writing systems would avoid many of the problems faced by other approaches and would also have many practical benefits. It also allows the typological description of a writing system to be as coarse or as detailed as required. We present a non-exhaustive typology covering some writing systems to illustrate the idea.

1 Introduction

As things currently stand, there has been more systematic work on the typology of languages (Greenberg, 1978; Comrie, 1989) than on the typology of writing systems, to the extent that language typology is a well recognized sub-field in linguistics. This might be partly because writing systems are, to a great extent, taken for granted and their study is not given that much importance. For many reasons, this is an undesirable situation, of which we will highlight only two major reasons. The first has been recognized widely in the writing systems research community and even outside, to some degree, and it is that understanding everything about the writing systems of the world can help in improving the literacy levels in the world and in making second language acquisition easier. The second reason is that, with the largescale, almost ubiquitous, shift to the digital medium, text processing has become a major area of research. And to process

text properly (accurately, effectively and efficiently) we need techniques that can deal with the nature and nuances of the writing systems.

On closer look, we can find out that designing or discovering a good typology of writing systems is more difficult that is usually supposed. This is evident from the fact there is as much (or perhaps more) disagreement about the existing typologies of writing systems than about the typologies of languages.

1.1 Typology vs. Taxonomy

One of the points that we make in this paper is that there is a difference in building an acceptable taxonomy or classification of writing systems and proposing a typology of writing systems, although the two terms are often used interchangeably. This point is analogous to the case of languages. It is one thing to have a taxonomy of the world's languages, and quite another to propose a typology of languages. As one of the most popularly accepted taxonomy of languages[1] shows, the categories and sub-categories of languages need not be described by any linguistic terms at all. Instead, they can just be described by some geographical (Germanic, Indo-Iranian, Sino-Tibetan) or historical-cultural (Semitic, Dravidian, Romance) terms. Language typology, on the other hand, has to be in terms of the linguistic characteristics of the languages. Similarly, it should be reasonably easy to come to a consensus on an acceptable taxonomy of writing systems, but just proposing a good typology of writing systems requires considerable work, let alone achieving a consensus about it.

A good typology should be able to account for the differences in the 'nature and nuances' of the writing systems, thereby clearly showing how one writing system differs from and is similar to some writing systems. Several attempts have been made to achieve this goal (Sampson, 1985; Daniels & Bright, 1996; Sproat, 2000; Rogers, 2005). However, most of these attempts are at building a classification scheme or a taxonomy of writing systems. It is true that some of these attempts (Sproat, 2000) are based much more on the linguistic nature of the world's writing systems as compared to the language taxonomy mentioned above. Also, it is difficult to make a clear distinction between a typology and a taxonomy and these 'classifications' are actually partially typologies. Still, the primary goal is to classify, whereas we argue that typology should go hand in hand with the study of universals. It should not only be able to classify all the writing systems, but should be able to distinguish two writing systems, even when they are close[2]. The existing classifications do not attempt to do this, although there has been (separately) a great deal of study of different writing systems and even their relationships (Holme, 2004). The ideal typology would

[1] Ethnologue: Languages of the World, SIL International, `http://www.ethnologue.com/web.asp`

[2] Since a writing system concerns a script-language pair, closeness of two writing systems would depend on the closeness of two scripts as well as the writing-related aspects of the languages for which they are used.

describe writing systems in terms of the universals and with sufficient details to distinguish even two close writing systems. It would not be in terms of something like geographical or ethnic entities, as is the case with popular language taxonomies.

1.2 Taxonomies of Writing Systems

There had been informal classifications of writing systems before, but more scientific work on this problem probably started with Gelb's unilinear theory of development (Rogers, 2005) and has been influenced by it. According to this theory, writing systems have developed though a natural progression from pictographic to syllabic to phonemic. This linear and evolutionary aspect of the theory has now been more or less rejected (Penn & Choma, 2006). The shift from this one-dimensional classification has happened somewhat slowly. Since then a two dimensional classification scheme (Sproat, 2000) has been suggested and improved upon (Rogers, 2005). Rogers, in fact, first provides a basic three way classification: semantic, glottographic[3] and phonemic. But then he explains that neither a purely semantic, not a purely phonemic writing system exists, though both are theoretically possible. Therefore, all the writing systems of the world can be termed as glottographic. He then classifies these glottographic writing systems using a scheme that is a modification of the two dimensional classification proposed by Sproat, with the dimensions being labeled as phonography and morphography instead of phonography and logography. Arguing that Sproat's classification is too unstructured, Rogers suggests orthographic depth as the second dimension, rather than logography, which also captures orthographic depth to an extent. Writing systems can have either shallow or deep orthographies, with the 'depth' indicating the degree of correspondence between the letters (or graphemes) and phonemes, such that Finnish would classify as a shallow orthography, whereas English would be deep. Sproat's list of the type of phonographies has *consonantal*, *polyconsonantal*[4], *alphabetic*, *moraic*[5] and syllabic. In contrast, the list proposed by Rogers is: *abjad*[6], *alphabetic*, *abugida*[7], *moraic* and *syllabic*. However, neither Sproat not Rogers explain how orthographic depth might be measured, leaving the positioning of the writing systems a subjective matter.

One problem with these classifications is that just as there is no purely

[3]Glottographic writing systems are those which are neither purely semantic (like Bliss) nor purely phonetic (like IPA), but provide 'enough information to the reader to construct appropriate semantic and phonological representation'.

[4]One symbol for a sequence of consonants.

[5]'Mora' is a unit of sound that determines syllable weight and is possibly shorter than a syllable. Moraic writing systems usually have one symbol for one mora.

[6]Writing systems with symbols for only consonants and not vowels, e.g. the semitic writing systems.

[7]The consonant symbol usually includes a vowel (shwa) sound.

phonemic writing system, there is no purely syllabic writing system. Moreover, since these are mainly classifications, they do not give other typological information about the writing systems that could be relevant for certain purposes. For example, it does not specify that Japanese mixes different kinds of writing systems or that Chinese has an important semantic aspect. They have the advantage of brevity and elegance, but a more informative typology will be useful for many purposes.

