Beyond cognacy

Gerhard Jäger, Tübingen University

Computational phylogenetics and language (pre)history: Achievements, challenges and prospects

Ρέθυμνο, May 25, 2023

phylogenetic linguistics

- main goal: infer **phylogenetic trees** from **lexical data**

input

Showing 1 to 38 of 38 entries ← Previous 1 Next →

No.	Meaning	Concepticon	Word	Loan
Search	Search		Search	--any--
1	I	I	n3k	False
2	you	THOU	k3j	False
3	we	WE	nucna	False
11	one	ONE	yan	False
12	two	TWO	zuZ	True
18	person	PERSON	insan	True
19	fish	FISH	amal3h	True
21	dog	DOG	ayda	False
23	tree	TREE	tagig3t	False
25	leaf	LEAF	afraw	False
28	skin	SKIN	lZ3ld	True
30	blood	BLOOD	a83m	False
31	bone	BONE	ax3s	False
34	horn	HORN (ANATOMY)	as3kaw	False
39	ear	EAR	am3zux	False
40	eye	EYE	tit	False
41	nose	NOSE	tax3nfurt	True
43	tooth	TOOTH	asan	False
44	tongue	TONGUE	il3s	False
47	knee	KNEE	afud	False
51	breast	BREAST	sd3r	True
53	liver	LIVER	l3fwad	True
54	drink	DRINK	su	False
57	see	SEE	zar	False
58	hear	HEAR	s3l	False
61	die	DIE	mu8	False

Coordinates WGS84	35°19'N, 4°58'W 35.31, -4.96
number of speakers	10,000
status	alive

Classification

WALS
AA > Berber
Glottolog
Afro Asiatic > Berber > Kabyle Atlasberber > Atlasberber > Northwesternmoroccanberber
Ethnologue
Afro Asiatic > Berber > Northern > Zenati > Ghomara

Sources

El Hannouche 2010
Arabic influence in Ghomara Berber. M.A. thesis, Leiden University.

phylogenetic linguistics

- main goal: infer **phylogenetic trees** from **lexical data**

input

- *important*: cognate classification

example data: `dunnielex` from lexibank:

Out[4]: 21×4 DataFrame

Row	Language_ID	Parameter_ID	Segments	Cognateset_ID
	String15?	String15?	String?	Int64?
1	urdu	180_tooth	d̪ ɑ̃ ʈ̪	328
2	catalan	180_tooth	d e n	328
3	armenianmod	180_tooth	ɑ t ɑ m	328
4	bretonst	180_tooth	d ã n t	328
5	czech	180_tooth	z ʊ p	502
6	german	180_tooth	ts aː n	328
7	italian	180_tooth	d ɛ n t e	328
8	swedish	180_tooth	t a n d	328
9	greekmod	180_tooth	ð o̪ n d i	328
10	marathi	180_tooth	d a t	328
11	polish	180_tooth	z ɔ̃ p	502
12	portuguesest	180_tooth	d ẽ t ɨ	328
13	russian	180_tooth	z u b	502
14	spanish	180_tooth	d j e n t e	328
15	danish	180_tooth	d̪ʰ/dʰ a n	328
16	dutchlist	180_tooth	t ɑ n t	328
17	english	180_tooth	t uː θ	328
18	french	180_tooth	d ɑ̃	328
19	russian	180_tooth	d e s n a	328
20	bihari	180_tooth	d ã t	328
21	oriya	180_tooth	d a n tɔ	328

phylogenetic linguistics

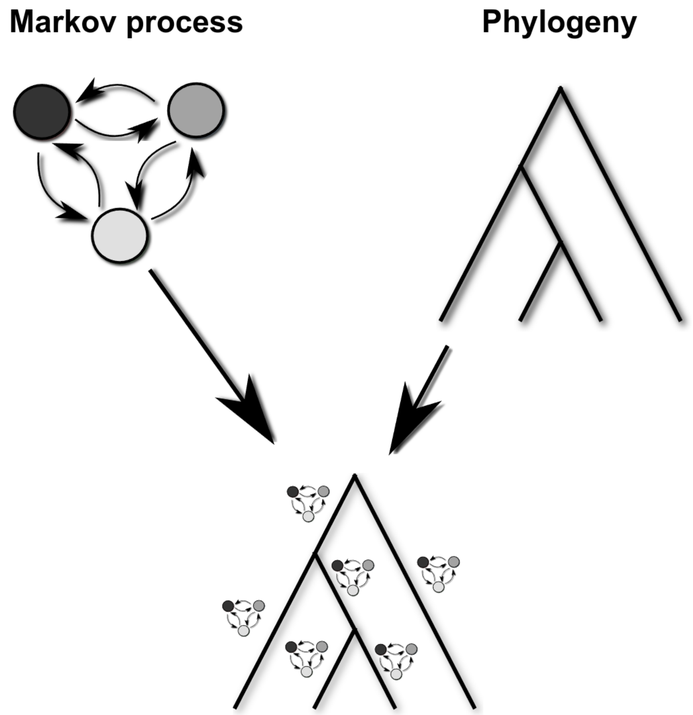
intermediate step

convert cognate classification into binary character matrix

```
TNG.ENGAN.MAIBI                 1000000000000000000000000000000000000000000000
TNG.ENGAN.POLE                  0000000000000000000000000000000000000010000000
TNG.ENGAN.SAU                   0000000000000000000000000000000000000000000000
TNG.ENGAN.YARIBA                1000000000000000000000000000000000000000000000
TNG.FASU.FASU                   0000000000000000000000000000000000000010000000
TNG.FASU.NAMUMI                 0000000000000000000000000000000000000000001000
TNG.FINISTERRE-HUON.AWARA       0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.BORONG      0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.BURUM       0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.BURUM_MIND  0000000000000000000000000000000000001010000000
TNG.FINISTERRE-HUON.DEDUA       0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.HUBE        0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.KATE        0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.KOMBA       0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.KOSORONG    0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MAPE        0000000000000000000000000000000000001000000000
TNG.FINISTERRE-HUON.MAPE_2      0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MIGABAC     0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MINDIK      0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.MOMOLILI    0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.NABAK       0000000000000000000000000000000000001000000000
TNG.FINISTERRE-HUON.NANKINA     0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.NEK         0000000000000000000000000000000000000001000000
TNG.FINISTERRE-HUON.NUKNA       0000000000000000000000000000000000000001000000
TNG.FINISTERRE-HUON.ONO         0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.SELEPET     0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.TIMBE       0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.TOBO        0000000000000000000000000000000000000010000000
TNG.FINISTERRE-HUON.WANTOAT     0000000000000000000000000000000000000110000000
TNG.FINISTERRE-HUON.YOPNO       0000000000000000000000000000000000000001000000
TNG.GOILALAN.AFOA               0000000000000000000000000000000000000110000000
TNG.GOILALAN.KUNIMAIPA          0000000000000000000000000000000000000010000000
TNG.GOILALAN.MAFULU             0000000000000000000000000000000000000010000000
```

phylogenetic linguistics

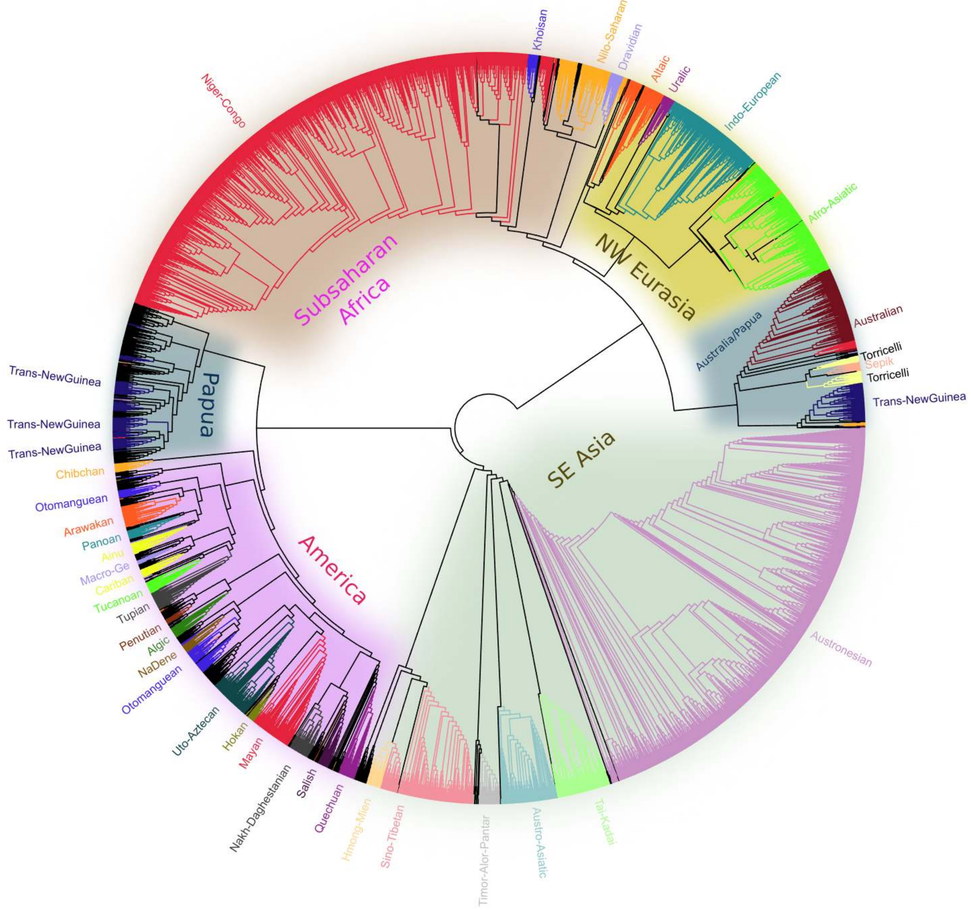
stochastic model: continuous time Markov chain



