Oct 18, 2011 — Santa Clara, CA
35th Internationalization and Unicode Conference

An Abstract Model for the Typography of Perso-Arabic Script

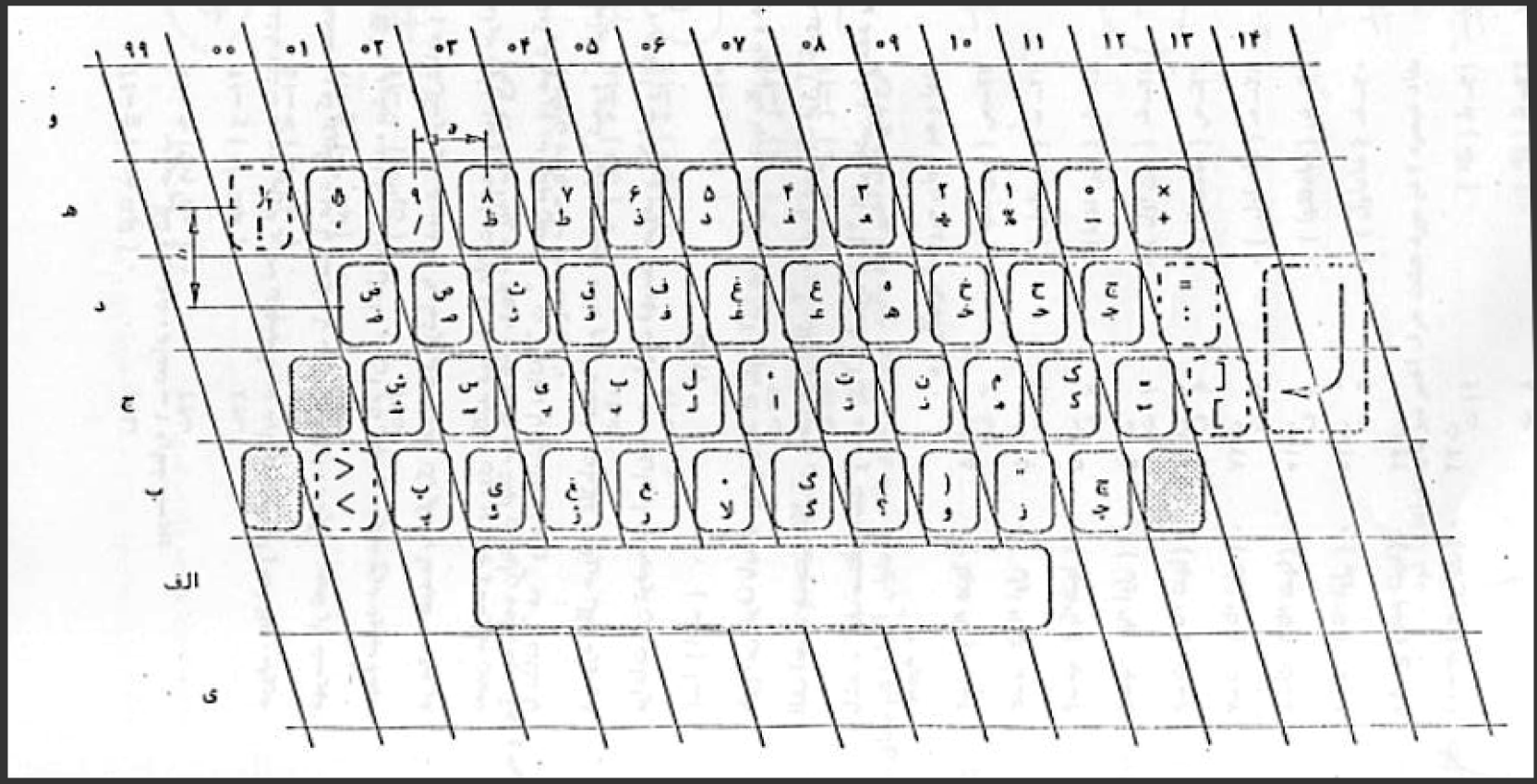
Behnam Esfahbod
Persian Internet Society
behnam@esfahbod.info