1.3 Outline

To summarize the above discussion, the result of the work on the typology of writing systems will not be a taxonomy where each writing system is described by one term that refers to its category, sub-category or sub-sub-category[8]. Rather, it would be a description of how the writing systems of the world differ from and resemble each other in terms of specific linguistic characteristics. As Comrie also notes (Comrie, 1989), the study of language universals and the study of language typology might seem to be opposites, even conflict with one another, but in practice they do proceed in parallel. It is very difficult to classify a given piece of work as being specifically on language universals as opposed to language typology or vice versa. Something similar can be said about writing systems. In this paper we take the position that studying the typology of writing systems has to go hand in hand with the study of 'universals'. In a way, the inventory of modules and their interfaces of the writing systems of the world represents, at least partly, the collection of the universals of writing systems.

Towards this aim, we claim that one of the most effective ways to study the typology of writing systems is by taking an architectural view of the 'nature and nuances' of writing systems. By this we mean that all the writing systems of the world can be seen as being built from a finite universal inventory of modules and their interfaces. Moreover, these modules and interfaces are not purely about the morphemic and sub-morphemic symbols. As has been pointed out, a writing system is a mapping from a script to a language. This implies that the nature of writing systems cannot be completely explained without going into at least a bit of the nature of the language, apart from the nature of the script. We will further elaborate on these points in the following sections.

But before proceeding, we would like to add a clarification about the terminology. Since, for the purposes of this paper, we used certain terms in ways which may not be universally acceptable, we will specify what we mean by these terms. One of these terms is 'grapheme'. In this paper, we do not use this

[8]One such fairly exhaustive taxonomy is in terms of four levels: 'Typology' (the type, such as Abjad), 'Identity' (e.g., Arabic, Latin), 'Subsidiary' (e.g., Urdu, Arwi) and 'Constituency' (the inventory of symbols). Actually, at the fourth level, it goes beyond being just a taxonomy. Still, it is basically a classification scheme rather than a typology, although it uses the word 'typology' for the first level of classification. More details can be found at: `http://en.wikipedia.org/wiki/Wikipedia:WikiProject\_Writing\_systems`

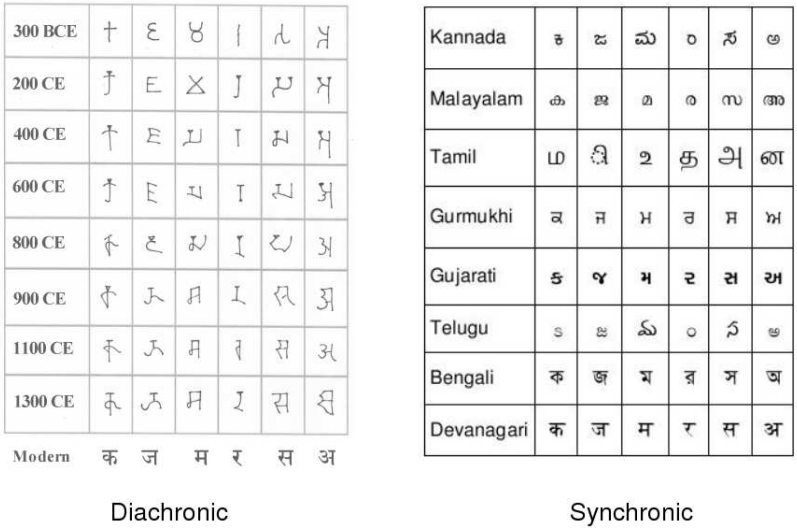

Figure 1: Synchronic and diachronic mapping of graphemes to letters: Some examples from Brahmi origin scripts

term to mean a 'letter'. We make a clear distinction between the two. Thus, a letter will be considered distinct from the grapheme(s) by which it is represented physically, say on the paper. Further, an alphabet will be supposed to mean a collection of letters, not of graphemes. A grapheme will be taken to mean an abstraction of shapes (graphs), whereas a letter will mean an abstraction over graphemes. The more generalized version of this distinction is the difference between symbols and symbol abstractions. A graph or a grapheme is a symbol, whereas a letter is actually a symbol abstraction. The letter 'a' may be represented by different graphemes in different scripts (say, Latin and Cyrillic) and is a symbol abstraction over these different graphemes. Similar is the case of Brahmi origin scripts, which basically share the same set of letters, but those letters are written very differently in different scripts, even if all the ways of writing them are derived from the same way, i.e., the same graphemes that were used in the remote past. Figure 1 shows (for Brahmi origin scripts) how the same letter can map to different graphemes, both synchronically and diachronically.

2 Modules and Interfaces

For answering any question about the possible modules and interfaces for writing systems, we have to consider what are the kinds of information that writing systems allow to be represented (or encoded) and also how they do it. The kinds of information obviously have to be mainly a subset of the information represented by languages. In other words, we can use the well established categories of linguistic information such as phonological (or phonemic), phonetic,

morphological, syntactic and semantic. According to the architectural view of writing systems, there could be a writing system module for each of these kinds of linguistic information. We will call them L-modules (L for Language). In addition, there could be other modules which represent the devices used by a writing system to encode information. These devices again are all quite well known and would include letters, graphs, graphemes etc. The modules representing these devices basically take care of the symbols (graphs) and symbol abstractions (letters) used by the writing systems. We will call them D-modules (D for Device). These modules (whether representing the information being encoded or the devices being used for encoding) could be arranged and connected together by means of some interfaces. For example, there could be an interface between the phonological or the phonetic module and the graphemic module. Similarly, there could be an interface between the graphemic and the letter modules.

Since it is not the responsibility of the writing systems to relate one kind of linguistic information (say, phonological) to another (say, morphological) — that being the responsibility of the language — there is no need of any interfaces between the L-modules. Therefore, interfaces can only be either between the D-modules or between the D-modules and the L-modules. The first kind would specify how the devices (symbols and symbol abstractions) are combined, whereas the second kind would specify how the devices are used to encode linguistic information.