phylogenetic linguistics

- main goal: infer **phylogenetic trees** from **lexical data**

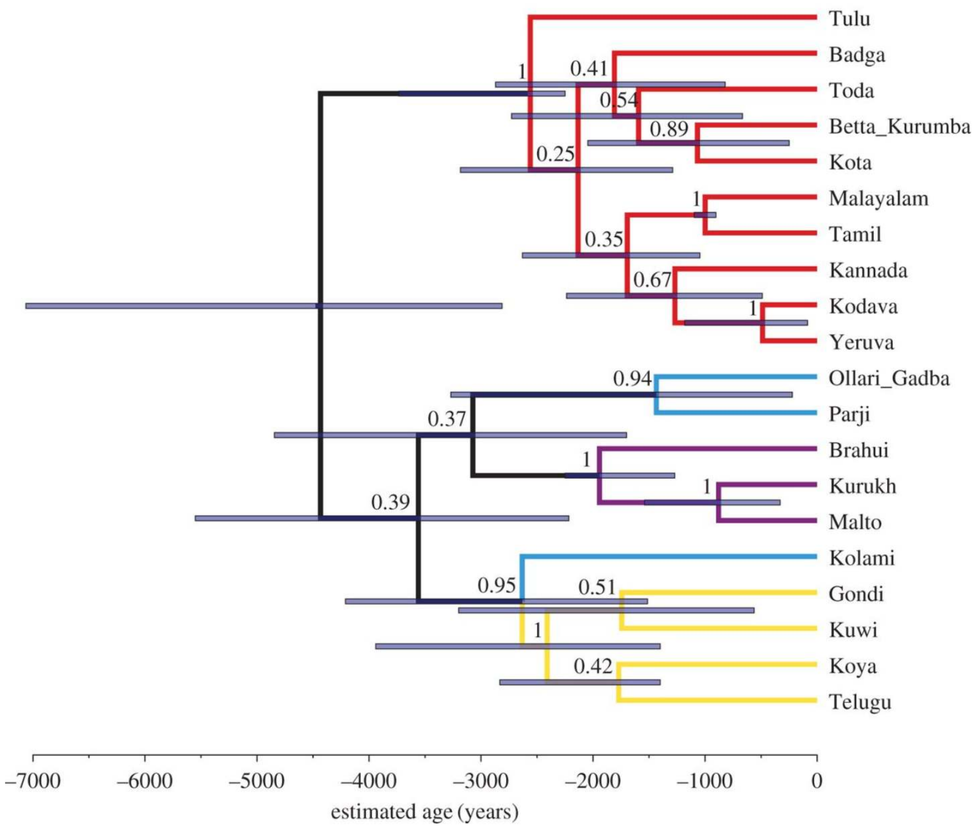
output



phylogenetic linguistics

- main goal: infer **phylogenetic trees** from **lexical data**

output



(source: Kolipakam et al. 2018, https://doi.org/10.1098/rsos.171504)

applications

- control for common ancestry in statistical models *(Jäger & Wahle 2021, ...)*
- estimate time depth and geographic location of ancestral populations (Bouckaert et al 2012)
- reconstruct properties of ancestral populations (Cathcart et al 2021, Carling & Cathcart 2021a,b, ...)
- statistic identification of patterns of language change *(Blasi et al. 2019)*
- ...

from word lists to trees

1. perform cognate classification (manual or automatic)
2. construct binary *character matrix*
3. let computer search the tree(s) that best explain(s) the distribution of 0s and 1s in the character matrix

manual cognate annotation

manual cognate annotation

- labor intensive
- available date are geographically skewed
- requires tons of prior classical historical linguistics work

automatic cognate detection

- lot of computational research over the past years to automate the process
- some relevant papers: Hauer & Kondrak (2011), List (2014), Rama (2015), Jäger, List & Sofroniev (2017)
- results are usable but far from perfect

Out[5]: 25×5 DataFrame

Row	language	concept	ASJP	cc	ccInferred
	String31	String7	String7	Int64	Int64
1	ANCIENT_GREEK	water	hidor	988	162
2	BELARUSIAN	water	vada	988	162
3	BULGARIAN	water	vuda	988	162
4	CZECH	water	voda	988	162
5	DANISH	water	van	988	162
6	DANISH_FJOLDE	water	van	988	162
7	DUTCH	water	wat3r	988	162
8	ELFDALIAN	water	wotn	988	162
9	ENGLISH	water	wot3	988	162
10	FAROESE	water	vatn	988	162
11	FRISIAN	water	vEt3r	988	162
12	GERMAN	water	vasa	988	162
13	GUTNISH_LAU	water	vatn	988	162
14	ICELANDIC	water	vatn	988	162
15	IRISH	water	iSk3	988	165
16	MACEDONIAN	water	voda	988	162
17	NORWEGIAN_RIKSMAL	water	wan	988	162
18	OLD_CHURCH_SLAVONIC	water	vodo	988	162
19	OLD_IRISH	water	uske	988	165
20	OLD_NORSE	water	vatn	988	162
21	OLD_SWEDISH	water	vatn	988	162
22	OSSETIC	water	don	988	162
23	OSSETIC_DIGOR	water	don	988	162
24	OSSETIC_IRON	water	don	988	162
25	POLISH	water	voda	988	162

Out[6]: 25×5 DataFrame

Row	language	concept	ASJP	cc	ccInferred
	String31	String7	String7	Int64	Int64
1	RUSSIAN	water	voda	988	162
2	SERBO-CROATIAN	water	voda	988	162
3	SLOVAK	water	voda	988	162
4	SLOVENIAN	water	voda	988	162
5	SORBIAN_LOWER	water	w3da	988	162
6	SORBIAN_UPPER	water	woda	988	162
7	STAVANGERSK	water	watn	988	162
8	SWEDISH	water	vat3n	988	162
9	UKRAINIAN	water	woda	988	162
10	ARMENIAN_EASTERN	water	jur	989	162
11	CLASSICAL_ARMENIAN	water	jowr	989	162
12	BIHARI	water	pain	990	162
13	MARATHI	water	oani	990	162
14	ORIYA	water	pani	990	162
15	URDU	water	poni	990	162
16	CATALAN	water	ayxw3	991	162
17	FRENCH	water	o	991	163
18	ITALIAN	water	akkwa	991	162
19	LATIN	water	aka	991	162
20	PORTUGUESE	water	agwa	991	162
21	SPANISH	water	axwa	991	162
22	GREEK	water	nero	992	164
23	MAGAHI	water	poni	993	162
24	MIDDLE_CORNISH	water	dour	994	162
25	OLD_IRISH	water	dobur	994	162

phylogenetic signal below cognacy

- sound change and morpological change contains relevant phylogenetic information

Out[7]: 21×4 DataFrame

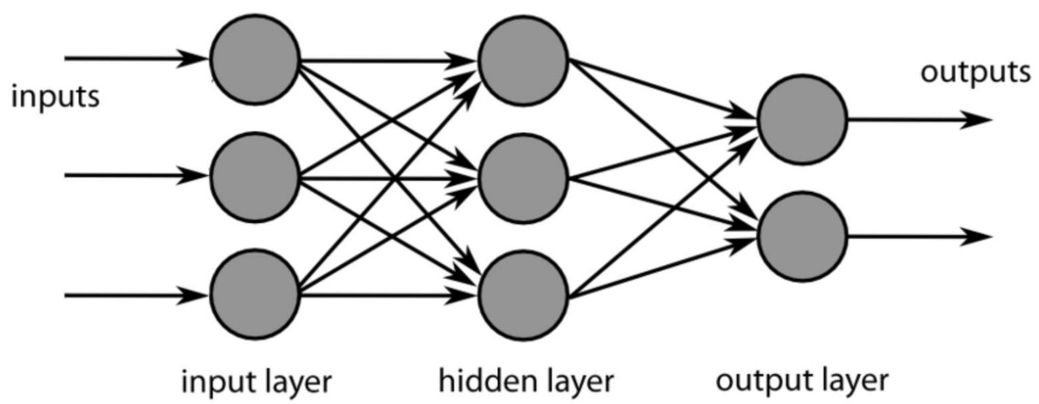
Row	Language_ID	Parameter_ID	Segments	Cognateset_ID
	String15?	String15?	String?	Int64?
1	urdu	180_tooth	d̪ ɑ̃ ʈ̪	328
2	catalan	180_tooth	d e n	328
3	armenianmod	180_tooth	ɑ t ɑ m	328
4	bretonst	180_tooth	d ã n t	328
5	german	180_tooth	ts aː n	328
6	italian	180_tooth	d ɛ n t e	328
7	swedish	180_tooth	t a n d	328
8	greekmod	180_tooth	ð o̞ n d i	328
9	marathi	180_tooth	d a t	328
10	portuguesest	180_tooth	d ẽ t ɨ	328
11	spanish	180_tooth	d j e n t e	328
12	danish	180_tooth	d̪ʰ/dʰ a n	328
13	dutchlist	180_tooth	t ɑ n t	328
14	english	180_tooth	t uː θ	328
15	french	180_tooth	d ɑ̃	328
16	russian	180_tooth	d e s n a	328
17	bihari	180_tooth	d ã t	328
18	oriya	180_tooth	d a n t ɔ	328
19	czech	180_tooth	z ʊ p	502
20	polish	180_tooth	z ɔ̃ p	502
21	russian	180_tooth	z u b	502