Yahya Tabesh
Sharif Institute of Technology
tabesh@sharif.edu

Encoding Arabic Script

Typographic Encodings

- Used in early computing era
  - Came from typewriters
  - One code-point for each “shape”
- Unicode obsolete Arabic blocks
  - “Arabic Presentation Forms-A” (U+FB50 – U+FDFF)
  - “Arabic Presentation Forms-B” (U+FE70 – U+FEFF)
- Hard to process
  - Up to four code-points for letters
  - Easy to visualize
- One glyph for each code-point
  - Easy to compare shapes



Persian Typewriter Standard

Iranian National Standard, ISIRI 820

Four shapes for AIN and GHAIN
Three shapes for HEH
One shape for ALEF

First Persian Encoding Standard

Iranian National Standard, ISIRI 2900

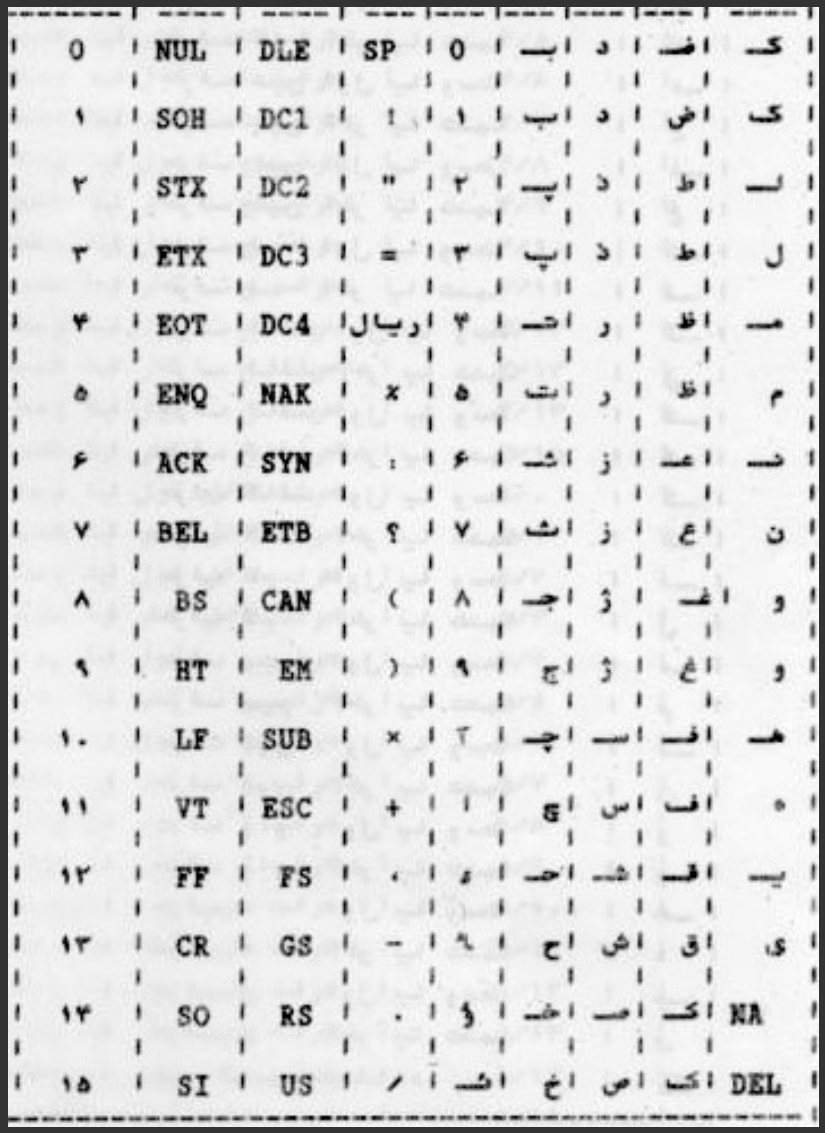
A 7-bit code-page

Two shapes/code-points for almost all letters

One shape for ALEF family

One shape for HIGH HAMZA ligatures

No room left for LIGATURE ALEF WITH MADDA family



۰	NUL	DLE	SP	۰	ا	ب	ف	ک
۱	SOH	DC1	!	۱	ا	ب	ض	ک
۲	STX	DC2	"	۲	ا	پ	ط	گ
۳	ETX	DC3	=	۳	ا	پ	ط	ل
۴	EOT	DC4	ریال	۴	ر	ت	ظ	م
۵	ENQ	NAK	x	۵	ر	ت	ظ	م
۶	ACK	SYN	:	۶	ز	ث	ع	ن
۷	BEL	ETB	؟	۷	ز	ث	غ	ن
۸	BS	CAN	(	۸	ژ	ج	ف	و
۹	HT	EM	)	۹	ژ	چ	غ	و
۱۰	LF	SUB	×	۲	س	چ	ف	ه