In many cases, interfaces can be just mappings as in traditional discussions about writing systems. However, in many other cases there might be more complicated relationships. For example, in Devanagari, the vowel sign *i* is positioned before the consonant or the consonant cluster to which it is attached, which is equivalent to first mapping the letter *i* to the grapheme *i* and then having a 'move' operation on the grapheme. In fact this is how the relationship is represented on computers using encodings like ISCII and Unicode. Another example is that of consonant clusters. Brahmi origin scripts usually have a number of symbols for consonant clusters like *pra* (प्र), in addition to the core alphabet. The best way to represent these symbols (all of which are possible syllables) is as generation using a simple context free grammar or Finite State Transducer (FST). Figure 2 shows a grammar for generating all *akshars* (orthographic syllables) for Brahmi origin scripts. This grammar can also generate the consonant clusters for which there are single symbols. Figure 3 shows the two examples of letter-grapheme relationships mentioned.

3 An Architectural View of Writing Systems

Having explained the concept of modules and interfaces with regard to writing systems, we now describe the architectural view of writing systems as a view in which a writing system is seen to be a combination of modules and interfaces in

Akshar ::= Vowel Akshar | Consonant Akshar
Vowel Akshar ::= Vowel [Vowel Modifier]
Consonant Akshar ::= Pure Consonant
    | Modified Consonant
Pure Consonant ::= Consonant [Nukta] Halant
Modified Consonant ::= Consonant [Nukta]
    [Maatraa] [Vowel Modifier]

S: Start
C: Consonant
V: Vowel
M: Maatraa
N: Nukta
H: Halant
D: Vowel modifier

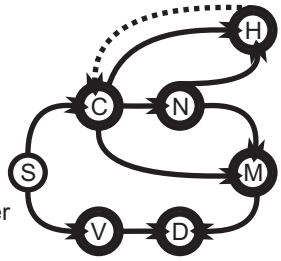

Figure 2: The *akshar* grammar represented as an FST and also in the Backus-Naur Form (BNF). The dotted arc is for (optionally) allowing consonant clusters.

क + ि = कि प + ◌् + र = प्र

k + i = ki pa + halant + r = pra

Figure 3: Two examples of relationships going beyond simple and direct mapping: Although it may be possible to represent these relationships simply by enumerating all such cases as mappings, they can be more concisely represented by other means such as a grammar or set of rules (an interface).

a particular arrangement that typifies that writing system. The specifications of the modules and the interfaces as well as the way they are combined determine the nature of a writing system and, consequently, also determine the place of that writing systems in the general typology. Thus, to categorize a writing system is to *minimally describe* it in terms of modules and interfaces. The study of the typology of writing systems would then consist of four major steps:

1. Prepare the inventory of modules and interfaces
2. Prepare the meta-descriptions or the 'schemas' of these modules and interfaces
3. Prepare the minimal descriptions of the modules and the interfaces such that their meta-descriptions are filled in to give concrete descriptions. Minimal description is the least detail that is required to distinguish one writing system from another. However, it can be dropped if we need to make only coarse distinctions among the writing systems.
4. Minimally describe individual writing systems. Such description would constitute the typological description of the writing system.

The next section illustrates how we can follows these four steps to build a typology based on an architectural view.

4 A Typology through Architectural Modeling

Since this paper is the first effort to propose a typology in terms of modules and interfaces, the typology that we present here is not exhaustive or complete but only illustrative. We will consider writing systems that use three families of script: the scripts of Latin, Semitic and Brahmi origins. One commonly accepted typological specification of these three families of scripts will that they are Alphabetic, Abjad and Abugida respectively. But the Brahmi origin scripts are also often called syllabic scripts or syllabaries. We will not go into much details about the problems with these labels as they have been discussed elsewhere. However, just to point out a few problems, it has been argued that the defining characteristic of one is often present in another, e.g. some of the Latin scripts have syllabic properties, Abjad and Abugida scripts have alphabetic properties and Brahmi origin scripts do not really have a unique symbol for each syllable, as should be the case with purely syllabic scripts. These problems actually point the way to the solution based on the architectural view because all of these writing systems seem to share a lot of characteristics and the way they differ is not so much with respect to one characteristic but in the way these characteristics are combined.

Therefore, to proceed with the architectural description of these writing systems, we have to first ask what are the characteristics present in these writing systems. The answer can be that, first, they all have an alphabet. An alphabet is defined as a set of letters such that these letters primarily serve to represent the phonology of the language. Second, at least one of them has syllabic properties (in the sense that there is a direct, but not complete, correspondence between syllables and symbols). Third, they all have some degree of phonetic correspondence between letters and phonemes. Fourth, they all have shapes assigned to symbols in a way that is not completely arbitrary, at least in the sense that there are some basic shapes which are reused to form different symbols. Then there are some other more specific characteristics such that the Latin scripts have different symbols for different 'cases' (small and capital), the Abjad scripts have different symbols for the same letter based on the position in the word (initial, final, medial) and the Brahmi origin scripts have different symbols for some consonant conjuncts and they also have different symbols for dependent and independent vowels. Finally, they also may have characteristics which actually depend on the properties of the language, e.g. in the case of Brahmi origin scripts, the tokenization (i.e., the segmentation of written text with space) may be either idiosyncratic or may be related to the morphology of the language. For example, Thai has no word segmentation. Also, in the South Indian writing systems (Malayalam, Telugu), case markers and some other morphological markers are usually joined together with the word, whereas they are usually written separately in the North Indian writing systems, though both of them use Brahmi origin (Abugida) scripts.