Deep learning

- modern machine learning should be capable to detect relevant signal all by itself, without imposing theoretical categories such as *lexical change* and *sound change*

Deep learning

- variant of *artificial neural networks* with several layers of neurons



(image from Wikipedia)

- each neuron performs an *affine transformation* of the input, followed by a non-linear function

$$\mathbf{x} \mapsto f(\langle \mathbf{a}, \mathbf{x} \rangle + b)$$

Deep learning

- training a model means finding the best values for $\mathbf{a}$, b for each neuron $\rightarrow$ easily millions of parameters to train
- since ten years ago, progress in hardware and theoretical understanding made it possible to effectively train such models on massive amounts of data
- performance for NLP, image processing etc. etc. easily beats all other machine learning approaches
- conceptual jump for us linguists: **Deep learning requires continuous data, a.k.a. vectors of real numbers, rather than discrete categories.**

discrete ⟶ continuous

- discrete date over a finite, known set can easily converted into vectors via *one-hot encoding*

dense embedding of sound classes

X	G	7	q	s	t	k	C	T	a	E	3
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	1

F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
0.040	0.371	−0.912	−0.267	0.537	−0.063	0.991	0.430	−0.819	−0.279
−0.039	0.572	−0.466	−0.919	−0.240	−0.370	0.431	0.905	−0.796	−0.423
−0.329	0.485	−1.067	0.450	−0.619	−0.002	0.121	−0.477	−0.826	0.301
−0.358	0.270	−0.552	−0.233	−0.066	−0.048	0.807	0.651	−0.794	−0.481
−0.942	−0.316	−0.752	0.044	0.634	−0.491	−0.063	−0.441	−0.691	−0.816
−0.390	−0.279	0.622	−0.550	1.391	0.128	0.156	−0.519	0.990	−0.323
−0.617	0.552	−0.202	−0.698	−0.473	−0.318	−0.221	0.640	−0.892	−0.716
−0.962	−0.384	−0.728	−0.601	1.232	0.126	0.072	0.053	0.108	−0.046
−0.811	−0.556	−0.840	−0.295	1.015	0.327	−0.323	−0.031	0.264	0.187
−0.070	0.360	−0.133	−0.156	−0.345	−0.476	0.297	−0.043	0.872	0.207
−0.360	−0.367	−0.217	−0.041	−0.359	−0.155	−0.370	−0.071	0.836	0.694
−0.042	−0.018	−0.112	0.336	−0.222	−0.203	0.214	−0.341	0.470	0.766

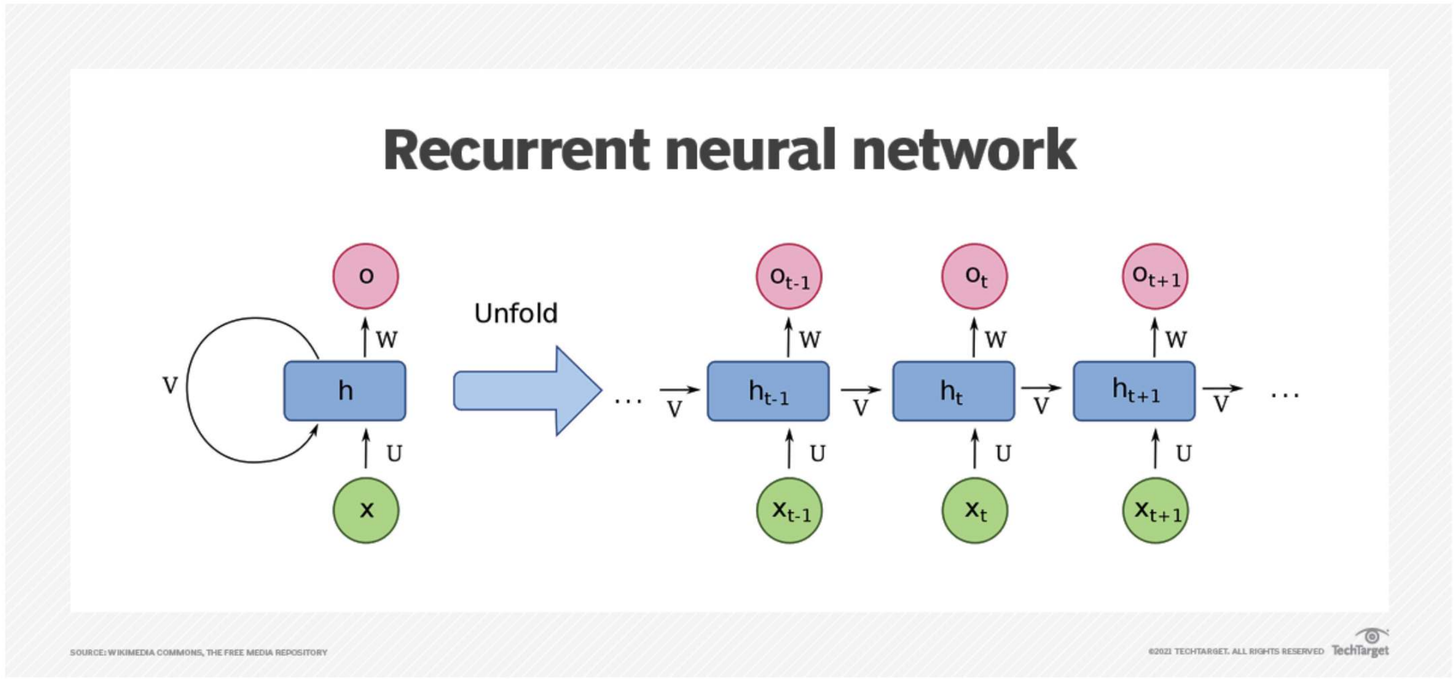
- feeding such a one-hot vector into a single layer results in a *dense embedding*, i.e. a vector with lots of real numbers

discrete ⟶ continuous

- one-hot approach can be upscaled to sequences of discrete symbols *of fixed size*
- words, sentences etc. are of variable size though

sequence-to-sequence models: decoder

- variable-sized input can be mapped to a fixed-size vector via **recurrent** neural networks

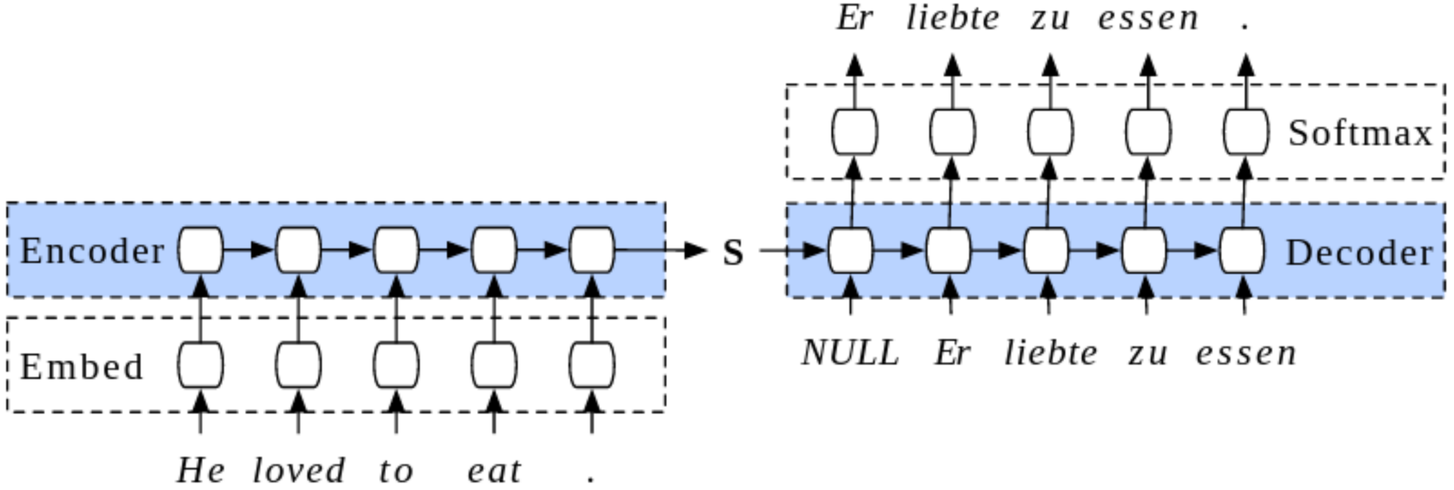


- input sequence of length n produces output sequence $\vec{o}$ of size n **plus fixed-size hidden state v**

hidden state v can be used as representation of the entire input sequence

sequence-to-sequence models: decoder

- can also be done with recurrent network
- initial state is taken from decoder
- initial input is some fixed start-of-sequence symbol
- output is probability distribution over vocabulary → final output drawn from this distribution
- output in step n is used as input in step $n + 1$
- generation stops when end-of-string symbol is produced
- often used in autmatic machine translation (outdated → nowadays everybody does attention)



