Computational experiments in Adamawa sub-classification

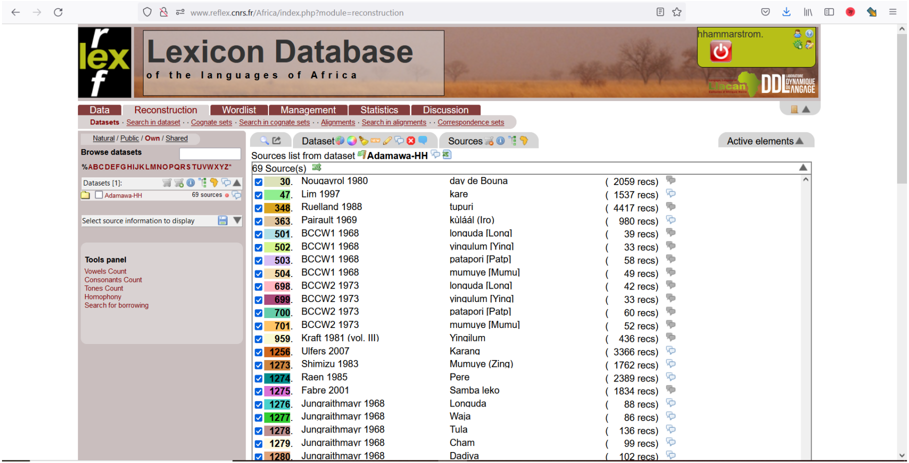
Harald Hammarström
Guillaume Segerer

6 Nov 2021

Today

- Computational Subgrouping and Reconstruction
  - Fully automatic, bottom-up and transparent
  - Does **not** supersede manual work, hopefully **assists** it
- Adamawa (Boyd 1989, Güldemann 2018, Kleinewillinghöfer 2014a, 2020)
  - About a dozen microgroups
  - Possible internal relationships largely undetermined
  - NB: This exercise sees only Adamawa, it cannot prove or disprove the unity of Adamawa or its place(s) within Niger-Congo

Data from RefLex (Segerer 2016)

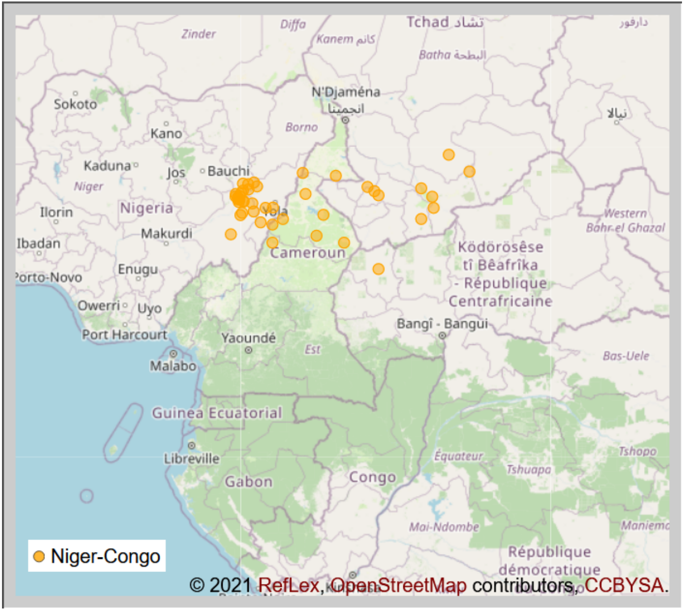


43 Adamawa doculects

Lexical data with unified translation

Language	# items	Language	# items	Language	# items
dadiya	99	goundo	225	mono	794
tula	1033	kim	226	longuda	135
waja	72	mumuye	992	tiba	602
awak	105	gimnime	1262	burak	397
dijim-bwilim	75	momi	1221	loo	306
gula iro	603	samba leko	1439	mághdì	307
bolgo	649	dii (yag dii)	2031	mak	313
noy	48	peere	1705	kyak	312
niellim	48	yendang	345	moo	308
tunia	401	bali	339	leelau	308
kam	1254	yoti	345	dza	309
fali sud	1969	kpasam	343	mingang doso	312
yingilum	458	karang	1689	tha	305
day	1397	tupuri	1133		
besme	226	kare	912		

43 Adamawa doculects: Map



13 Adamawa “Microgroups”

13 of up to 17 microgroups (Kleinewillinghöfer 2020:223)
[Bena-Mboi, Baa, Nimbari, Gueve-Duli not featured]

Language	# items	Language	# items	Language	# items
dadiya	99	goundo	225	mono	794
tula	1033	kim	226	longuda	135
waja	72	mumuye	992	tiba	602
awak	105	gimnime	1262	burak	397
dijim-bwilim	75	momi	1221	loo	306
gula iro	603	samba leko	1439	mághdì	307
bolgo	649	dii (yag dii)	2031	mak	313
noy	48	peere	1705	kyak	312
niellim	48	yendang	345	moo	308
tunia	401	bali	339	leelau	308
kam	1254	yoti	345	dza	309
fali sud	1969	kpasam	343	mingang doso	312
yingilum	458	karang	1689	tha	305
day	1397	tupuri	1133		
besme	226	kare	912		

Data Snapshot Example

file:///C:/Python38/westermann_tula-waja1.html

Language	Items
dadiya	99
tula	1033
waja	72
awak	105
dijim-bwilim	75
Total	1105

Details

Download .tab

Show 10 entries

Search:

Item	awak	dadiya	dijim-bwilim	tula	waja	# lgs
arbre	tī/tī/pūtɔ́	tīyà/tīyà/tìn	ríy/áŋ/rī:/té	tīyāŋ/tī:nī	sóū/sóūrí	5
bouche	ɲī	ɲīyɔ̰̀	ɲí/ɲùní	yī/yī:nī	ɲíyá/ú/ɲíyá/ndí	5
cheval	tówá	tōwà/tōwà/tìn	gə̄rmā/r/gə̄rmā/tē	tʷě/tʷētī	gwɛ́rɛ́/gwíyándí	5
cheveu	yītēhú	yītīn	yīr	rī:	kó/ó/kò:/kə́nà	5
dent	nūŋún	nūŋùn	núŋ/ún/nūŋ/té	nū:m	nʷí/nʷíyéŋŋé	5
eau	mɔ̰́ɛ̰́	mɔ̰̄ɛ̰̄	gə̄mɛ̰́	mʷè	gúndú	5
enfant	bʷɛ́/nābīyāŋ	bʷɛ̀/sə̀n/bɛ̀:/tì	bū/ɛ̄/bū/túbī	bʷɛ́/bʷɛ́sɔ́n	bʷāl/bùyà/kʷàrá	5
front	də̄gādū	tɛ́:kə̄gā	ká/k/ká/té	tīkābū/tīnīkābīnī	ɲátú/ɲátú/ndí	5
fumée	kūdú	yū:lə̀n	yɔ̄àn	ywə̀n	sí:/má	5
homme	bā/ré/bā/tə́m	nə̄/l/nə̄/b	ní:/nə̄/bɛ́	kʷārtā	nɛ́r/ɔ́/nùwá	5

Search Item | Search awak | Search dadiya | Search dijim-bwilim | Search tula | Search waja | Search # lgs

Showing 1 to 10 of 1,105 entries

Previous 1 2 3 4 5 … 111 Next

Automatic Reconstruction

1. Start from parallel wordlists
2. Shallow cognate detection
3. Find shallowest subgroup
4. Sound change extraction for this subgroup
5. Reconstruct proto-language for this subgroup
6. Repeat

Somewhat novel (unpublished) methods for several steps, still under development …

Cognate Detection

Given meaning-aligned wordlists judge which word-forms are historically related

English	Turkish	Persian	Kurdish	Arabic	Hindi	Swedish
wʌn	bir	yek	yek	wæːħed	ek	en
tuː	iki	do	dû	etneːn	doː	tvoː
θri	ytʃ	se	sê	tælæːtæ	tiːn	treː
neɪm	isim/ad	esm	naw	ʔesm	naːm	namn
noʊs	burun	damaːgh	lût	mænæxiːr	naːk	nɛːsa
watər	su	aːb	aw	mɑjjɑ	paːniː	vaten
hed	baʃ/kafa	sar	ser	rɑːs	sar	hʉːvʊd
naɪt	gedʒe	ʃab	ʃev	leːlæ	raːtriː	nat
boʊn	kemik	ostokhaːn	hestî	ʕɑdm	haḍḍiː	beːn
nɪuː	yeni	naw/taːze	nwê	gediːd	nayaː	ny
wiː	biz	maː	ême	eħnæ	ham	viː

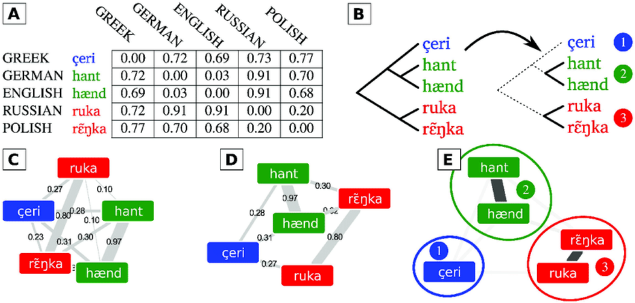
For today, let us conveniently ignore some complications

- Non-monomorphemic forms
- Meaning shift
- …

Cognate Detection: State-of-the-Art

Nearly all past work in automated cognate detection (e.g., List et al. 2018, List 2014, Kondrak 2009, Steiner et al. 2011, List et al. 2017, St Arnaud et al. 2017 and references therein)

1. Align words phonetically
2. Compute similarity of aligned words
3. Group cognates that exceed a certain similarity threshold



Thresholds in Cognate Identification

Require tuning a threshold to cut a similarity-based score into a yes/no cognate decision

Dataset	Words	Conc.	Lang.	Cog.	Div.
Austronesian (Greenhill et al., 2008) [1]	4358	210	20	2864	0.64
Bai (Wang, 2006) [27]	1028	110	9	285	0.19
Chinese (Hóu, 2004) [28]	2789	140	15	1189	0.40
IndoEuropean (Dunn, 2012) [2]	4393	207	20	1777	0.38
Japanese (Hattori, 1973) [29]	1986	200	10	460	0.15
ObUgrian (Zhivlov, 2011) [30]	2055	110	21	242	0.07
TOTAL	16609	977	95	6817	0.30

doi:10.1371/journal.pone.0170046.t002

"The key parameter we need to estimate is the best thresholds for cognate identification in some of the methods" (List et al. 2017:3)