۱۱	VT	ESC	+	۱	س	ح	ف	ه
۱۲	FF	FS	،	۲	ش	ح	ق	ی
۱۳	CR	GS	-	۳	ش	خ	ق	ی
۱۴	SO	RS	.	۴	ص	خ	ک	NA
۱۵	SI	US	/	۵	ص	خ	ک	DEL

Semantic Encodings

- Unicode Arabic blocks
  - “Arabic” (U+0600 – U+06FF)
  - “Arabic Supplement” (U+0750 – U+077F)
- Easy to process
  - Up to four code-points for letters
- Hard to visualize
  - Joining Algorithm
  - Bidirectional Algorithm
- Hard to compare shapes

Why An Abstract Model?

Fonts

- Dots in non-Isolated forms
- Same base shapes
    - Yeh-based letters
- Different base shapes
    - Heh-based letters
    - Keh-based letters
    - Yeh-based letters
- Ligatures
    - YEH + HAMZA ABOVE should not have any dots
- Other Cases

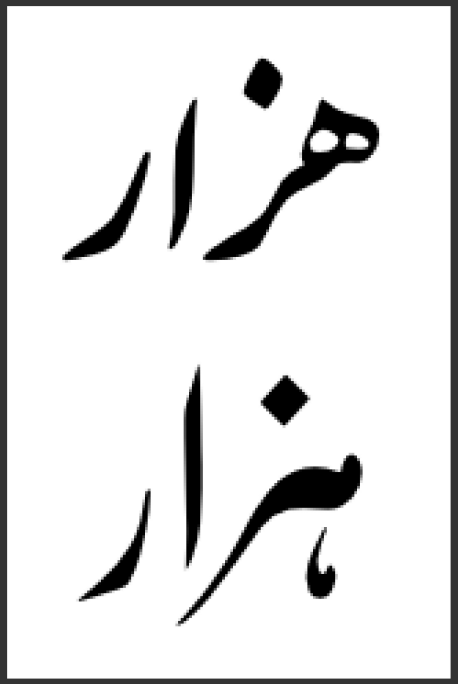
Heh-based Letters

Typical default shapes for…		*Isolate*	*Final*	*Medial*	*Initial*
U+0647	ARABIC LETTER HEH	ه	ـه	ـهـ or ـهٔـ	هـ
U+06BE	… DOACHASHMEE	ھ	ـھ	ـھـ	ھـ
U+06C1	… GOAL	ہ	ـہ	ـہـ	ہـ
U+06FF	… WITH INVERTED SMALL V ABOVE	ۿ	ـۿ	ـۿـ	ۿـ
Urdu					
U+0647	ARABIC LETTER HEH	ہ	ـہ	ـہـ	ہـ
Sindhi					
U+0647	ARABIC LETTER HEH	ھ	ـھ or ـه	ـھـ or ـهـ	ھـ
Parkari					
U+0647	ARABIC LETTER HEH	ھ	ـھ	ـھـ	ھـ
Kurdish					
U+0647	ARABIC LETTER HEH	ھ	ـھ	ـهـ	ھـ