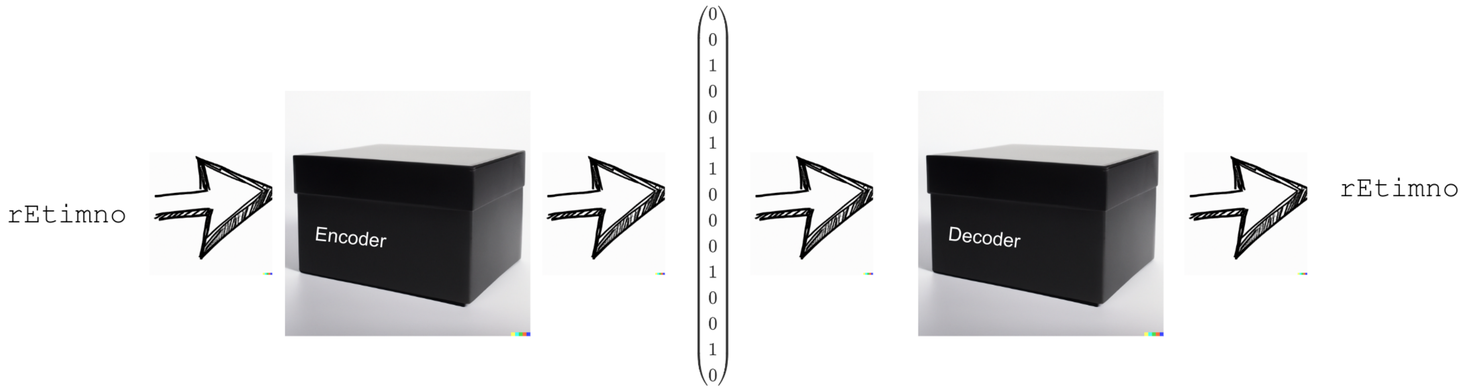
(image from https://github.com/shangeth/Seq2Seq-Machine-Translation)

Building a string autoencoder

- seq2seq models require tons of training data
- my goal is relatively modest though - dense embeddings of strings of length up to ca. 20 over a few dozen symbols
- **autoencoder architecture**:
  - input and output string are identical for training
  - final hidden state of the trained encoder is used as fixed-size vector representation of the input string
  - trained decoder represents function that maps vector representation back to a string

continuous ⟶ discrete

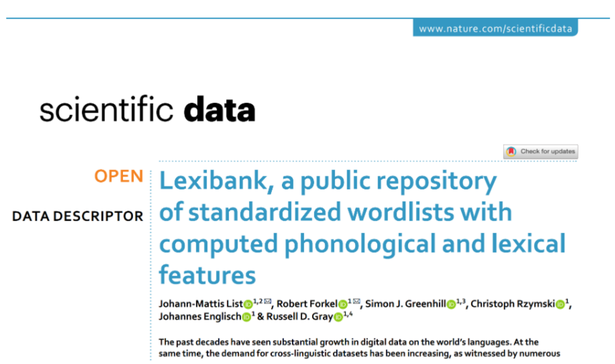
- for phylogenetic inference, we know how to do it with discrete data (and we love it)
- trick to convert between continuous and binary vectors: **straight-through estimator** (cudos to Chundra Cathcart, p.c., for pointing this out to me)



- **crucial**: Similar strings are mapped to similar representations!

Details

- implemented in `pytorch`
- encoder: sound embedding layer; 1-layer GRU, step-function output layer (all layers with 100 neurons)
- decoder: sound embedding layer; 1-layer GRU, dense hidden layer (all with 100 neurons), softmax output
- Data from *Lexibank*; selection `lexibank-analysed` (https://github.com/lexibank/lexibank-analysed) set up by Johann-Mattis List



- 340,515 words in total, from 1,298 doculects
- all string converted to ASJP sound classes
- 70% used for training and 10% for validation
- cognate class annotation only used for comparison of phylogenetic inference

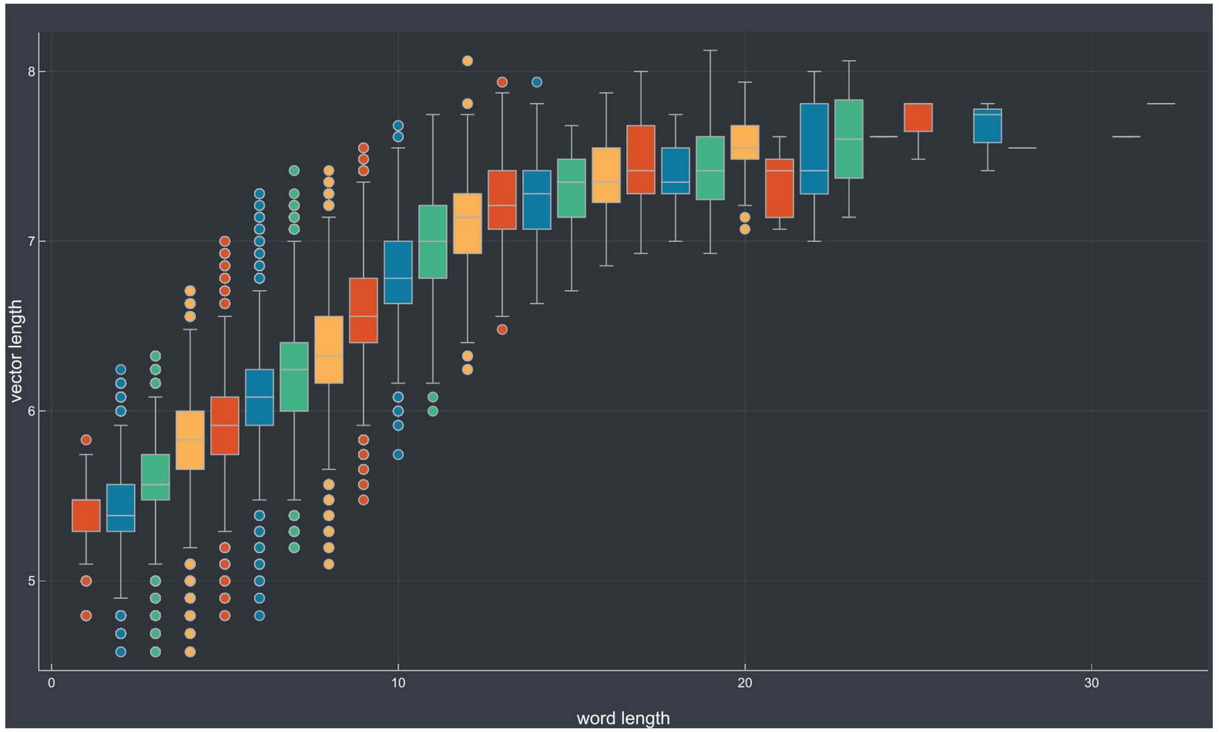
Out[8]:

25×7 DataFrame

Row	db	language	concept	cc	Glottocode	IPA	word
	String31	String31	String	String	String15	String	String?
1	abvdoceanic	AkeiPenantsiro	131_cloud	cloud-90	akei1237	k o k o	koko
2	abvdoceanic	AnejomAneityum	81_sharp	sharp-82	anei1239	a r i ɲ	ari5
3	abvdoceanic	Canala	12_skin	skin-1	xara1244	k ɐ̃	ko
4	abvdoceanic	Loyop	28_tobreathe	tobreathe-14	leha1244	mʷ o ŋ s e	mwoNse
5	abvdoceanic	Ponapean	148_white	white-11	pohn1238	pʷ e t e pʷ e t	pwetepwet
6	bdpa	Dinevohask	244_them	285	bulg1262	i mʲ ə	imy3
7	bdpa	OostendeBeWv	446_dutchregen	345	dutc1256	r eː h ə	reh3
8	bdpa	Zhaltushaard	170_bread	211	bulg1262	ˈh lʲ ɑ p	hlyop
9	blustaustronesian	maloh	151_green	85	emba1238	m a m a t a	mamata
10	bowernpny	Djabugay	220_oldersister	15747	dyaa1242	b a d i n i	badini
11	bowernpny	Margany	4_bloodwood	23546	marg1253	d̪ i ɹ i ɲ	diri5
12	cals	Uhitoy2	short	short-b	uzbe1247	k ɛ l t e	kElte
13	chaconcolumbian	EmberaChami	8_armadillo	1930	embe1262	ĩ tʃ u r	itSur
14	chaconcolumbian	Muinane	29_shamanfolkhealer	6997	muin1242	t aa b u m ɨ n aa p i	taabum3naapi
15	dravlex	Malto	66_red	red-A	saur1249	eː s o	eso
16	gaotb	BijangBai	42_stonerock	42_stonerock-36	nort2724	t ũ ⁴²	tu
17	leekoreanic	SJeolla	191_topoundbeat	191_topoundbeat-1	chol1278	t̪ ɛ + ɾ i n	tErin
18	peirosaustroasiatic	Nge	47_lie	1419	ngeq1245	ʔ b e c/ts	7bets
19	robinsonap	kaera	125_fruit	125_fruit-0	kaer1234	t e i + i s i	teiisi
20	robinsonap	kamang	260_sickpainful	260_sickpainful-7	kama1365	m a t	mat
21	sagartst	Bokar	181_shy	2476-shy	boka1249	ɦ e n + nˌ i ŋ	heniN
22	savelyevturkic	OldUyghur	169_sandn	158	oldu1238	q u m	qum
23	utoaztecan	Mayo	67_hear	67_hear-3	mayo1264	h í/i kː a h a	hikaha
24	yanglalo	Mangdi	528_feedanimals	518	sout3210	ts ɑ ²¹	tso
25	yanglalo	Yijiacun	217_moss	921	ekaa1234	n a ²⁴ + m ĩ ²¹	nami

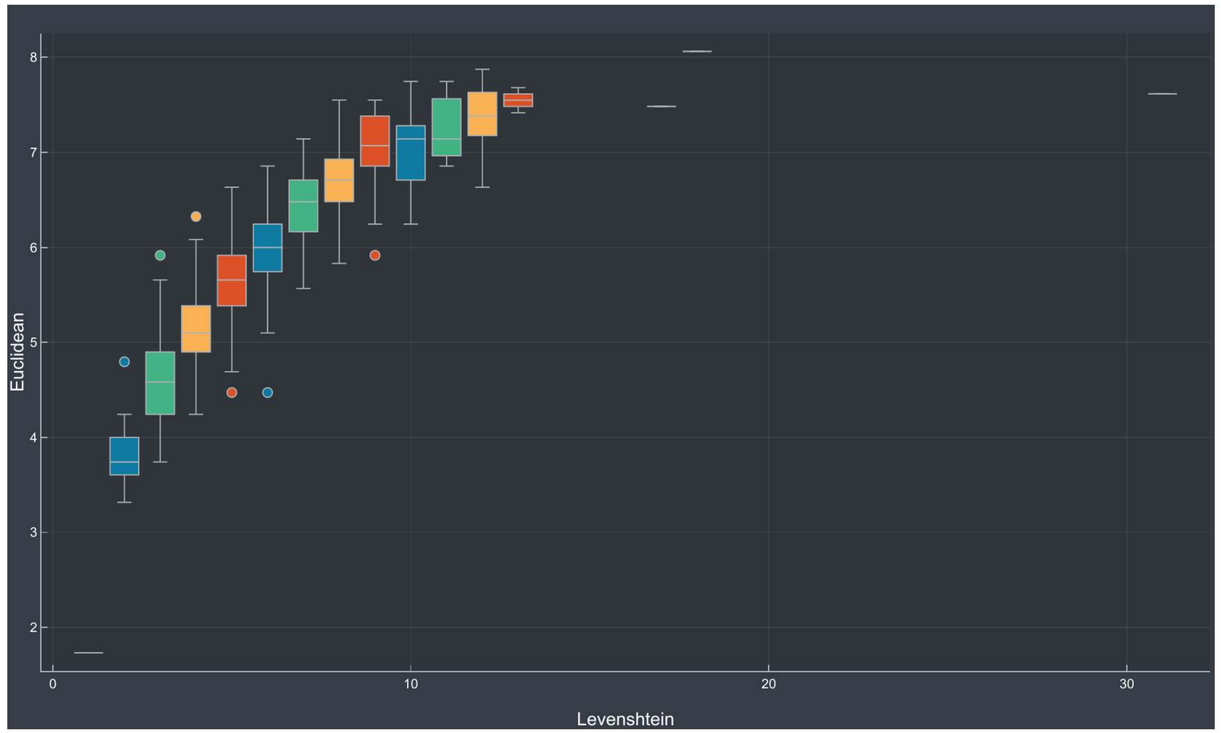
relationship between word lengths and corresponding vector lengths

Out[10]:



String distance vs. vector distance

Out[12]:



generating random words by sampling random vectors and converting them to strings

Out[13]: 20×1 DataFrame

Row	word
	String
1	Nwy3a
2	varyo
3	Nonr
4	suri
5	kkev
6	vlgpa
7	raoN
8	kuneko
9	zn3d3
10	7arta
11	n7axi
12	romeye
13	buto5
14	kbot3
15	sodZi
16	gur3
17	ylvei
18	hoati
19	haav3
20	khuom

exploring the environment of a word

Out[14]: 20×2 DataFrame

Row	neighbor	distance
	String?	Float64
1	hai5ode	0.0
2	hololе	4.12311
3	haihene	4.24264
4	hirode	4.24264
5	gaa5i7u	4.24264
6	hetso7	4.24264
7	hekolE	4.24264
8	kakole	4.3589
9	haiahoia	4.3589
10	saodZo	4.3589
11	a5oo7u	4.3589
12	haime7e	4.3589
13	hatSize	4.3589
14	hatholo	4.3589
15	ailolo	4.47214
16	kauhoe	4.47214
17	kahore	4.47214
18	aibon	4.47214
19	haiona	4.47214
20	haiono	4.47214

random walk from one word to another

```
katL
katL
katL
kats
katL
katL
katL
katL
katLi
satLi
satLi
satLi
satLi
7at5i
nat5i
nat5i
not5i
sat5i
satSi
sat5i
satSi
natSi
notSi
notShi
notShi
natShi
notShi
notShi
notShi
notShi
notShi
notShi
notShi
```

Pilot study: cognate clustering

- data from Jäger et al. (2017)
- 13 datasets of Swadesh lists with expert cognate annotation
- ca. 10,000 words in total
- ca. 180,000 word pairs wich could be cognate (same family, same meaning)

Jäger, List & Sofroniev (2017)

- each datapoint is a pair of words from same family with same meaning
- dependent variable: cognate or not
- independent variables:
    - string similarity
    - language similarity
    - average word length of words for the concept in question (proxy for diachronic stability)
    - correlation between word distance and language distance for words for that concept (proxy for borrrowability)
- supervised training of a **Support Vector Machine**

current approach

- data as described above
- independent variables:
    - all variables mentioned above
    - word embeddings for both words
- supervised training of a five-layer neural binary classifier
    - input layer: 207 neurons
    - hidden layers of sizes 250, 128, 64
    - output layer of size 1 (probability of cognacy)

comparison

B-cubed F-score, cross-validation

Out[16]: 13×4 DataFrame

Row	dataset	neural classifier	SVM	delta
	String15	Float64	Float64	Float64
1	abvd2-part1	0.806	0.718	0.087
2	Chinese-180-18	0.66	0.61	0.05
3	Japanese-200-10	0.951	0.896	0.055
4	ObUgrian-110-21	0.951	0.906	0.044
5	IELex-2016	0.771	0.617	0.154
6	Afrasian	0.722	0.412	0.31
7	Huon	0.746	0.509	0.238
8	Kadai	0.848	0.797	0.051
9	Kamasau	0.961	0.896	0.065
10	Mayan	0.772	0.661	0.111
11	Miao-Yao	0.837	0.868	-0.031
12	Mixe-Zoque	0.93	0.839	0.091
13	Mon-Khmer	0.793	0.605	0.187