The Threshold is the Problem

- The threshold can either be human-tuned or pre-trained with respect to some supervision/gold standard data set
- Cognate detection and evaluation is typically done on data sets which include both shallow cognates and deep cognates
  - Shallow cognate: German 'fünf' vs English 'five'
  - Deep cognate: Prasuni 'wuču' vs Sardinian 'chimbe'
- Dilemma
  - Strict threshold: Only shallow cognates are found
  - Loose threshold: Junk is found (along with shallow and deep cognates)

First Step Cognate Detection (FSCD)

- Suppose you do not already know
  - The relevant sound-shifts
  - The classificatory tree of the input languages

  Let's call this variant **First Step Cognate Detection**

- For a solution to be possible (whether for a human or machine cognate detector), one has to assume that cognates are more similar *on average* than non-cognates

$$\frac{\sum_{x \neq y \in C_i} Sim(x,y)}{|\{(x,y)|x \neq y \in C_i\}|} > \frac{\sum_{x \neq y \notin C_i} Sim(x,y)}{|\{(x,y)|x \neq y \notin C_i\}|}$$

  Let's call this property the **Similarity Criterion**

I Propose

- **Shallow** first step cognate detection
  - Can be done
  - Can be done without a threshold
  - Shallow cognate = obeys the similary criterion
- **Deep** first step cognate detection
  - Cannot be done
  - (Deep cognate detection must thus be done in several steps or with more information)
  - Deep cognate = does not obey the similary criterion

Threshold-Free FSCD

- Thanks to the similarity criterion, there exists an optimization solution that maximizes

$$\frac{\sum_{x \neq y \in C_i} Sim(x,y)}{|\{(x,y)|x \neq y \in C_i\}|} - \frac{\sum_{x \neq y \notin C_i} Sim(x,y)}{|\{(x,y)|x \neq y \notin C_i\}|}$$

- The intuition is to contrast the cost of judging something cognate (penalty: dissimilarity) and judging something not cognate (penalty: similarity)
- Afaik, the only cognate detection paper in the literature that exploits this dichotomy is Ellison (2007)

  This formulation is restricted to the case with exactly two input languages

The Present Approach

1. Input: Set of n word forms with the same meaning
2. Pairwise Similarity: Calculate the pairwise similarity between each pair of the n words using a suitable similarity measure $S(x, y)$
3. Signficance Similarity: Measure the significance $SS(x, y)$ of the similarity $S(x, y)$ by comparing $S(x, y)$ to $S(x, z)$ for random strings z of the same length
4. Divide the n forms into subsets such that the average $SS(x, y)$ internally in a cognate set $+$ average $1 - SS(x, y)$ between non-cognates is maximized ($=$ correlation clustering)

Notes on Form Similarity

- Form similarity is calculated with Edit Distance using phonetically informed weights (Mortensen et al. 2016)
- Tones are represented as separate phonemes (following their host)
    - Loo 'marcher' *wēlé* is represented as *w e ¯ l e ´*
- Form similarity (and cognacy) is assessed separately for alternative forms for the same meaning
- Form similarity (and cognacy) is assessed separately for polymorphemic forms marked as such in the input (space or dash)
- Meaning change conveniently ignored