To put it formally, the typological description of a writing system $\mathcal{W}$ can be specified as:

$$\mathcal{W} = \{\mathcal{M}, \mathcal{I}, \mathcal{M}_{\mathrm{s}}, \mathcal{I}_{\mathrm{s}}, \mathcal{M}_{\mathrm{d}}, \mathcal{I}_{\mathrm{d}}\} \tag{1}$$

where $\mathcal{M}$ is the list of the modules that the writing system uses, $\mathcal{I}$ is the list of the interfaces used, $\mathcal{M}_{\mathrm{s}}$ is the list of the module schemas, $\mathcal{I}_{\mathrm{s}}$ is the list of the interface schemas, $\mathcal{M}_{\mathrm{d}}$ is the list of minimal descriptions of the modules and $\mathcal{I}_{\mathrm{d}}$ is the list of minimal descriptions of the interfaces, such that there are only as many schemas and minimal descriptions as there are modules and interfaces:

$$|\mathcal{M}| = |\mathcal{M}_{\mathrm{s}}| = |\mathcal{M}_{\mathrm{d}}| \text{ and } |\mathcal{I}| = |\mathcal{I}_{\mathrm{s}}| = |\mathcal{I}_{\mathrm{d}}| \tag{2}$$

As stated earlier, a writing system uses modules and interfaces out of a universal set:

$$\mathcal{M} \subset \mathcal{M}_{\mathrm{u}} \text{ and } \mathcal{I} \subset \mathcal{I}_{\mathrm{u}} \tag{3}$$

Theoretically, there can be an interfaces between any two modules:

$$|\mathcal{I}| \leq |\mathcal{M}| \times |\mathcal{M}| \text{ and } |\mathcal{I}_{\mathrm{u}}| \leq |\mathcal{M}_{\mathrm{u}}| \times |\mathcal{M}_{\mathrm{u}}| \tag{4}$$

subject to the condition that there are no interfaces between two L-modules. For convenience, we are not making a notational distinction between the D-modules and the L-modules in this section.

4.1 The Inventory of Modules and Interfaces

From the above discussion, which is not exhaustive, we can try to derive the list of modules and interfaces. The proposed list of modules is:

1. The Alphabetic Module, $\mathcal{A}$ (D-module)
2. The Graphemic Module, $\mathcal{G}$ (D-module)
3. The Syllabic Module, $\mathcal{S}$ (L-module)
4. The Phonological Module, $\mathcal{P}$ (L-module)

Thus, we have:

$$\mathcal{M} = \{\mathcal{A}, \mathcal{G}, \mathcal{S}, \mathcal{P}\} \tag{5}$$

Assuming that only binary interfaces (i.e., interfaces between two modules, not three or more) are possible, the proposed interfaces for the three families of writing systems are:

1. Alphabetic-Graphemic Interface, $\mathcal{A} \diamond \mathcal{G}$
2. Alphabetic-Syllabic Interface, $\mathcal{A} \diamond \mathcal{S}$
3. Alphabetic-Phonological Interface, $\mathcal{A} \diamond \mathcal{P}$
4. Syllabic-Graphemic Interface, $\mathcal{S} \diamond \mathcal{G}$

So that the list of the interfaces is:

$$\mathcal{I} = \{\mathcal{A} \diamond \mathcal{G},\ \mathcal{A} \diamond \mathcal{S},\ \mathcal{A} \diamond \mathcal{P},\ \mathcal{S} \diamond \mathcal{G}\} \tag{6}$$

The above list does not mean that, for example, there is no interaction or connection between the Syllabic and Phonological Modules, but that the description of such interaction or connection is redundant because it can be derived from the other interfaces, viz. the Alphabetic-Syllabic and the Alphabetic-Phonological Modules. The fact about which interfaces are required and which are redundant is an important part of the typology of writing systems, just like the fact about which modules are required and which are redundant or derivable (if not completely inapplicable). These two facts alone may be enough for a coarse typology of writing systems, but a third fact is needed for a less coarse typology. This is discussed in the next sub-section.

4.2 Schemas of Modules and Interfaces

Once we have arrived at an inventory of modules and interfaces, we might also need to describe their schemas for differentiating two related writing systems. A schema is a meta-description, i.e., it describes what kind of elements can constitute a module or an interface, whereas a (minimal) description details the actual elements that constitute the module of the interface.

4.2.1 The Alphabetic Module Schema

The schema for the Alphabetic Module would describe how many different kinds of letter are there in a writing system and what those kinds are:

$$\mathcal{A}_s = \{\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_{|\mathcal{A}_s|}\} \tag{7}$$

where $\mathcal{A}_s$ is the schema of the Alphabetic Module and λ denotes the kinds of letters that the writing system has, with the subscript s denoting the schema.

An example (for writing systems without the distinction between capital letters and small letters) would be:

$$\mathcal{A}_s = \{\mathrm{free\ letter},\ \mathrm{bound\ letter}\} \tag{8}$$

where *bound letter* roughly corresponds to a diacritic but is more general. This schema would apply for the Semitic writing systems.

Another example (for scripts which have diacritics as well as the distinction between capital letters and small letters) would be:

$$\mathcal{A}_s = \{\mathrm{small\ free\ letter},\ \mathrm{capital\ free\ letter},\ \mathrm{bound\ letter}\} \tag{9}$$

Latin scripts as used for the Romance languages would fall into the above category.

For the Latin script as used for English, which does not have any diacritics, it would just be:

$$\mathcal{A}_s = \{\mathrm{small\ free\ letter},\ \mathrm{capital\ free\ letter}\} \tag{10}$$

For some other writing systems (e.g. Brahmi origin scripts as used for Indian languages), there is also the distinction between free vowels and bound vowels (*maatraas*):

$$\begin{aligned}\mathcal{A}_s = \{&\mathrm{consonant},\ \mathrm{free\ vowel},\ \mathrm{bound\ vowel},\\ &\mathrm{consonant\ modifier},\ \mathrm{vowel\ modifier}\}\end{aligned} \tag{11}$$

where *consonant modifier* is a kind of diacritic that only modifies a consonant, whereas a *vowel modifier* only modifies a vowel.