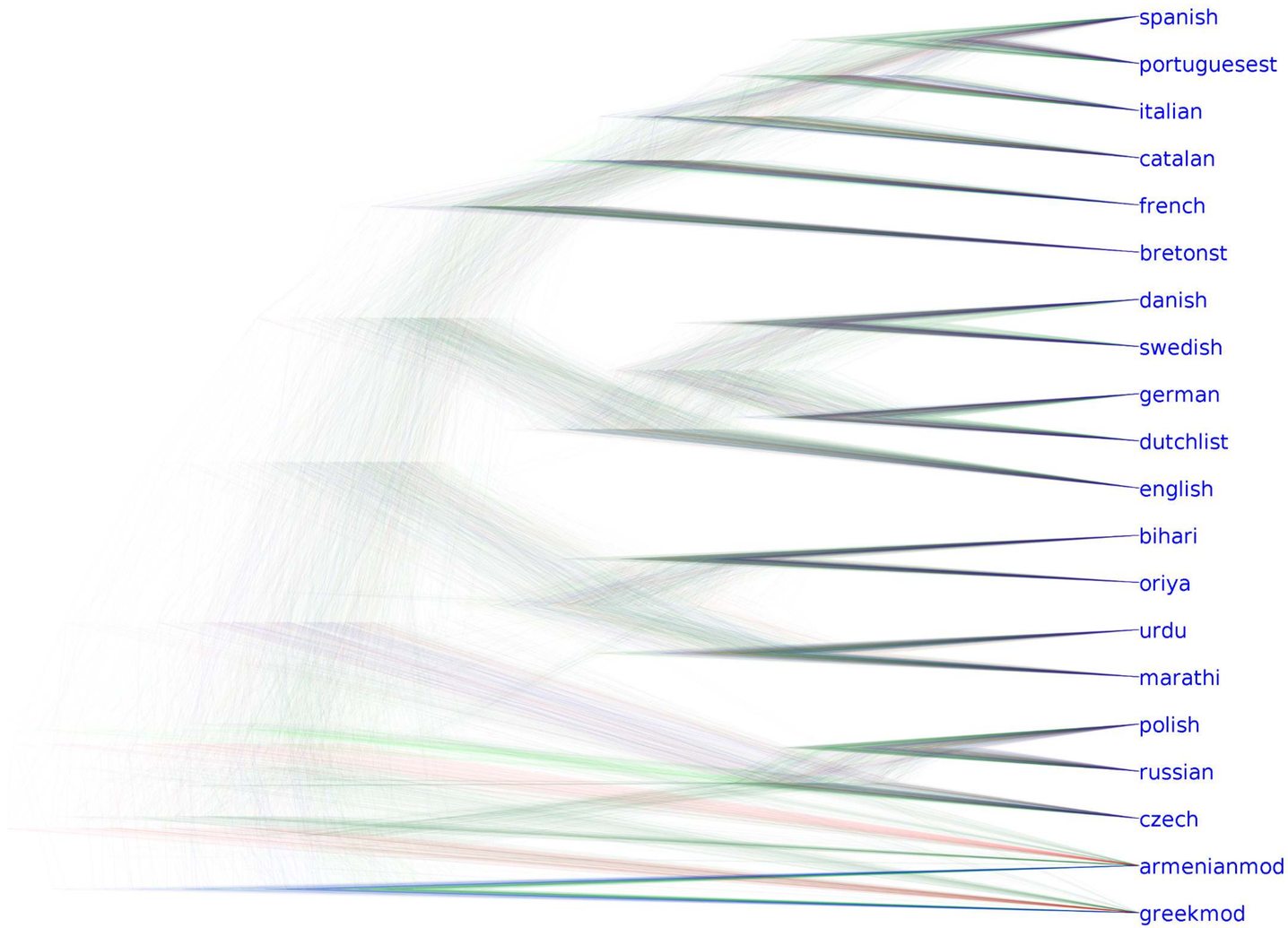
Out[17]: 1×4 DataFrame

Row	dataset	neural classifier	SVM	delta
	String	Float64	Float64	Float64
1	mean	0.827	0.718	0.109

Phylogenetic inference with autoencoded strings

- each string is coded as binary vector of length 100
- for a Swadesh list with 200 concepts, each language is represented by a binary vector of length $100 \times 200 = 20,000$
- if a database contains several translations for a concept, one is chosen at random
- if an entry is missing, the corresponding region of the vector is filled with `missing` values
- result is fed to standard Bayesian phylogenetics software (`MrBayes`)