Example Phonetic Distances

file:///C:/Python38/westermann_tula-waja1.html

IPA		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	r	s	t	u	w	y	~	ŋ	œ	ǹ	ɐ	ɓ	ɔ	ɗ	ə	ɛ	ɩ	ɲ	ʃ	ʉ	ʊ
	-	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977
a	0.977	-	0.881	0.885	0.883	0.575	0.871	0.889	0.830	0.850	0.803	0.893	0.844	0.857	0.859	0.950	0.885	0.844	0.871	0.887	0.950	0.826	0.950	0.977	0.865	0.375	0.859	0.875	0.885	0.675	0.887	0.575	0.275	0.977	0.857	0.871	0.950	0.950
b	0.977	0.881	-	0.777	0.768	0.873	0.336	0.781	0.816	0.881	0.852	0.785	0.822	0.656	0.807	0.889	0.266	0.830	0.787	0.771	0.889	0.859	0.881	0.977	0.820	0.881	0.807	0.881	0.266	0.889	0.771	0.881	0.873	0.977	0.812	0.795	0.885	0.896
c	0.977	0.885	0.777	-	0.779	0.877	0.791	0.762	0.805	0.869	0.832	0.758	0.834	0.816	0.818	0.893	0.773	0.834	0.791	0.775	0.893	0.855	0.885	0.977	0.801	0.885	0.818	0.885	0.781	0.893	0.783	0.885	0.877	0.977	0.672	0.783	0.889	0.885
d	0.977	0.883	0.768	0.779	-	0.875	0.789	0.783	0.818	0.883	0.854	0.787	0.805	0.807	0.656	0.891	0.771	0.812	0.578	0.266	0.906	0.877	0.898	0.977	0.822	0.883	0.789	0.883	0.771	0.891	0.266	0.883	0.875	0.977	0.814	0.625	0.902	0.898
e	0.977	0.575	0.873	0.877	0.875	-	0.863	0.881	0.822	0.575	0.795	0.885	0.836	0.850	0.852	0.700	0.877	0.844	0.863	0.879	0.950	0.818	0.675	0.977	0.857	0.100	0.852	0.600	0.877	0.400	0.879	0.300	-	0.977	0.850	0.863	0.675	0.675
f	0.977	0.871	0.336	0.791	0.789	0.863	-	0.803	0.445	0.871	0.842	0.799	0.812	0.811	0.828	0.879	0.320	0.820	0.770	0.785	0.879	0.850	0.871	0.977	0.842	0.871	0.828	0.871	0.352	0.879	0.793	0.871	0.863	0.977	0.834	0.777	0.875	0.887
g	0.977	0.889	0.781	0.762	0.783	0.881	0.803	-	0.816	0.873	0.836	0.266	0.838	0.820	0.822	0.881	0.785	0.830	0.803	0.787	0.881	0.844	0.889	0.977	0.656	0.889	0.822	0.873	0.785	0.881	0.787	0.873	0.881	0.977	0.797	0.795	0.893	0.873
h	0.977	0.830	0.816	0.805	0.818	0.822	0.445	0.816	-	0.830	0.672	0.812	0.617	0.793	0.795	0.838	0.812	0.648	0.445	0.814	0.854	0.516	0.846	0.977	0.793	0.830	0.795	0.830	0.820	0.838	0.822	0.830	0.822	0.977	0.785	0.791	0.850	0.846
i	0.977	0.850	0.881	0.869	0.883	0.575	0.871	0.873	0.830	-	0.787	0.877	0.844	0.857	0.859	0.950	0.885	0.844	0.871	0.887	0.700	0.811	0.400	0.977	0.850	0.375	0.859	0.875	0.885	0.675	0.887	0.575	0.275	0.977	0.842	0.871	0.400	0.400
j	0.977	0.803	0.852	0.832	0.854	0.795	0.842	0.836	0.672	0.500	-	0.840	0.758	0.828	0.830	0.811	0.855	0.758	0.842	0.857	0.811	0.700	0.803	0.977	0.812	0.803	0.830	0.803	0.855	0.811	0.857	0.803	0.795	0.977	0.805	0.834	0.807	0.803
k	0.977	0.893	0.785	0.758	0.787	0.885	0.799	0.266	0.812	0.877	0.840	-	0.842	0.824	0.826	0.885	0.781	0.834	0.799	0.783	0.885	0.848	0.893	0.977	0.672	0.893	0.826	0.877	0.789	0.885	0.791	0.877	0.885	0.977	0.801	0.791	0.896	0.877
l	0.977	0.844	0.822	0.834	0.805	0.836	0.812	0.838	0.617	0.844	0.758	0.842	-	0.799	0.781	0.852	0.826	0.594	0.793	0.809	0.867	0.602	0.859	0.977	0.814	0.844	0.781	0.844	0.826	0.852	0.809	0.844	0.836	0.977	0.807	0.805	0.863	0.859
m	0.977	0.857	0.656	0.816	0.807	0.850	0.811	0.820	0.793	0.857	0.828	0.824	0.799	-	0.768	0.865	0.672	0.807	0.826	0.811	0.865	0.836	0.857	0.977	0.781	0.857	0.768	0.857	0.793	0.865	0.811	0.857	0.850	0.977	0.773	0.834	0.861	0.873
n	0.977	0.859	0.807	0.818	0.656	0.852	0.828	0.822	0.795	0.859	0.830	0.826	0.781	0.768	-	0.867	0.811	0.789	0.809	0.672	0.883	0.854	0.875	0.977	0.383	0.859	0.750	0.859	0.811	0.867	0.793	0.859	0.852	0.977	0.352	0.820	0.879	0.875
o	0.977	0.950	0.889	0.893	0.891	0.700	0.879	0.881	0.838	0.950	0.811	0.885	0.852	0.865	0.867	-	0.893	0.852	0.879	0.895	0.575	0.803	0.875	0.977	0.857	0.300	0.867	0.400	0.893	-	0.895	0.100	0.400	0.977	0.865	0.879	0.875	0.275