In the case of Brahmi origin scripts, there is another special kind of letter (called *halant*, see Figure 3) that is used to represent shwa deletion, i.e., it converts a syllable into a phoneme:

/kə/ + *halant* = /k/

We need another category for such letters which act as operators to convert one linguistic unit (e.g. syllable) to another (e.g. phoneme), as against the normal modifier letters or diacritics which are just additional symbols to create new symbols of the same form (or to create composite symbols by addition rather than subtraction) to reduce the total number of symbols required by the alphabet. We could call them *operator symbols*, so that the extended schema for the Alphabetic Module for Brahmi origin scripts would be:

$$\begin{aligned}\mathcal{A}_s = \{&\mathrm{consonant},\ \mathrm{free\ vowel},\ \mathrm{bound\ vowel},\\ &\mathrm{consonant\ modifier},\ \mathrm{vowel\ modifier},\ \mathrm{operator\ letter}\}\end{aligned} \tag{12}$$

It may be mentioned here that the case of Tamil, which also uses a Brahmi origin script, is slightly different in this respect. The equivalent of *halant* (called *pulli*) is actually just like a bound vowel or *maatraa*, rather than an operator that deletes a *maatraa* from a consonant plus vowel syllable:

/k/ + *pulli* = /kə/

4.2.2 The Graphemic Module Schema

Whereas the Alphabetic Module is abstract, the Graphemic Model is concrete in the sense that it models the actual symbols (shapes) that can be drawn for a specific language to represent linguistic units like letters, phonemes, syllables, morphemes etc. Thus, it is likely that the same basic Alphabetic Module might be shared among a large number of writing systems, but these same writing systems might have very different Graphemic Modules because the shapes are very different and not 'mutually intelligible'. This difference between the Alphabetic Module and the Graphemic Module is much less for writing systems based on the Latin script, but is much more for Brahmi origin scripts. The difference is also more if, say, one writing system uses the Latin script while another uses the Cyrillic script, even though the Alphabetic Modules for these two writing systems may not be that different.

The schema for the the Graphemic Module would describe how many different kinds of graphemes are there in a writing system and what those kinds are:

$$\mathcal{G}_s = \{\gamma_1,\ \gamma_2,\ \ldots,\ \gamma_{|\mathcal{G}_s|}\} \tag{13}$$

where $\mathcal{G}_s$ is the schema of the Graphemic Module and γ denotes the kinds of graphemes that the writing system has. The simplest example would be:

$$\mathcal{G}_s = \{\texttt{free grapheme},\ \texttt{bound grapheme}\} \tag{14}$$

Since the Graphemic Module is basically the concrete realization of the Alphabetic Module, there are close parallel between the two, especially at the schema level. Thus, for the Latin script as used for English, the schema would be:

$$\mathcal{G}_s = \{\texttt{small free grapheme},\ \texttt{capital free grapheme}\} \tag{15}$$

Similarly, for the Latin script as used for the Romance languages, the schema would be:

$$\mathcal{G}_s = \{\texttt{small free grapheme},\ \texttt{capital free grapheme}, \texttt{bound grapheme}\} \tag{16}$$

Figure 4 lists some examples of the different kinds of graphemes.

Grapheme Types	Examples	Grapheme Types	Examples
Small Free Graphemes	a, b, α, β, δ	Free Vowels	अ, आ, इ, ई
Capital Free Graphemes	A, B, Γ, Δ, V	Bound Vowels	ा, ि, ी, ु
Bound Graphemes	◌́, ◌̃, ◌̃	Syllable	प्र, क्ष, त्र, ज्ञ
Free Grapheme[b]	ک, گ, ہ	Consonant Modifier	़ (Nukta)
Free Grapheme[m]	ڈ, ک, ف	Vowel Modifier	ं (Anusvar), ँ (Chandrabindu)
Free Grapheme[e]	ش, ط, ف	Operator Grapheme	् (Halant)
Consonants	क, ख, ग, घ		

Figure 4: Examples of grapheme types for Latin, Semitic and Brahmi origin scripts

For Semitic Writing systems like Arabic, the parallel does not strictly hold, as the Graphemic Module has the extra parameter of position in a word, i.e., the grapheme representing a letter may be different depending on whether the letter occurs in the beginning, the end or middle of a word. How to model this fact in the schema may be debatable. One possible way is:

$$\mathcal{G}_s = \{\mathit{free\ grapheme}[\mathrm{b}, \mathrm{m}, e],\ \mathit{bound\ grapheme}[\mathrm{b}, \mathrm{m}, e]\} \tag{17}$$

where $[\mathrm{b}, \mathrm{m}, e]$ denotes three kinds of graphemes used for the same letter (beginning, medial and end).

Comparing the Semitic writing systems like Arabic with either the Latin script based writing systems or those based on the Brahmi origin scripts brings out another factor that the Graphemic Module has to include: the direction of writing. From the Brahmi origin scripts, yet another factor can be observed: segmentation. Thus, the extended schema would be like:

$$\mathcal{G}_s = \{\gamma_1,\ \gamma_2,\ \ldots,\ \gamma_{|\mathcal{G}_s|}\}, \Delta, \Gamma \tag{18}$$

where D is the direction of writing and can be left to right (LTR), right to left (RTL) or top to bottom (TTB). Γ denotes the kind of segmentation used by the writing system and it can be Word (for Latin and many of the Brahmi based writing systems), Clause (e.g. for Myamarese) or $\mathit{Sentence}$ (e.g. for Thai).

The schema for the writing systems using Brahmi origin scripts would be:

$$\begin{aligned}\mathcal{G}_s = \{&\mathit{consonant},\ \mathit{syllable},\ \mathit{free\ vowel},\\ &\mathit{bound\ vowel},\ \mathit{consonant\ modifier},\ \mathit{vowel\ modifier},\}\\ &\mathit{operator\ grapheme}\}, \mathrm{LTR}, \mathit{Word}\end{aligned} \tag{19}$$

4.2.3 The Syllabic Module Schema

The schema for the Syllabic Module would specify what kind of syllables have distinct symbols associated with them. Brahmi origin scripts have been called syllabic scripts, but as noted earlier, they are not strictly syllabic. Also, they have characteristics which make them similar to scripts which are described as alphabetic, phonetic, featural etc. They have symbols for letters which are smaller than syllables. They have symbols which have very close correspondence with phonemes. Also the symbols (letters) are arranged in the alphabet according their approximate phonetic features. Thus, the representation of syllables in Brahmi origin scripts is non-trivial.