posterior distribution for `dunnielex` data

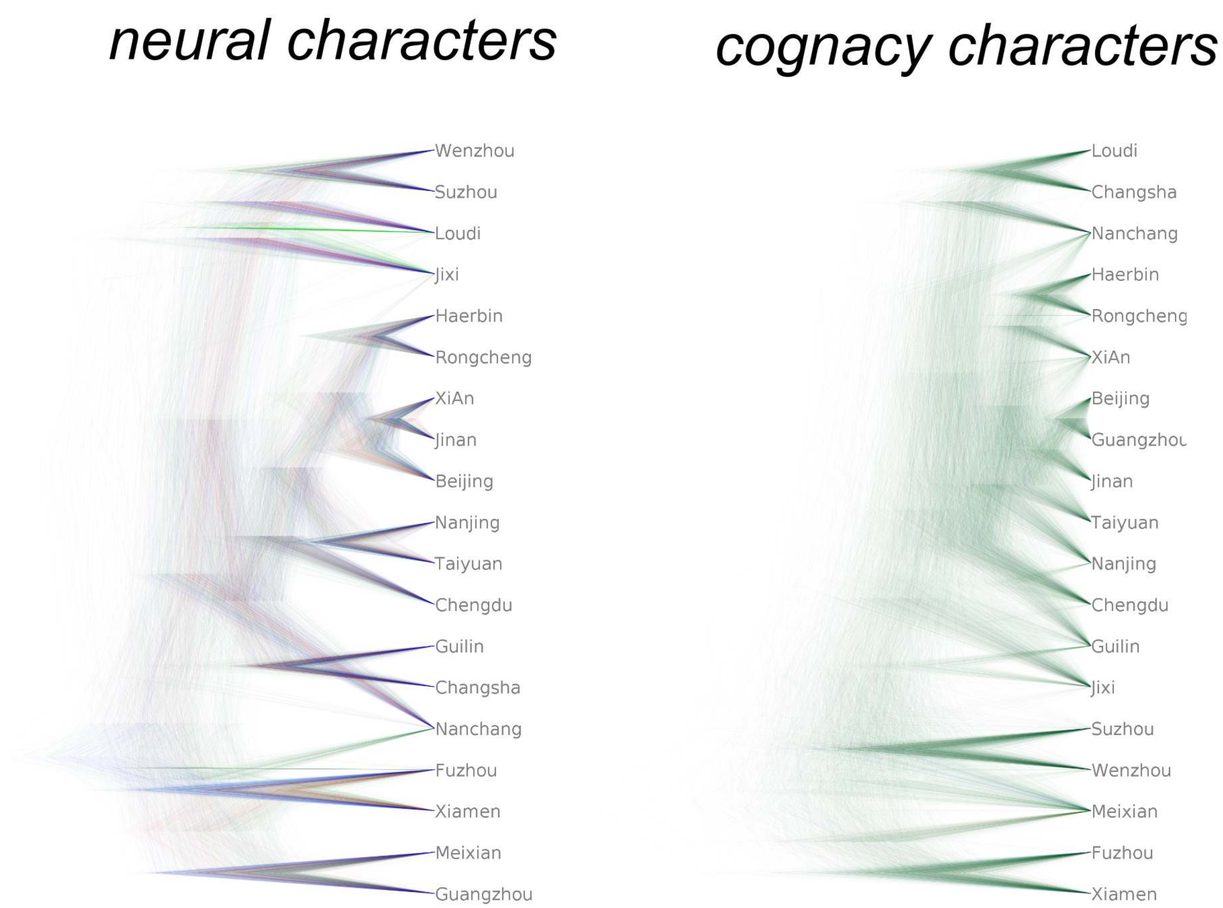


compare to tree based on manual cognate classification data

looking at a younger language family

Liú et al.'s "Collection of Basic Words in Chinese Dialects" from 2007

compare to tree based on manual cognate classification data

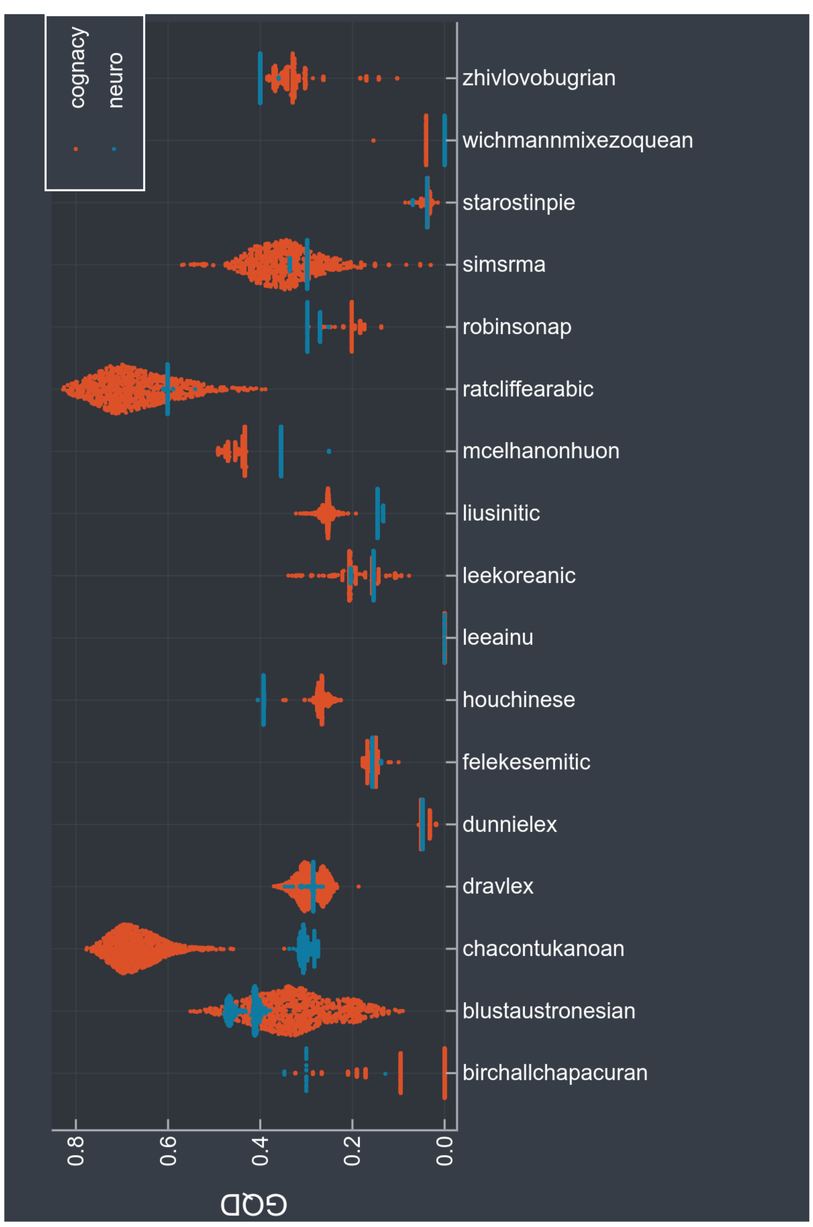


systematic comparison

- Generalized Quartet Distance: quantifies inconsistencies between automatically generated binary tree and a (often non-binary branching) goldstandard tree
- I use Glottolog classification as goldstandard
- posterior distributions of GQD for Bayesian phylogenies from neural vs. cognate class data

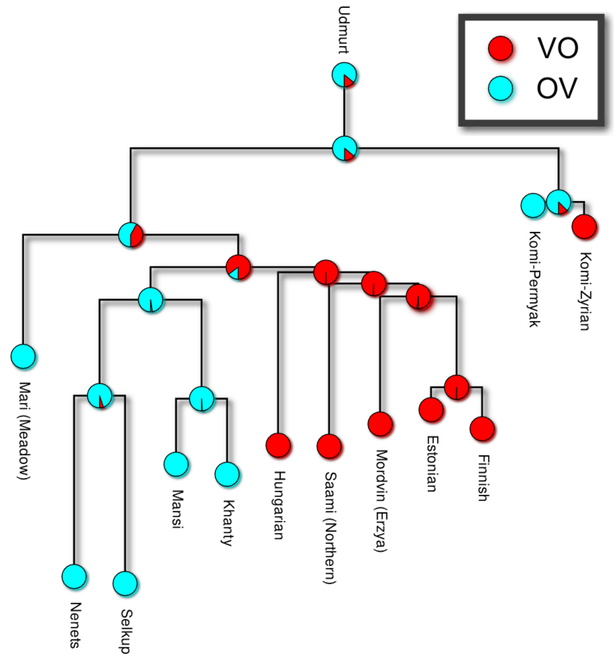
Out[19]: 18×4 DataFrame

Row	db	dl	cc	diff
	String31	Float64	Float64	Float64
1	chacontukanoan	0.3	0.67	-0.37
2	liusinitic	0.14	0.26	-0.11
3	mcelhanonhuon	0.35	0.45	-0.1
4	ratcliffearabic	0.6	0.67	-0.07
5	simsrma	0.31	0.34	-0.03
6	wichmannmixezoquean	0.0	0.03	-0.03
7	leekoreanic	0.17	0.19	-0.02
8	mean	0.25	0.25	-0.01
9	dravlex	0.29	0.29	-0.01
10	felekesemitic	0.16	0.16	-0.0
11	leeainu	0.0	0.0	0.0
12	starostinpie	0.04	0.04	0.0
13	dunnielex	0.05	0.04	0.0
14	zhivlovobugrian	0.4	0.32	0.07
15	robinsonap	0.29	0.2	0.09
16	blustaustronesian	0.43	0.32	0.11
17	houchinese	0.39	0.27	0.13
18	birchallchapacuran	0.3	0.07	0.23


cognacy
neuro
zhivlovobugrian
wichmannmixezoquean
starostinpie
simsrma
robinsonap
ratcliffearabic
mcelhanonhuon
liusinitic
leekoreanic
leeainu
houchinese
felekesemitic
dunnielex
dravlex
chacontukanoan
blustaustronesian
birchallchapacuran
0.8
0.6
0.4
0.2
0.0
GQD

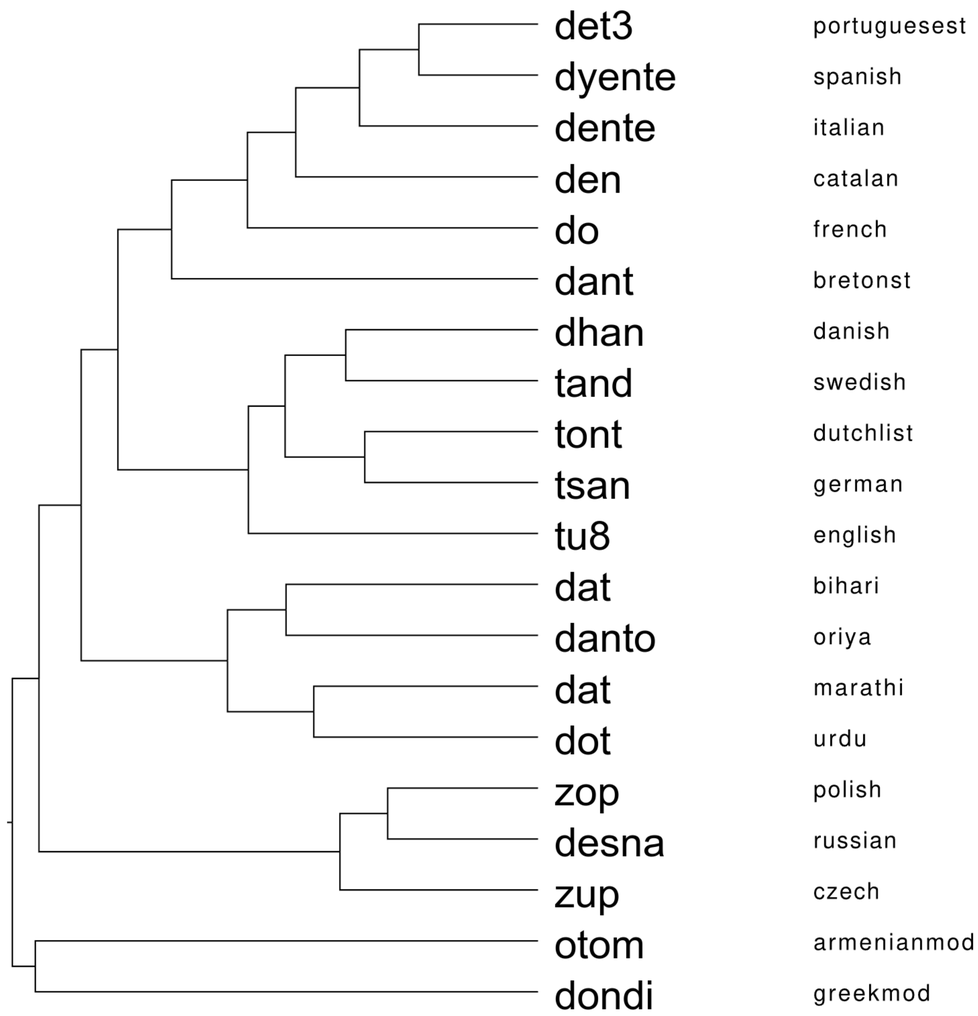
Ancestral state reconstruction

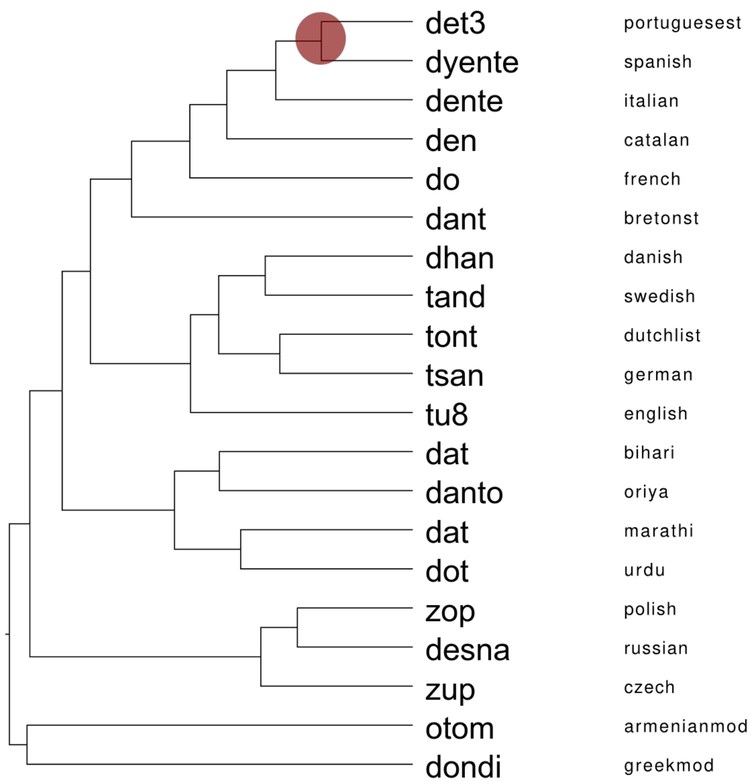
- Bayesian phyogenetic inference produces phylogenies (trees with branch lengths) and gain/loss rates
- those can be used to compute (posterior distributions of) **ancestral states**