p	0.977	0.885	0.266	0.773	0.771	0.877	0.320	0.785	0.812	0.885	0.855	0.781	0.826	0.672	0.811	0.893	-	0.834	0.783	0.768	0.893	0.863	0.885	0.977	0.824	0.885	0.811	0.885	0.281	0.893	0.775	0.885	0.877	0.977	0.816	0.791	0.889	0.900
r	0.977	0.844	0.830	0.834	0.812	0.844	0.820	0.830	0.648	0.844	0.758	0.834	0.594	0.807	0.789	0.852	0.834	-	0.801	0.816	0.859	0.570	0.859	0.977	0.807	0.852	0.789	0.844	0.834	0.852	0.816	0.844	0.844	0.977	0.807	0.812	0.855	0.852
s	0.977	0.871	0.787	0.791	0.578	0.863	0.770	0.803	0.445	0.871	0.842	0.799	0.793	0.826	0.809	0.879	0.783	0.801	-	0.312	0.895	0.865	0.887	0.977	0.842	0.871	0.809	0.871	0.791	0.879	0.594	0.871	0.863	0.977	0.834	0.297	0.891	0.887
t	0.977	0.887	0.771	0.775	0.266	0.879	0.785	0.787	0.814	0.887	0.857	0.783	0.809	0.811	0.672	0.895	0.768	0.816	0.312	-	0.910	0.881	0.902	0.977	0.826	0.887	0.793	0.887	0.775	0.895	0.281	0.887	0.879	0.977	0.818	0.359	0.906	0.902
u	0.977	0.950	0.889	0.893	0.906	0.950	0.879	0.881	0.854	0.700	0.811	0.885	0.867	0.865	0.883	0.575	0.893	0.859	0.895	0.910	-	0.787	0.600	0.977	0.857	0.575	0.883	0.675	0.893	0.275	0.910	0.375	0.675	0.977	0.865	0.895	0.600	-
w	0.977	0.826	0.859	0.855	0.877	0.818	0.850	0.844	0.516	0.811	0.700	0.848	0.602	0.836	0.854	0.803	0.863	0.570	0.865	0.881	0.500	-	0.795	0.977	0.820	0.811	0.854	0.811	0.863	0.803	0.881	0.811	0.818	0.977	0.828	0.857	0.799	0.795
y	0.977	0.950	0.881	0.885	0.898	0.675	0.871	0.889	0.846	0.400	0.803	0.893	0.859	0.857	0.875	0.875	0.885	0.859	0.887	0.902	0.600	0.795	-	0.977	0.865	0.275	0.875	0.950	0.885	0.575	0.902	0.675	0.375	0.977	0.857	0.887	0.300	0.300
~	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	-	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977
ŋ	0.977	0.865	0.820	0.801	0.822	0.857	0.842	0.656	0.793	0.850	0.812	0.672	0.814	0.781	0.383	0.857	0.824	0.807	0.842	0.826	0.857	0.820	0.865	0.977	-	0.865	0.783	0.850	0.824	0.857	0.826	0.850	0.857	0.977	0.281	0.834	0.869	0.850
œ	0.977	0.375	0.881	0.885	0.883	0.100	0.871	0.889	0.830	0.375	0.803	0.893	0.844	0.857	0.859	0.300	0.885	0.852	0.871	0.887	0.575	0.811	0.275	0.977	0.865	-	0.859	0.400	0.885	0.600	0.887	0.700	0.400	0.977	0.857	0.871	0.275	0.875
ǹ	1.156	0.859	0.807	0.818	0.789	0.852	0.828	0.822	0.795	0.859	0.830	0.826	0.781	0.768	0.750	0.867	0.811	0.789	0.809	0.793	0.883	0.854	0.875	1.156	0.783	0.859	-	0.859	0.811	0.867	0.793	0.859	0.852	1.156	0.775	0.820	0.879	0.875
ɐ	0.977	0.875	0.881	0.885	0.883	0.600	0.871	0.873	0.830	0.875	0.803	0.877	0.844	0.857	0.859	0.400	0.885	0.844	0.871	0.887	0.675	0.811	0.950	0.977	0.850	0.400	0.859	-	0.885	0.100	0.887	-	0.300	0.977	0.857	0.871	0.950	0.375
ɓ	0.977	0.885	0.266	0.781	0.771	0.877	0.352	0.785	0.820	0.885	0.855	0.789	0.826	0.793	0.811	0.893	0.281	0.834	0.791	0.775	0.893	0.863	0.885	0.977	0.824	0.885	0.811	0.885	-	0.893	0.768	0.885	0.877	0.977	0.816	0.799	0.889	0.900
ɔ	0.977	0.675	0.889	0.893	0.891	0.400	0.879	0.881	0.838	0.675	0.811	0.885	0.852	0.865	0.867	-	0.893	0.852	0.879	0.895	0.275	0.803	0.575	0.977	0.857	0.600	0.867	0.100	0.893	-	0.895	0.400	0.700	0.977	0.865	0.879	0.575	0.575
ɗ	0.977	0.887	0.771	0.783	0.266	0.879	0.793	0.787	0.822	0.887	0.857	0.791	0.809	0.811	0.793	0.895	0.775	0.816	0.594	0.281	0.910	0.881	0.902	0.977	0.826	0.887	0.793	0.887	0.768	0.895	-	0.887	0.879	0.977	0.818	0.641	0.906	0.902
ə	0.977	0.575	0.881	0.885	0.883	0.300	0.871	0.873	0.830	0.575	0.803	0.877	0.844	0.857	0.859	0.100	0.885	0.844	0.871	0.887	0.375	0.811	0.675	0.977	0.850	0.700	0.859	-	0.885	0.400	0.887	-	0.600	0.977	0.857	0.871	0.675	0.675
ɛ	0.977	0.275	0.873	0.877	0.875	-	0.863	0.881	0.822	0.275	0.795	0.885	0.836	0.850	0.852	0.400	0.877	0.844	0.863	0.879	0.675	0.818	0.375	0.977	0.857	0.400	0.852	0.300	0.877	0.700	0.879	0.600	-	0.977	0.850	0.863	0.375	0.950
ɩ	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	-	0.977	0.977	0.977	0.977