In general, the syllabic module for a writing system would be described as:

$$\mathcal{S}_s = \{\psi_1,\ \psi_2,\ \ldots,\ \psi_{|\mathcal{S}_s|}\} \tag{20}$$

where ψ_1, ψ_2 etc. are the kinds of syllabic symbols present in the writing system.

The syllabic module for Brahmi origin scripts would:

$$\mathcal{S}_s = \{\mathtt{minimal\ syllable},\ \mathtt{maximal\ syllable}\} \tag{21}$$

The important thing to note here is an *operator letter* (Figure 3) can reduce the value of a *minimal syllable* to something less than a *minimal syllable*. Also, modifier and bound letters can be combined with the symbols for *minimal syllable* (which could be a *free letter*) to form a *maximal syllable*.

Minimal syllables are the shortest possible syllables, whereas *maximal syllables* those syllables which are an extension of *minimal syllables*. For Brahmi origin scripts, some example of *minimal syllables* and *maximal syllables* are:

- Minimal: अ , क , ख , प [a, ka, kha, pa]
- Maximal: अं , का , कों , खि , प्र , प्रे [aM, kaa, koM, khi, pra, pre]

4.2.4 The Phonological Module Schema

Even though some researchers have argued against the idea that there are a finite number of phonemes that can account for all possible human languages, or even a finite number of articulatory features for that matter (Port & Leary, 2005). However, for the purposes of writing systems we can assume that there are a finite number of articulatory features that can cover all writing systems. In fact, according to Port and Leary and many others, phonemes are actually a 'convenient fiction' that have their origin in writing rather than speech.

The schema of the Phonological Module can, therefore, be described in terms of the universal set of phonemic (articulatory) features which define all possible phonemes that are represented in writing systems:

$$\mathsf{F} = \{\mathsf{f}_1,\ \mathsf{f}_2,\ \ldots,\ \mathsf{f}_{|\mathsf{F}|}\} \tag{22}$$

These phonemic features can take the following values:

$$\mathsf{V} = \{v_1,\ v_2,\ \ldots,\ v_{|\mathsf{V}|}\} \tag{23}$$

For a particular writing system W, the set of features will be a subset of F:

$$\mathsf{F}_W = \{\mathsf{f}_1, \mathsf{f}_2, \ldots, \mathsf{f}_{|\mathsf{F}_W|}\},\ \mathsf{F}_W \subset \mathsf{F} \tag{24}$$

These schema provide the basis for preparing the minimal descriptions of the modules as described below.

4.3 Minimal Descriptions of Modules

To differentiate very close writing systems, we might even have to minimally describe some or all of the modules and interfaces. How much description is needed depends on how detailed and fine-grained a typology we are looking for.

4.3.1 The Alphabetic Module

The description of alphabet will includes the list of letters of all the types that are found in the alphabet, the types having been specified in the schema. Thus, the Alphabetic Module for Brahmi origin scripts would be:

$$\mathcal{A} = \mathcal{L}_{\mathsf{c}} \cup \mathcal{L}_{\mathsf{fv}} \cup \mathcal{L}_{\mathsf{bv}} \cup \mathcal{L}_{\mathsf{cm}} \cup \mathcal{L}_{\mathsf{vm}} \cup \mathcal{L}_{\mathsf{ol}} \tag{25}$$

where c, fv, bv, cm, vm, ol represent *consonant*, *free vowel*, *bound vowel*, *consonant modifier*, *vowel modifier* and *operator letter*, respectively.

Similarly, the Alphabetic Module for Latin scripts as used for Romance languages would be:

$$\mathcal{A} = \mathcal{L}_{\mathsf{cf}} \cup \mathcal{L}_{\mathsf{sf}} \cup \mathcal{L}_{\mathsf{b}} \tag{26}$$

where cf, sf and b represent *capital free letter*, *small free letter* and *bound letter*, respectively.

The lists of letters of various kinds for Brahmi origin scripts would be like the following:

- $\mathcal{L}_{\mathsf{c}}$ = (क , ख , ग , घ , ...) [(ka, kha, ga, gha, ...)]
- $\mathcal{L}_{\mathsf{fv}}$ = (अ , आ , इ , ई , ...) [(a, aa, i, ii, ...)]

In the case of Brahmi origin scripts it should be noted that the shapes of the letters as displayed above are for Devanagari script, but we can instead use Latin or Roman letters to represent them (as done above in parenthesis) because letters are abstract symbols, distinct from graphemes. We have selected Devanagari, because (apart from being the most commonly used Brahmi origin script) its extended version is meant for all major Indian languages and is almost the superset of all other Brahmi origin scripts used for Indian languages (again, in terms of letters, not graphemes).

Similarly, for Latin origin scripts:

- $\mathcal{L}_{\mathtt{cf}}$ = (A, B, C, ...)
- $\mathcal{L}_{\mathtt{sf}}$ = (a, b, c, ...)

4.3.2 The Graphemic Module

The minimal description of the Graphemic Module would again be parallel to the Alphabetic Module except that the same letter in a family of writing systems might be realised as different graphemes in different members of that family. For Semitic writing systems, the same letter might be realised as different graphemes in the same writing system, depending on the position of the letter in the word, as noted earlier.

