3. Line justification.

3.1. Locate word bounds (using ICU break iterator).

3.1.1. For Arabic runs additionally look for kashida justification points in each word.

3.2.1. If shaped successfully, insert as many kashida clusters as possible.

3.2.2. Elongate (change advance) or insert spaces in the rest of word bounds.

This won't work with nastaliq

Some fonts can display manual
kashida variants as ligatures.

Something much more complex
should be used for automatic
justification?

Some fonts have broken kashida
glyph.

There should be user accessible
option to disable kashida
justification manually.

Script: Arab

Justifying Text using Cascading Style Sheets (CSS) in Internet Explorer 5.5 (via Wayback Machine)

Overview of the other justification techniques

Find the priority of the connecting opportunities in each word
Add expansion at the highest priority connection opportunity
If more than one connection opportunity have the same highest value, use the opportunity closest to the end of the word.

Priority Glyph Condition
1 User Inserted The user entered a Kashida in a position.
2 Seen and Sad Connecting to the next character.
3 Taa Marbutah, Haa, Dal Connecting to previous character.
4 Alef, Tah Lam and Caf and Gaf Connecting to previous character.
5 RA, Ya and Alef Maqsurah Connected to medial BAA
6 WAW, Ain, Qaf and Fa Connecting to previous character.
7 Other connecting characters Connecting to previous character.

Kashida Location

1 After the user inserted kashida
2 After the character.
3 Before the final form of these characters.
4 Before the final form of these characters.
5 Before preceding medial BAA
6 Before the final form of these characters.
7 Before the final form of these characters.

Script: Arab

Justifying Text using Cascading Style Sheets (CSS) in Internet Explorer 5.5 (via Wayback Machine)