Example Tonal Distances

w	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	-	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250
:	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	-	0.250	0.250	0.250	0.250	0.250	0.250	0.250
`	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	0.250	-	0.250	0.250	0.250	0.250	0.250	0.250
´	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	0.250	0.250	-	0.250	0.250	0.250	0.250	0.250
^	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	0.250	0.250	0.250	-	0.250	0.250	0.250	0.250
¯	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	0.250	0.250	0.250	0.250	-	0.250	0.250	0.250
ˇ	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	0.250	0.250	0.250	0.250	0.250	-	0.250	0.250
.	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-	0.250
~	0.250	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	1.156	0.977	0.977	0.977	0.977	0.977	0.977	0.250	0.977	0.977	0.977	0.977	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	-

- Tones easily lost
- Tones easily exchanged for other tones
- Tones not easily exchanged for vowels or consonants (needs revision!)

Example Output Cognate Detection

file:///C:/Python38/westermann_tula-waja1.html

1.2 Cognate Identification

Number of meanings	1105
Number of meanings with at least two language forms	131
Number of non-singleton cognate sets	176

Cognate Density	**awak**	**dadiya**	**dijim-bwilim**	**tula**	**waja**
awak	---- ----	65.7% (38.1/58)	54.4% (31/57)	63.5% (52.7/83)	51.0% (13.8/27)
dadiya	65.7% (38.1/58)	---- ----	64.4% (40.6/63)	65.6% (44.0/67)	36.9% (7.7/21)
dijim-bwilim	54.4% (31/57)	64.4% (40.6/63)	---- ----	56.7% (40.8/72)	32.8% (7.5/23)
tula	63.5% (52.7/83)	65.6% (44.0/67)	56.7% (40.8/72)	---- ----	45.5% (23.7/52)
waja	51.0% (13.8/27)	36.9% (7.7/21)	32.8% (7.5/23)	45.5% (23.7/52)	---- ----

Details

Download .tab

Show 10 entries   Search:

Item	awak	dadiya	dijim-bwilim	tula	waja	# lgs	# cognate sets
1sg POS	m	mì				2	1
année	tɛ̄r	jʷœ̄l/tìn	r̂/jī/yī/tē	sʷātū/sʷār		4	4
apporter	pītḗ			bɛ̀kə́n		2	1
arbre	pūtɔ́/tī	tìn/tīyà	áŋ/rī:/té/ríy	tīyāŋ/tī:nī	sóū/sóūrí	5	5
argent				kə̀mbìr/kə̀mbìrī	è/kén	2	3
attendre	kāntɔ́			yām		2	1
barbe				sūbí/sùbɛ̀	sɔ́wɛ̀	2	1
blanc				mʷɛ̄	pòpùlòk	2	2
boire	dé	nō		nɔ̄		3	2
bon				kìyìlàŋ	kùndí	2	1

Example Output Cognate Detection

file:///C:/Python38/westermann_tula-waja1.html

1.2 Cognate Identification

Number of meanings	1105
Number of meanings with at least two language forms	131
Number of non-singleton cognate sets	176

Cognate Density	**awak**		**dadiya**		**dijim-bwilim**		**tula**		**waja**	
awak	----	----	65.7%	(38.1/58)	54.4%	(31/57)	63.5%	(52.7/83)	51.0%	(13.8/27)
dadiya	65.7%	(38.1/58)	----	----	64.4%	(40.6/63)	65.6%	(44.0/67)	36.9%	(7.7/21)
dijim-bwilim	54.4%	(31/57)	64.4%	(40.6/63)	----	----	56.7%	(40.8/72)	32.8%	(7.5/23)
tula	63.5%	(52.7/83)	65.6%	(44.0/67)	56.7%	(40.8/72)	----	----	45.5%	(23.7/52)
waja	51.0%	(13.8/27)	36.9%	(7.7/21)	32.8%	(7.5/23)	45.5%	(23.7/52)	----	----

Details

Download .tab

Show 10 entries

Search:

Item	awak	dadiya	dijim-bwilim	tula	waja	# lgs	# cognate sets
1sg POS	m	mì				2	1
année	tɛ̄r	jʷœ̄l/tìn	r̂/jī/yī/tē	sʷātū/sʷār		4	4
apporter	pīté			bɛ̀kə́n		2	1
arbre	pūtɔ́/tī	tìn/tīyà	áŋ/rī:/té/ríy	tīyāŋ/tī:nī	sóū/sóūrí	5	5
argent				kə̀mbìr/kə̀mbìrī	è/kén	2	3
attendre	kāntɔ́			yām		2	1
barbe				sūbí/sùbɛ̀	sɔ́wɛ̀	2	1
blanc				mʷɛ̄	pòpùlòk	2	2
boire	dé	nō		nɔ̄		3	2
bon				kìyìlàŋ	kùndí	2	1

Example Inspect Cognate Judgment

file:///C:/Python38/westermann_tula-waja1.html

Show 10 entries

Search:

Item	awak	dadiya	dijim-bwilim	tula	waja	# lgs	# cognate sets
1sg POS	m	mì				2	1
année	tēr	jʷœ̄l/tìn	ı̂/jī/yī/tē	sʷātū/sʷār		4	4
apporter	pītê			bèkə́n		2	1
arbre	pūtɔ́/tī	tìn/tīyà	áŋ/rī:/té/ríy	tīyāŋ/tī:nī	sóū/sóūrí	5	5
argent				kə̀mbɨ̀r/kə̀mbɨ̀rī	è/kén	2	3
attendre	kāntɔ́			yām		2	1

attendre

d(x, y)	kāntɔ́	yām
kāntɔ́	-	3.910
yām	3.910	-
--Average	3.910	3.910

SS(x, y)	kāntɔ́	yām
kāntɔ́	1.000	0.637
yām	0.903	1.000
--Average	0.770	0.770

Cluster	Members	Internal Edges
0-1	{kāntɔ́, yām}	0.637, 0.903

From	To	External Edge

Score: 0.770, # clusters: 1

Item	awak	dadiya	dijim-bwilim	tula	waja	# lgs	# cognate sets
barbe				sūbí/sùbɛ̀	sɔ́wɛ̀	2	1
blanc				mʷɛ̄	pòpùlòk	2	2
boire	dé	nō		nɔ̄		3	2

Towards Deep Cognate Detection

- Shallow cognates provide evidence for (shallow) subgrouping
  - Factor out the most recent **subgroup**
  - **Reconstruct** its proto-language via *regular correspondences* found in the shallow cognates
  - Redo (shallow) **cognate detection**, this time with the proto-language of the recognized subgroup instead of the surface forms
- Repeat