Security and Usability

- Very complicated in multi-lingual environments
- What would be the shape for a string
- How user can type what they see
  - U+0647 ARABIC LETTER HEH (D)
  - U+06D5 ARABIC LETTER AE (R)
  - Ex: ئەگەر
- ICANN IDN Variants Issue Project
  - Same shape in at least one joining form (11 groups)
  - Same Shape in Composed and Decomposed forms (>70)
  - Tah-based letters

Heh in Nasta'liq



Different Writing Styles

- Properties of letter shapes are consistent in different writing styles
- Most common fonts have only 2 glyphs for ALEF, but a Nasta'liq font may have more than 20 glyphs
- Should consider all styles for security and usability

The Abstract Model

Concepts

- Define “Shape” in parallel with “Character”
- Shapes are building blocks of the script
  - How they teach it in school
- They exist only by name
  - May use code-points to encode
- Let's show them as S+<shape-name>
  - S+BehInit, S+SeenMedi, S+NoonFina, S+YehIsol
  - S+DotAbove, S+ThreeUpwardDotsBelow
- Defined in ShapesData.txt

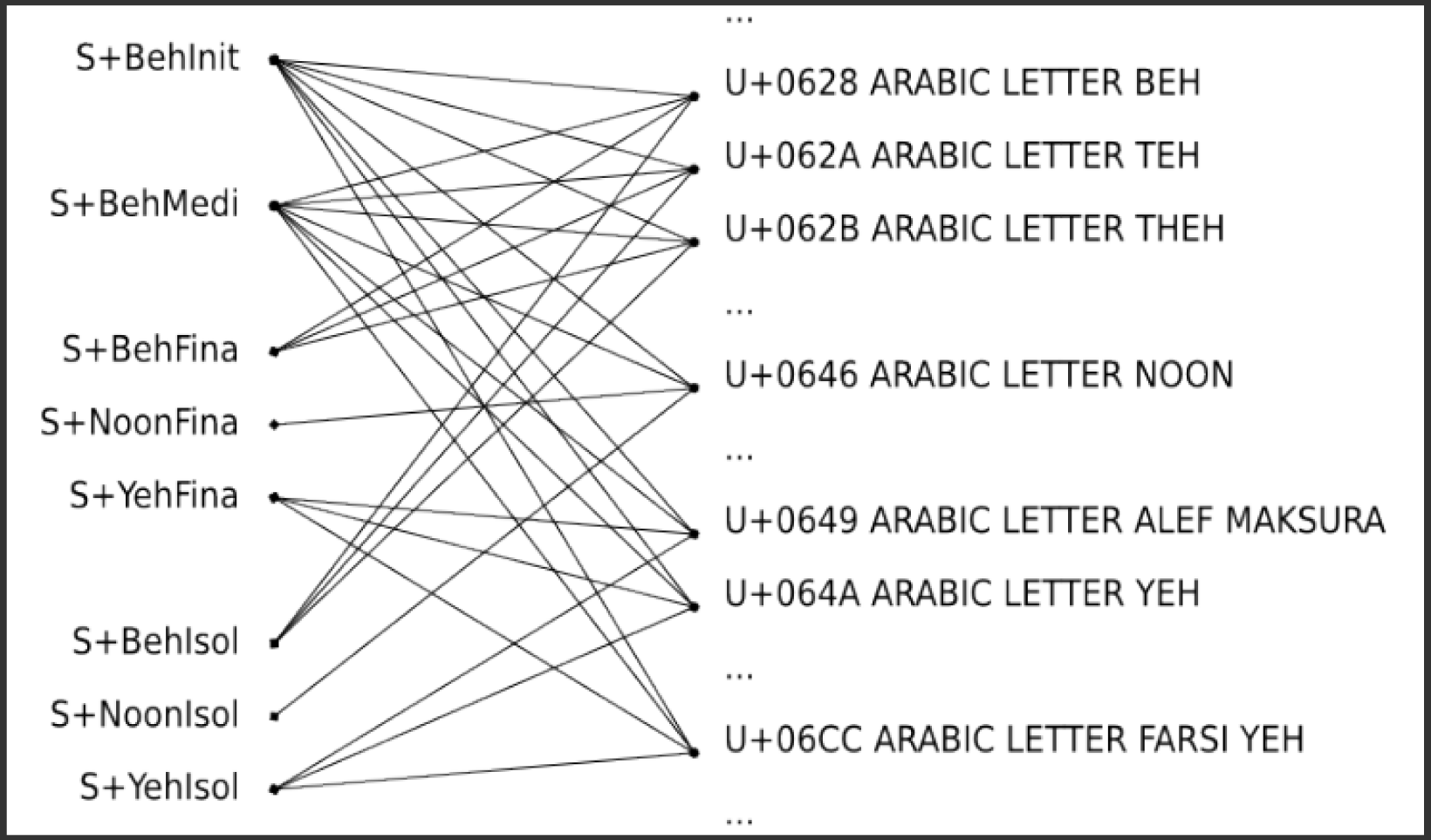
Character Shapes

- Arabic Contextual Joining Algorithm
  - Non-Joining letters 1 form
  - Right-Joining letters 2 forms
  - Dual-Joining letters 4 forms
- One BaseShape for each letter form
- Zero or more AuxShapes
- Ex: U+064A ARABIC LETTER YEH
  - Isol: [S+YehIsol, S+TwoDotsBelow]
  - Fina: [S+YehFina, S+TwoDotsBelo]