Thus, the Graphemic Module for Brahmi origin scripts would be:

$$\mathcal{G} = \mathcal{G}_{\mathtt{c}} \cup \mathcal{G}_{\mathtt{fv}} \cup \mathcal{G}_{\mathtt{bv}} \cup \mathcal{G}_{\mathtt{cm}} \cup \mathcal{G}_{\mathtt{vm}} \cup \mathcal{G}_{\mathtt{ol}} \tag{27}$$

But for Semitic writing systems, it would be:

$$\mathcal{G} = \mathcal{G}[\mathtt{b}, \mathtt{m}, \mathtt{e}]_{\mathtt{fg}} \cup \mathcal{G}[\mathtt{b}, \mathtt{m}, \mathtt{e}]_{\mathtt{bg}} \tag{28}$$

where $\mathtt{fg}$ and $\mathtt{bg}$ denote free grapheme and bound grapheme, respectively.

4.3.3 The Syllabic Module

This module would list the syllables which have an atomic symbol for them in the writing system. Out of the three families we are considering, only Brahmi origin scripts have some symbols that directly represent syllables. Since this family of writing systems also have an alphabet, this list of syllabic graphemes would be a subset of the list of graphemes:

$$\mathcal{S} = \mathcal{S}_{\mathtt{ms}} \cup \mathcal{S}_{\mathtt{xs}} \tag{29}$$

where $\mathtt{ms}$ and $\mathtt{xs}$ denote minimal and maximal syllables, respectively.

For Devanagari, the lists of *minimal syllables* and *maximal syllables* would be:

- $\mathcal{S}_{\text{ms}}$ = (अ, ..., क, ख, ..., प, ...)
- $\mathcal{S}_{\text{xs}}$ = (..., अं, ..., का, ..., कों, ..., खि, ..., प्र, ..., प्रे, ...)

4.3.4 The Phonological Module

Let the universal set of phonemes represented by writing systems in human languages be:

$$\Phi = \{\phi_1,\ \phi_2,\ \ldots,\ \phi_{|\Phi|}\} \tag{30}$$

A phoneme is defined as a set of feature-value pairs:

$$\phi = \{(f_1, v_1),\ (f_2, v_2),\ \ldots,\ (f_{|\phi|}, v_{|\phi|})\} \tag{31}$$

For a specific writing system W, the set of phonemes that occur in that language will be a subset of Φ:

$$\Phi_W = \{\phi_1, \phi_2, \ldots, \phi_{|\Phi_W|}\},\ \Phi_W \subset \Phi \tag{32}$$

4.4 Schemas of Interfaces

For many writing systems such as Latin as used for English, the schemas of the interfaces may be so trivial that they need not be described. This will happen because there is a one-to-one correspondence between letters and graphemes and there is no Syllabic Module. We might still have an Alphabetic-Phonological interface schema. In most cases, the schema for an interface is just the fact there is a mapping from elements of one module to the other, e.g. from letters to phoneme or from syllables to graphemes. However, even for the restricted set of writing systems we are looking at, there are cases when the schema might say that there is more than just a simple mapping. For example, Brahmi origin scripts would have an Alphabetic-Syllabic interface schema that would say that the modules are related by a grammar that can be described by Backus-Naur formalism.

4.5 Minimal Descriptions of Interfaces

As mentioned in the previous section, the interfaces could be either too trivial to be described or they could be described by mappings or grammars, depending on the writing system.

4.5.1 Alphabetic-Graphemic Interface

This interface would be one of the things that would distinguish one writing system belonging to the family that uses Brahmi origin scripts. The description would be a mapping from letters to graphemes. In the case of Semitic writing systems, one letter might map to three graphemes for the three possible positions of the letter (initial, medial and final).

4.5.2 Alphabetic-Syllabic Interface

For Brahmi origin scripts, the 'grammar' (also see Figure 2) that describes this interface can be represented in the Backus-Naur form (BNF) as:

Akshar ::= Vowel Akshar | Consonant Akshar
Vowel Akshar ::= Vowel [Vowel Modifier]
Consonant Akshar ::= Pure Consonant | Modified Consonant
Pure Consonant ::= Consonant [Nukta] Halant
Modified Consonant ::= Consonant [Nukta] [Maatraa] [Vowel Modifier]

Whether there should be a similar grammar for Semitic writing systems can be a matter of debate because they do have a process of composition of larger units when letters are joined together, but these larger units are not necessarily syllables.

4.5.3 Alphabetic-Phonological Interface

This would again provide a mapping, this time from letters to phonemes and phonemic features. For Brahmi origin scripts, this mapping is almost one-to-one, whereas for Latin scripts, especially as used for English, the mapping would be one-to-many.

4.5.4 Syllabic-Graphemic Interface

This would provide a mapping from syllables to graphemes and (for Brahmi origin script, which is the only one having this interface) it would be a subset of the mapping provided for the Alphabetic-Graphemic interface.

4.6 Minimal Description of a Writing System

Given the level of detail we need in our typology, we can use the four increasingly specific sets of facts to minimally describe a writing system. A hierarchical minimal description of all the writing system being considered gives us the typology from the architectural point of view.

5 Practical Advantages

Studying writing systems from the architectural point of view have some very clear advantages as compared to the more traditional methods. The fact that a writing system can be described as being composed of some universal modules and interfaces has direct implications for many practical purposes. In this section we will consider some of these.

5.1 For Teaching and Learning

The first obvious application of the architectural view is that it can be used to design techniques for teaching or learning more effectively. We can design teaching or learning aids which correspond to these modules or at least benefit from their description. This may be true for learning a writing system for the first time (i.e., literacy), but it is likely to be much more true for learning a second writing system. For example, if someone already knows how to read and write in Latin (English), the the architectural view of writing system can help us in devising ways to make it easy for him to learn Perso-Arabic (Urdu). This is because the minimal descriptions of writing systems will easily identify the ways in which the two writing systems differ, so that we can focus on those differences.

Also, since all writing systems are described in terms of the same universal inventory of modules and interfaces, we can devise techniques in such a way that they can be easily adapted for other writing systems (first time learning) or other pairs of writing systems (second time learning). This part of the advantage is really important for regions with a multitude of languages and writing systems, i.e., linguistically highly diverse regions of the world, which also happen to lack the infrastructure and other resources.

