RESEARCH ARTICLE

ANTECEDENTS OF INFORMATION SYSTEMS SOURCING STRATEGIES IN U.S. HOSPITALS: A LONGITUDINAL STUDY

Corey M. Angst, Kaitlin D. Wowak, Sean M. Handley, and Ken Kelley
IT, Analytics, and Operations Department, University of Notre Dame, Mendoza College of Business,
Notre Dame, IN 46556 U.S.A.
{cangst@nd.edu} {katie.wowak@nd.edu} {shandley@nd.edu} {kkelley@nd.edu}

Appendix

Levenshtein Distance and adist

Levenshtein distance is a measure of the similarity between two strings. The R function, *adist,* calculates Levenshtein distance by computing the minimal possibly weighted number of insertions, deletions and substitutions needed to transform one string into another. The function as implemented in R is

adist(x, y = NULL, costs = NULL, counts = FALSE, fixed = TRUE, partial = !fixed, ignore.case = FALSE, useBytes = FALSE)

where

- *x* = a character vector. Long vectors are not supported.
- *y* = a character vector, or NULL (default) indicating taking x as y.
- *costs* = a numeric vector or list with names partially matching insertions, deletions and substitutions giving the respective costs for computing the Levenshtein distance, or NULL (default) indicating using unit cost for all three possible transformations.
- *counts* = a logical indicating whether to optionally return the transformation counts (numbers of insertions, deletions and substitutions) as the "counts" attribute of the return value.
- *fixed* = a logical. If TRUE (default), the x elements are used as string literals. Otherwise, they are taken as regular expressions and partial = TRUE is implied (corresponding to the approximate string distance used by agrep with fixed = FALSE.
- *partial* = a logical indicating whether the transformed x elements must exactly match the complete y elements, or only substrings of these. The latter corresponds to the approximate string distance used by agrep (by default).
- *ignore.case* = a logical. If TRUE, case is ignored for computing the distances.
- *useBytes* = a logical. If TRUE distance computations are done byte-by-byte rather than character-by-character.

Example:

Distance from AAAAB to AAAAA. Number of substitutions = 1 (substitute B for A). Then for the *Adjusted* score, we normalize by dividing *adist* by the number of modules adopted (ranges from 2 to 5) and subtract it from 1. Thus, the score is 1 – 1/5 = 0.80. The *unadjusted* distance always divides by 5, so in this example, both are 1 – 1/5 = 0.80.

Adjusted distance from CAAB– to AAAAA. Number of substitutions = 2 (substitute C and B for A), so the score is 1 – 2/4 = 0.50; Note that because the sequence length is 4 (i.e., one missing), there is no insertion.

Unadjusted distance from CAAB– to AAAAA. Number of substitutions = 2 (substitute C and B for A), Number of insertions= 1 (from missing [–] to A), so the score is 1 – 3/5 = 0.40.