  - Init: [S+BehInit, S+TwoDotsBelow]
  - Medi: [S+BehMedi, S+TwoDotsBelow]
- Defined in CharacterShapes.txt

Base Shapes

- Four groups of BaseShapes
- Group “Isol”, 24 shapes
  - ا, ر, و, د, ب, ں, ی, ے, ل, ﻟ, ک, ح, س, ص, ط, ع, ف, ق, م, ﻣ, ہ, ھ, ۂ, ء
- Group “Fina”, 23 shapes
  - ا, ر, و, د, ب, ں, ی, ے, ل, ئ, ک, ح, س, ص, ط, ع, ف, ق, م, ﻣ, ہ, ھ, ۂ
- Group “Init”, 11 shapes
  - ر, ل, ک, ح, س, ص, ط, ع, و, م, ھ
- Group “Medial”, 12 shapes
  - ر, ل, ک, ح, س, ص, ط, ع, ہ, م, ھ, ھـ

Base Shapes vs. Characters



Auxiliary Shapes

- Three groups of AuxShapes
- Group “Above”, 28 shapes
  S+SarkeshAbove S+StrokeAbove S+RingAbove S+DotAbove S+TwoDotsAbove S+TwoVerticalDotsAbove S+ThreeDotsAbove S+ThreeDownwardDotsAbove S+FourDotsAbove S+VAbove S+InvertedVAbove S+MaddaAbove S+AlefAbove S+HamzaAbove S+WavyHamzaAbove S+WaslAbove S+TahAbove S+ShaddaAbove S+DigitTwoAbove S+DigitThreeAbove S+DigitFourAbove S+FathaAbove S+KasraAbove S+DammaAbove S+FathatanAbove S+KasratanAbove S+DammatanAbove S+SukunAbove
- Group “Below”, 12 shapes
  S+StrokeBelow S+RingBelow S+DotBelow S+TwoDotsBelow S+TwoVerticalDotsBelow S+ThreeDotsBelow S+ThreeUpwardDotsBelow S+ThreeHorizontalDotsBelow S+FourDotsBelow S+VBelow S+InvertedVBelow S+CommaBelow S+AlefBelow S+HamzaBelow S+WavyHamzaBelow S+TahBelow S+DigitFourBelow S+KasraBelow S+KasratanBelow
- Group “End”, 1 shape
  S+TailEnd