5.2 For Computation

One still more obvious advantage of the view being proposed is that it has the potential of being almost directly applied for computation as higher level design of practical computational systems can be viewed in terms of modules and interfaces. The support for writing systems on computers could be improved by having a design that takes into account the inventory of modules and interfaces, rather than mostly focusing on alphabets or character lists and graphemes. Even if generalized support for writing systems is not designed along the lines that this paper suggests, special text processing or language processing applications can extract more information from the text in a more systematic way by considering the architectural view of writing systems. In this section we will mention some such applications.

The first obvious application is spell checking. The way spell checking is performed depends, or should depend, on the characteristics of the writing sys-

tem. Thus, simple edit distance[9] based methods (Juan & Vidal, 2000) might be good for writing systems based on scripts of Latin origin, whereas for the writing systems used for Chinese and Japanese other methods (Lee, Ng, & Lu, 1999) might give better results. Most of the edit distance based methods (implicitly) take into account only the Alphabetic Module, whereas other modules might also be relevant for the purposes of spell checking. For example, a Phonological Module can help improve spell checking for writing systems (e.g. Brahmi origin scripts) which have a close correspondence between letters and phonemes (Singh, 2006).

Another category of applications are those like letter to phoneme (L2P) transcription (Bartlett, Kondrak, & Cherry, 2008) and transliteration (Knight & Graehl, 1997; Haizhou, Min, & Jian, 2004). Both these applications can gain from a method designed to take into account the characteristics of the writing systems. For example, it is possible to build a general and better transliteration system for all Brahmi origin scripts that includes an Alphabetic Module and a Phonological Module (Surana & Singh, 2008). Similarly, having a Semantic Module can lead to improvement in transliteration of Chinese or Japanese text (Li, Sim, Kuo, & Dong, 2007). The same applies to yet another category of applications that involve calculating distances between languages and building phylogenetic trees (Ellison & Kirby, 2006).

6 Conclusion

In this paper we proposed a view of the typology of writing systems that is based on two major ideas. The first is that the typology of writing systems requires a minimal description based approach and the second is that a writing system can be described in terms of modules and interfaces. The inventory of these modules and interfaces is, in a way, representative of the writing system 'universals'. The modules can belong to one of the two categories. In the first category (L-modules) come those modules which represent the linguistic information that is encoded through writing systems. The second category (D-modules) consists of those modules which represent the devices (symbols and symbol abstractions) used for encoding linguistic information. Interfaces define how these modules combine together. There can only be interfaces between either two D-modules or between a D-modules and an L-module, as the interfaces between two L-modules are not required by writing systems (that being the domain of the language). Based on this idea of modules and interfaces, the study of the typology of writing systems would consists of a four step process: preparing the inventory of modules and interfaces used by writing systems; preparing meta-descriptions or 'schemas' of the modules and the interfaces; minimally describe the modules

[9]The distance between two string is taken as the number of operations (inserting, deletion, substitution etc.) required to convert one into the other.

and the interfaces; and minimally describe individual writing systems in terms of modules and interfaces. We also presented an illustrative example of this architectural view the typology of writing systems and discussed how it can be beneficial for teaching, learning and especially for computation.

References

Bartlett, S., Kondrak, G., & Cherry, C. (2008, June). Automatic syllabification with structured SVMs for letter-to-phoneme conversion. In *Proceedings of acl-08: Hlt* (pp. 568–576). Columbus, Ohio: Association for Computational Linguistics.

Comrie, B. (1989). *Language universals and linguistic typology* (Second Edition ed.). Oxford: Blackwells.

Daniels, P., & Bright, W. (1996). *The world's writing systems*. New York: Oxford University Press.

Ellison, T. M., & Kirby, S. (2006). Measuring language divergence by intra-lexical comparison. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics.* Sydney, Australia: Association for Computational Linguistics.

Greenberg, J. H. (1978). *Universals of human language*. Stanford, California: Stanford University Press.

Haizhou, L., Min, Z., & Jian, S. (2004). A joint source-channel model for machine transliteration. In *Acl '04: Proceedings of the 42nd annual meeting on association for computational linguistics* (p. 159). Morristown, NJ, USA: Association for Computational Linguistics.

Holme, R. (2004). *Literacy: an introduction*. Edinburgh University Press.

Juan, A., & Vidal, E. (2000). On the use of normalized edit distances and an efficient k-nn search technique (k-aesa) for fast and accurate string classification. In (p. Vol II: 676-679).

Knight, K., & Graehl, J. (1997). Machine transliteration. In *Proceedings of the eighth conference on european chapter of the association for computational linguistics* (pp. 128–135). Morristown, NJ, USA: Association for Computational Linguistics.

Lee, K. H., Ng, M. K. M., & Lu, Q. (1999). Text segmentation for chinese spell checking. *Journal of the American Society for Information Science*, *50*(9), 751-759.

Li, H., Sim, K. C., Kuo, J.-S., & Dong, M. (2007). Semantic transliteration of personal names. In *Acl.*

Penn, G., & Choma, T. (2006). Quantitative methods for classifying writing systems. In *Hlt-naacl.*

Port, B., & Leary, A. (2005). Against formal phonology. *Language*, 81(4):927–964.

Rogers, H. (2005). *Writing systems. a linguistic approach*. United Kingdom, Blackwell Publishing.

Sampson, G. (1985). *Writing systems: A linguistic introduction*. Stanford: Stanford University Press.

Singh, A. K. (2006). A computational phonetic model for indian language scripts. In *Proceedings of the constraints on spelling changes: Fifth international workshop on writing systems.* Nijmegen, The Netherlands.

Sproat, R. (2000). *A computational theory of writing systems*. Cambridge: Cambridge University Press.

Surana, H., & Singh, A. K. (2008). A more discerning and adaptable multilingual transliteration mechanism for indian languages. In *Proceedings of the third international joint conference on natural language processing (ijcnlp).* Hyderabad, India: Asian Federation of Natural Language Processing.