This way, deep cognates may be recognized iff surface divergent surface forms become similar by a series of nested regular correspondences

Subgrouping and Reconstruction: Some Heuristic Approaches

- **Subgrouping:** A greedy solution
  - For every meaning, guess which cognate set is the oldest
  - The cognate set shared across the *deepest divide* is most likely the oldest
  - Thus this is the retention, the other cognate sets innovations
  - Once innovations are distinguished from retentions, we can test for the subgroup best selected for by shared innovations
- **Reconstruction:** A greedy solution
  - In every cognate set, try one of the forms as ancestral
  - This gives equations to all modern forms
  - From such equations collect a set of potential sound changes
  - A potential sound change can be tested for significance across all cognate sets
  - Majority vote + play back of significant sound changes provide the reconstruction

Cognate Matrix to Most Demarcated Terminal Subgroup

1. For every meaning, guess which cognate set was present in the proto-language
   - Heuristic: the value cognate set across the deepest divide is the most likely value for the proto-language
2. Throw away the retention & singleton isoglosses
3. Find the *Most Demarcated Terminal* (MDT) subgroup
   - Heuristic: The MDT subgroup is the subset with the highest amount of supportive innovation isoglosses and the least amount of conflicting innovation isoglosses
4. Replace the languages of the MDT subgroup with its protolanguage

Retention vs Innovation

- Which of A, B, C, D are innovations/retention?

	Agei [aif]	Aiku [ymo]	Aro [tei]	Bragat [aof]	Chinapeli [van]	...
two	A	B	A	C	A	

- Across **all** 184 meanings, the overall cognate distances between the languages are

	Agei [aif]	Aiku [ymo]	Aro [tei]	Bragat [aof]	Chinapeli [van]
Agei [aif]	0.0	0.669	0.689	0.701	0.644
Aiku [ymo]	0.669	0.0	0.672	0.666	0.660
Aro [tei]	0.729	0.672	0.0	0.655	0.678
Bragat [aof]	0.701	0.666	0.655	0.0	0.685
Chinapeli [van]	0.644	0.660	0.678	0.685	0.0

- The deepest divide (0.729) is between Aro and Agei which both share the *A* cognate
- Let us therefore guess that *A* is a retention in this case
- That makes *B* and *C* innovations

Innovations to MDT Subgroup

- Throw away retention isoglosses & singleton innovations
- We are now left with a list of *innovation* isoglosses that select various subsets of the languages at hand
- The MDT should be one which has the most unequivocal support isoglosses (the most supporting innovations and the least conflicting innovations)
- Heuristic: For each subset S with at least one innovation
  - Do a Fisher Exact Test (FET) to measure how well each innovation i selects S

$$Subgroup(S,I) = \prod_{i \in I} FET(S,I) = \prod_{i \in I} \sum_{k \geq |S \cap i|} \frac{\binom{|S|}{k}\binom{|L \setminus S|}{|i|-k}}{\binom{|L|}{|i|}}$$

  - Check if it beats what can be expected by random
  - Check that it doesn't have a more recent subgroup within it
- If there is S that beats random and has no more recent subgroup within it, S is the MDT

Reconstruct the MDT Subgroup Proto-Language

- Suppose the Most Demarcated Terminal subgroup is $S = \{L_1, L_2, L_3\}$

	$\overbrace{L1 \quad L2 \quad L3}^{S}$			...	L10
M1	A	A	B	...	B
M2	A	B	C	...	B
...					

- For each meaning
  - Determine which cognate to project to proto-S:
    - Project the most common (in S) cognate set to the proto-language, e.g., for meaning M1 project cognate set A to proto-S
    - In case of a tie, e.g., M2, prefer the cognate set (here B) which is found outside S
  - Reconstruct the form for that cognate in proto-S

    See next slides

Collecting Potential Sound Changes

- Given a set of forms x, y, z, ...
- Assume the proto-sound and proto-condition for every sound change is preserved in at least one modern form
- Then the equations $*x \rightarrow x, *x \rightarrow y, *x \rightarrow z, *y \rightarrow x, *y \rightarrow z, \ldots$ etc encompass **all** relevant potential sound changes
- E.g. with $\{varm, worm, warm\}$, the equations

Ancestral		Modern	Potential sound change(s)
varm	$\rightarrow$	worm	$v > w$, $a > o$, $v- > w-$, $Ca > Co$, ..
varm	$\rightarrow$	warm	$v > w$, $v- > w-$, ...
worm	$\rightarrow$	varm	$w > v$, $o > a$
worm	$\rightarrow$	warm	$o > a$
...			

- I experimented with all uni- and bigram sound changes

Testing Potential Sound Changes

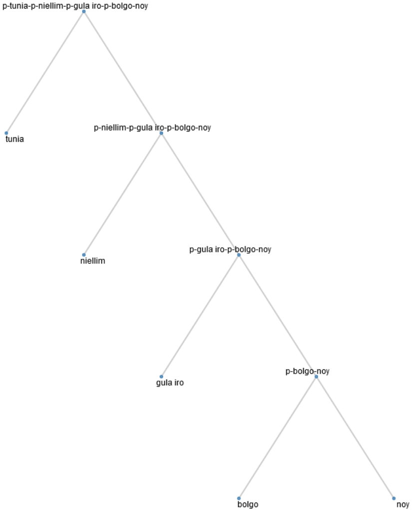
- Reverse-apply the sound change to *all* words
- Check how much the edit distance to its cognates improved/worsened (“gain”)
- If the gain is better than random accept the sound sound change
  - Permutation tests (many variants) can represent the null hypothesis
  - Control for multiple testing of sound changes, e.g., if 560 potential sound changes are checked, an accepted sound change must be better than 560 random ones