Shapes Sequence

- Shapes Seq for a Unicode string
  - Arabic Contextual Joining
  - A seq of characters in specific joining forms
  - Concatenate the shapes
- Ex: “یونیکد”
  - [FARSI YEH, WAW, NOON, FARSI YEH, ZWNJ, KEHEH, DAL]
  - [S+BehInit, S+TwoDotsBelow, S+BehInit, S+DotAbove, S+YehFina, S+KehInit, S+DalFina]

Alternate Shapes

- Based on language and/or style
- Ex: U+0647 ARABIC LETTER HEH
  - Normal
    - S+HehIsol
    - S+HehFina
    - S+HehInit
    - S+HehMedi
  - Iranian Nasta'liq
    - S+HehIsol
    - S+HehFina
    - S+BehInit CommaBelow
    - S+BehMedi CommaBelow

The Shape Distant

Shape Distant

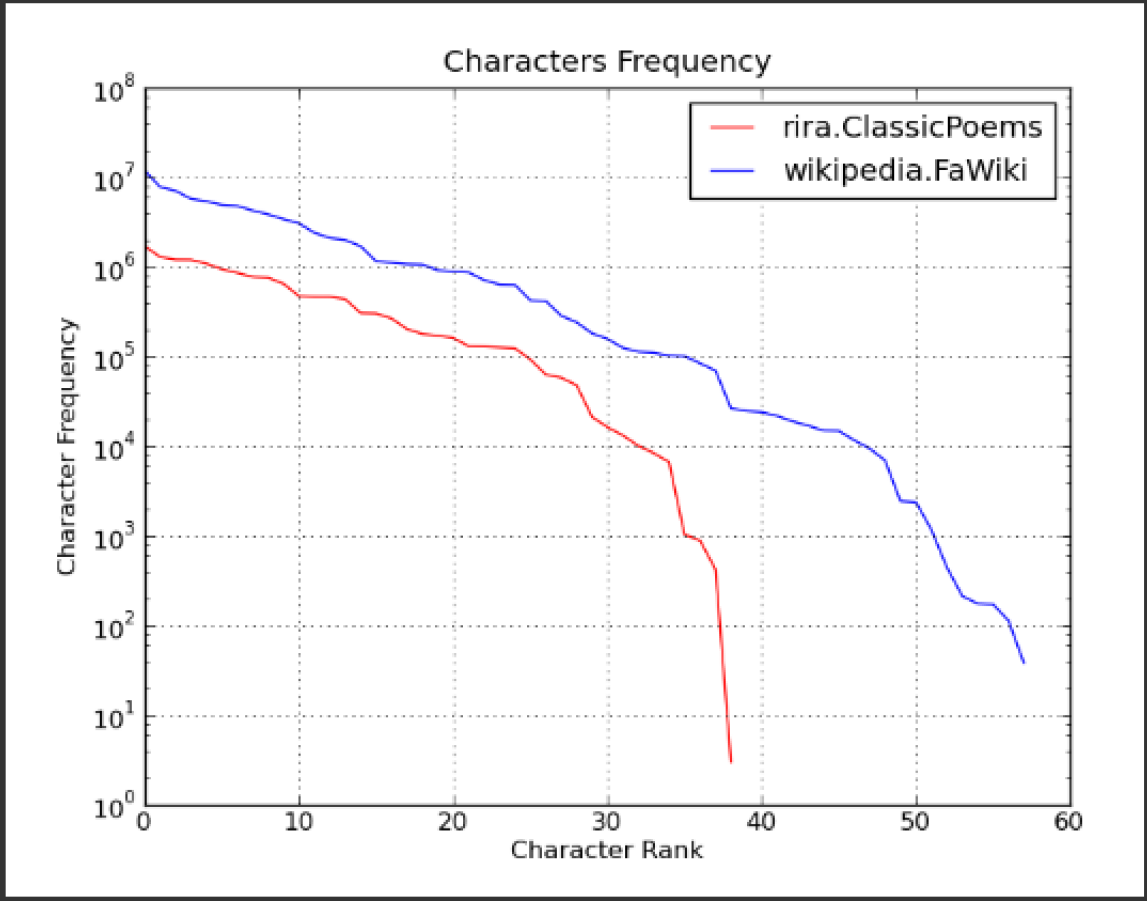
- A metric distant
- Based on Levenshtein distance
- Compares the Shapes Seqs for two strings
- BaseShapes weigh more than AuxShapes
- Also use alternate Shapes Seqs
- May be customized for a specific style