Table A1. Actual Sequences Used by Hospitals and Associated Closeness Scores

Actual Sequence	Adjusted Closeness	Unadjusted Closeness	Actual Sequence	Adjusted Closeness	Unadjusted Closeness	Actual Sequence	Adjusted Closeness	Unadjusted Closeness
AAAAA[1]	1.00	1.00	-A-BA	0.67	0.40	A-BZ-	0.33	0.20
AAAA-	1.00	0.80	ABAB-	0.50	0.40	AC-B-	0.33	0.20
A-AAA	1.00	0.80	ABAC-	0.50	0.40	-ACB-	0.33	0.20
-AAAA	1.00	0.80	ABBA-	0.50	0.40	ADBC-	0.25	0.20
AAAAB	0.80	0.80	ABBAC	0.40	0.40	AZ- - -	0.50	0.20
AAABA	0.80	0.80	ABCA-	0.50	0.40	A- -Z-	0.50	0.20
AABAA	0.80	0.80	ABZA-	0.50	0.40	-A-Z-	0.50	0.20
ABAAA	0.80	0.80	ACAB-	0.50	0.40	BA- - -	0.50	0.20
BAAAA	0.80	0.80	A-CAB	0.50	0.40	B- -A-	0.50	0.20
AAA- -	1.00	0.60	ACBA-	0.50	0.40	-BA- -	0.50	0.20
AA-A-	1.00	0.60	ADACB	0.40	0.40	-B-A-	0.50	0.20
AA- -A	1.00	0.60	ADCBA	0.40	0.40	- -BA-	0.50	0.20
A-AA-	1.00	0.60	AZ-A-	0.67	0.40	- - -BA	0.50	0.20
A- -AA	1.00	0.60	AZAB-	0.50	0.40	BA-C-	0.33	0.20
-AAA-	1.00	0.60	AZBA-	0.50	0.40	BA-Z-	0.33	0.20
-A-AA	1.00	0.60	BA-A-	0.67	0.40	-BAZ-	0.33	0.20
- -AAA	1.00	0.60	B-AA-	0.67	0.40	-BCA-	0.33	0.20
AAAB-	0.75	0.60	-BAA-	0.67	0.40	BZ-A-	0.33	0.20
AA-AB	0.75	0.60	-B-AA	0.67	0.40	BZCA-	0.25	0.20
AABA-	0.75	0.60	BAAB-	0.50	0.40	CA-B-	0.33	0.20
AACAB	0.60	0.60	BABA-	0.50	0.40	C-AB-	0.33	0.20
ABAA-	0.75	0.60	B-AZA	0.50	0.40	-CAB-	0.33	0.20
AB-AA	0.75	0.60	BBAA-	0.50	0.40	CB-A-	0.33	0.20
ABABA	0.60	0.60	BCAA-	0.50	0.40	-CBA-	0.33	0.20
AZAA-	0.75	0.60	BC-AA	0.50	0.40	Z-A- -	0.50	0.20
BAAA-	0.75	0.60	CAAB-	0.50	0.40	Z- -A-	0.50	0.20
BCAAA	0.60	0.60	-CAAB	0.50	0.40	-Z-A-	0.50	0.20
CAAAB	0.60	0.60	CBAA-	0.50	0.40	- -ZA-	0.50	0.20
ZAAA-	0.75	0.60	CDAAB	0.40	0.40	ZA-B-	0.33	0.20
AA- - -	1.00	0.40	ZA-A-	0.67	0.40	-Z-AB	0.33	0.20
A-A- -	1.00	0.40	Z-AA-	0.67	0.40	ZA-BC	0.25	0.20
-AA- -	1.00	0.40	ZABA-	0.50	0.40	Z-AZ-	0.33	0.20
-A-A-	1.00	0.40	ZBAA-	0.50	0.40	ZAZZ-	0.25	0.20
- -AA-	1.00	0.40	ZZAA-	0.50	0.40	ZB-A-	0.33	0.20
- - -AA	1.00	0.40	A- - - -[2]	1.00	0.20	ZBCA-	0.25	0.20
AA-B-	0.67	0.40	-A- - -	1.00	0.20	ZB-ZA	0.25	0.20
A-AB-	0.67	0.40	- -A- -	1.00	0.20	ZZA- -	0.33	0.20
A- -AB	0.67	0.40	- - -A-	1.00	0.20	ZZ-A-	0.33	0.20
AABB-	0.50	0.40	AB- - -	0.50	0.20	ABCDE[3]	0.20	0.20
AA-BC	0.50	0.40	A-B- -	0.50	0.20	ZZBA-	0.25	0.20
AAZ- -	0.67	0.40	A- -B-	0.50	0.20	ZZZA-	0.25	0.20
A-AZ-	0.67	0.40	-A-B-	0.50	0.20	Z- - - -	0.00	0.00
ABA- -	0.67	0.40	- -AB-	0.50	0.20	Z- -Z-	0.00	0.00
AB-A-	0.67	0.40	AB-C-	0.33	0.20	-Z-Z-	0.00	0.00

[1] We define this as the “target” prototypical single-sourcing sequence.
[2] While this sequence was present in our dataset, it was not used in the analysis because empirically we do not view the sourcing decision of a single module as being indicative of a sourcing strategy.
[3] We define this as a prototypical multisourcing sequence.

Table A2. Descriptives for *Unadjusted* Closeness to Single-Sourcing Variable

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013
Mean (StdDev)	0.439 (0.209)	0.514 (0.260)	0.593 (0.283)	0.632 (0.286)	0.691 (0.272)	0.720 (0.274)	0.736 (0.271)	0.790 (0.274)	0.820 (0.253)

Table A3. Mixed Effects Regression Results with *Unadjusted* Closeness to Single-Sourcing as DV

		Model Number and Description					
	Parameter	**1 Intercept Only**	**2 + Slope**	**3 + Strategic Orientation**	**4 + Formal Structure**	**5 + Internal Dynamics**	**6 + Interactions**
	Intercept	0.686*** (0.003)	0.515*** (0.004)	0.545*** (0.005)	0.541*** (0.005)	0.533*** (0.017)	0.424*** (0.026)
Control Variables	Slope (YearSince2005)		0.041*** (0.001)	0.041*** (0.001)	0.041*** (0.001)	0.042*** (0.001)	0.068*** (0.005)
	For-Profit (FP)			-0.103*** (0.007)	-0.085** (0.007)	-0.058*** (0.008)	-0.024* (0.010)
	Teaching (Teach)			0.069*** (0.013)	0.034* (0.014)	0.039* (0.014)	0.019 (0.018)
	Hospital Size (ln_StfBed)				0.016*** (0.003)	0.010* (0.004)	-0.014* (0.006)
	System Size (ln_SysSize)				-0.015*** (0.001)	-0.014*** (0.002)	-0.004† (0.002)
	Hospital Age (lnAge)				0.005 (0.004)	0.009* (0.004)	0.018*** (0.005)
	Case Complexity (CMI)					-0.000 (0.013)	0.074*** (0.019)
	lnAge × YearSince2005						-0.003*** (0.0009)
Interactions of Interest	FP × YearSince2005						-0.008*** (0.002)
	Teach × YearSince2005						0.004 (0.003)
	ln_StfBed × YearSince2005						0.006*** (0.001)
	ln_SysSize × YearSince2005						-0.003*** (0.0004)
	CMI × YearSince2005						-0.017*** (0.003)
	Pseudo-R^2	.528	.788	.786	.786	.770	.771
	Model Comparison		2 vs. 1 $\chi^2(3) = 12410$***				6 vs. 5 $\chi^2(6) = 168$***
	n-Hospitals	3,417	3,417	3,322	3,318	2,824	2,824
	n-Observations	29,033	29,033	28,241	28,210	22,809	22,809
	LL	-894	5305	5854	5926	5077	5126

†$p < 0.10$, *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$