Example: Tula-Waja

Tentatively, I assume a Tula (core) group consisting of Tula, Dadiya, and Bangwinji. Ma and Yebu [= Awak — HH] possibly form their own branch to it. Tso on the one hand and Cham [= Dijim-Bwilim — HH], on the other hand appear to be earlier off-shoots from the main group. Unclear is the position of Waja, it seems to be the only member of a distinct branch. (Kleinewillinghöfer 2014b:2)

- Quite different at face value
- Less different under the hood and in the details

Example: Bua



Boyeldieu et al. (2018:60)

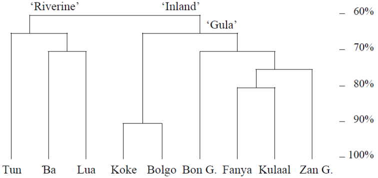


Table 1. Lexicostatistic classification (branch average)

	Bolgo	Koke	Fanya	Bon G.	Zan G.	Kulaal	Lua	Ɓa
Koke	901							
Fanya	742	731						
Bon G.	598	636	733					
Zan G.	677	601	800	717				
Kulaal	660	712	816	678	722			
Lua	568	641	647	523	621	570		
Ɓa	583	595	625	517	583	596	728	
Tun	582	636	627	553	615	607	629	652

Table 2. Lexicostatistics: similarity matrix[12]

- Position of Niellim [= Lua] different
- Lexical similarity matrices quite similar

Discussion

- In the present conceptualization
  - Subgrouping needs cognate information
  - Cognate detection is dependent on subgrouping
- In the present approach, this is done in a greedy see-saw manner (CD1, SG1, CD2, SG2, …)
- Why not go Bayesian?

  Search space is prohibitive already with the tree topology, let alone with branch lengths, cognates judgment and regular sound changes intertwined. Heuristics needed to control the search space in Bayesian formulations. Preferable from a linguistic perspective to have more **transparent** *heuristics than those.*

Conclusion

- Arguments to separate shallow and deep cognate detection
- Good hope that shallow cognate detection can be done, even without recourse to thresholds
- Deep cognate detection adressed via iterative subgrouping and reconstruction
  - Heuristic subgroup detection
  - Heuristic discovery of sound changes
  - Heuristic iterated reconstruction
- Transparent, so hopefully of use to “real” historical linguists working on Adamawa or any other larg-ish set of languages with open classification questions

Boyd, R. (1989). Adamawa-ubangi. In Bendor-Samuel, J., editor, *The Niger-Congo Languages: A Classification and Description of Africa's Largest Language Family*, pages 178–215. Lanham, MD: University Press of America.

Boyeldieu, P., Kastenholz, R., Kleinewillinghöfer, U., and Lionnet, F. (2018). The bua group languages (chad, adamawa 13): A comparative perspective. In Kramer, R. and Kießling, R., editors, *Current approaches to Adamawa and Gur languages*, volume 34 of *Afrika und Übersee: Beiheft*, pages 53–126. Cologne: Rüdiger Köppe.

Ellison, T. M. (2007). Bayesian identification of cognates and correspondences. In *Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*, SigMorPhon '07, pages 15–22. Association for Computational Linguistics, Stroudsburg, PA, USA.

Güldemann, T. (2018). Historical linguistics and genealogical language classification in africa. In Güldemann, T., editor, *African Languages and Linguistics*, pages 58–444. Berlin: DeGruyter Mouton.

Kleinewillinghöfer, U. (2014a). Adamawa. Paper presented at the Linguistisches Kolloquium des Seminars für Afrikawissenschaften, Humboldt-Universität zu Berlin, 4 November, 2014.

Kleinewillinghöfer, U. (2014b). Tula-waja group. Ms.

Kleinewillinghöfer, U. (2020). Adamawa. In Vossen, R. and Dimmendaal, G. J., editors, *The Oxford Handbook of African Languages*, pages 220–230. Oxford: Oxford University Press.

Kondrak, G. (2009). Identification of cognates and recurrent sound correspondences in word lists. *Traitement Automatique des Langues*, 50(2):201–235.

List, J.-M. (2014). *Sequence comparison in historical linguistics*. PhD thesis, Düsseldorf: Heinrich Heine University.

List, J.-M., Greenhill, S. J., and Gray, R. D. (2017). The potential of automatic word comparison for historical linguistics. *PLOS ONE*, 12(1):1–18.

List, J.-M., Walworth, M., Greenhill, S. J., Tresoldi, T., and Forkel, R. (2018). Sequence comparison in computational historical linguistics. *Journal of Language Evolution*, 3(2):130–144.

Mortensen, D. R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., and Levin, L. (2016). Panphon: A resource for mapping ipa segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. The COLING 2016 Organizing Committee.

Segerer, G. (2016). Reflex: la reconstruction sans peine. *Faits de Langues*, 47:201–214.

St Arnaud, A., Beck, D., and Kondrak, G. (2017). Identifying cognate sets across dictionaries of related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2519–2528. Association for Computational Linguistics.

Steiner, L., Stadler, P. F., and Cysouw, M. (2011). A pipeline for computational historical linguistics. *Language Dynamics & Change*, 1:89–127.