Proposal

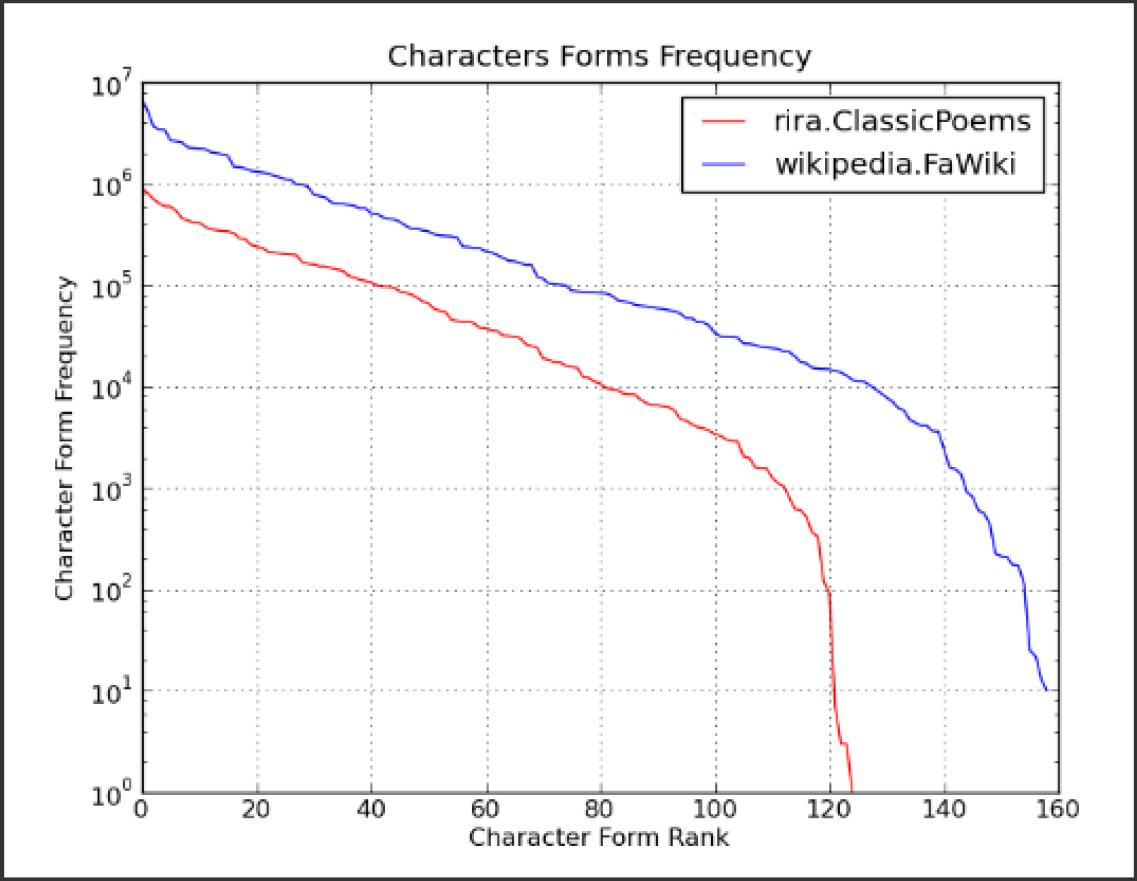
- Unicode Technical Note
- Data files
  - ShapesData.txt
  - CharacterShapes.txt
- Converting Unicode string to/from Shapes
- How to use the Shapes seq to compute similarities
- Should be usable by ICANN, IETF, etc

Corpus Analysis

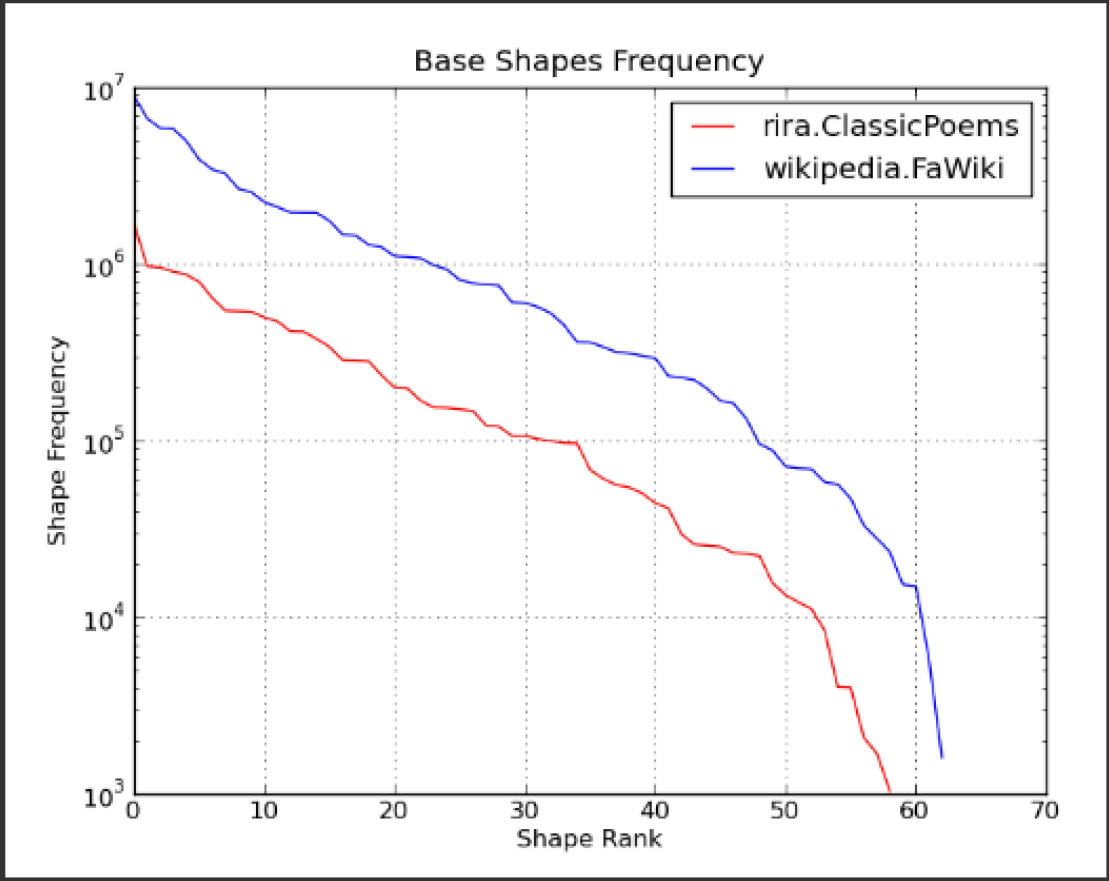
Characters Frequency



Characters Forms Frequency



Shapes Frequency



Thank You!