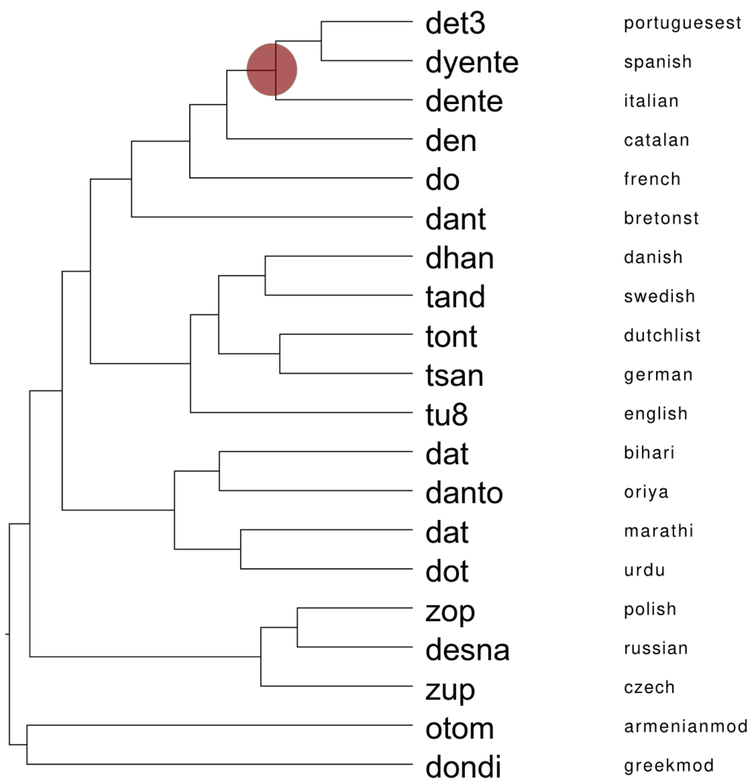
Ancestral state reconstruction

- for a given concept, we can apply this to each cell of the corresponding length-100 binary vector
- reconstructed ancestral vectors can be fed into the decoder to produce (distributions of) ancestral word forms

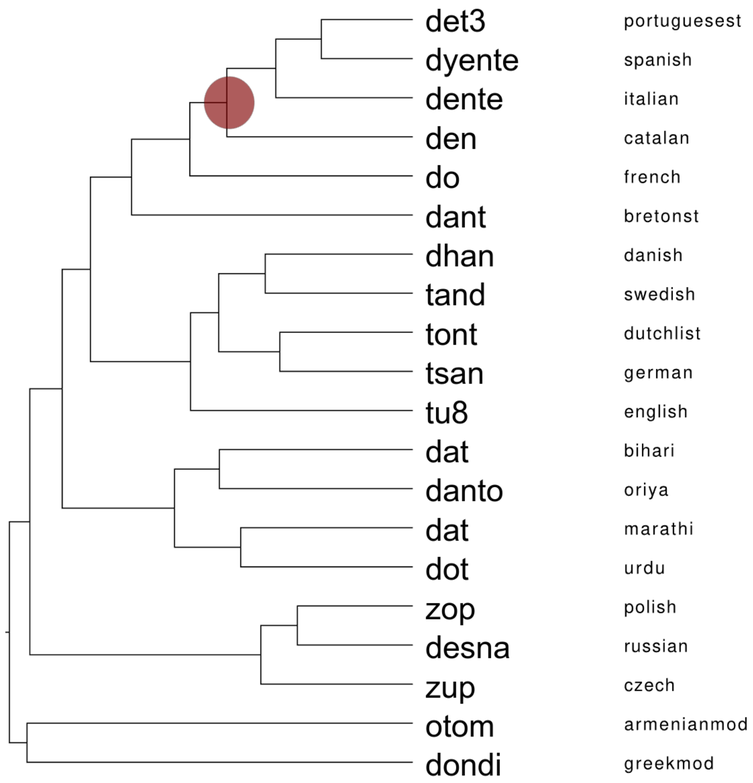

det3
dyente
dente
den
do
dant
dhan
tand
tont
tsan
tu8
dat
danto
dat
dot
zop
desna
zup
otom
dondi
portuguesest
spanish
italian
catalan
french
bretonst
danish
swedish
dutchlist
german
english
bihari
oriya
marathi
urdu
polish
russian
czech
armenianmod
greekmod



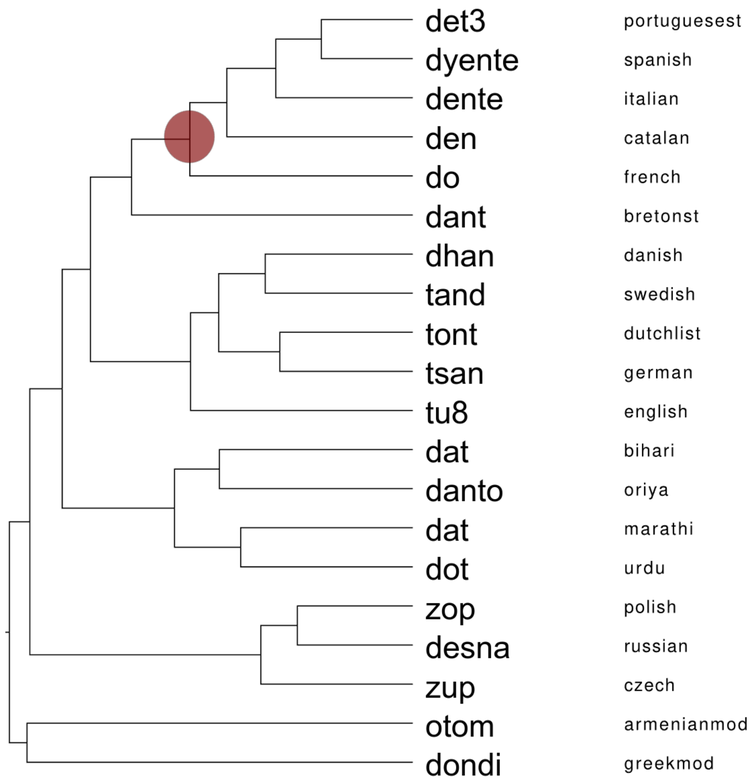
dent	78%
dente	20%
dyent	1%
dnet	1%



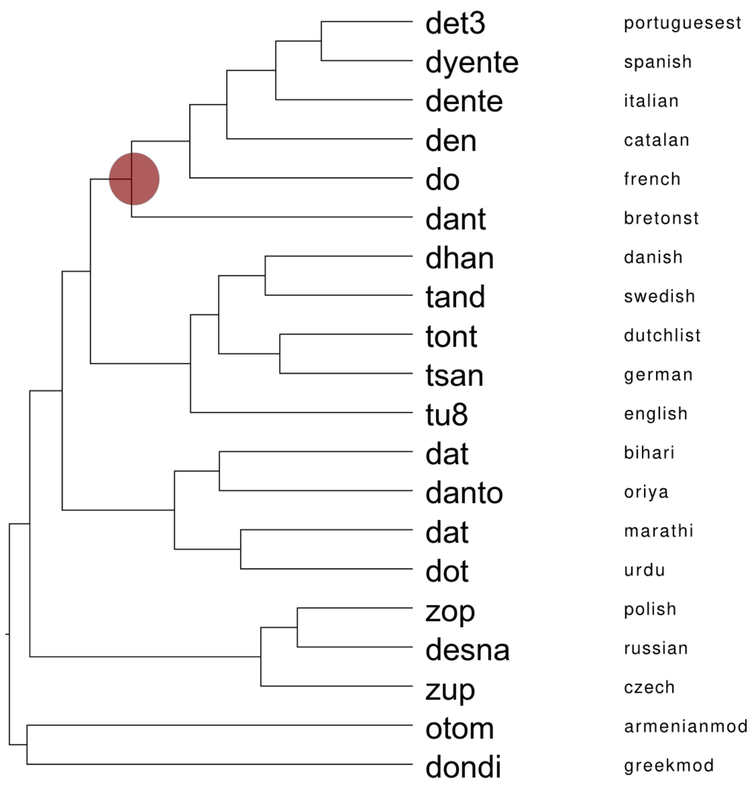
dent	82%
dente	18%



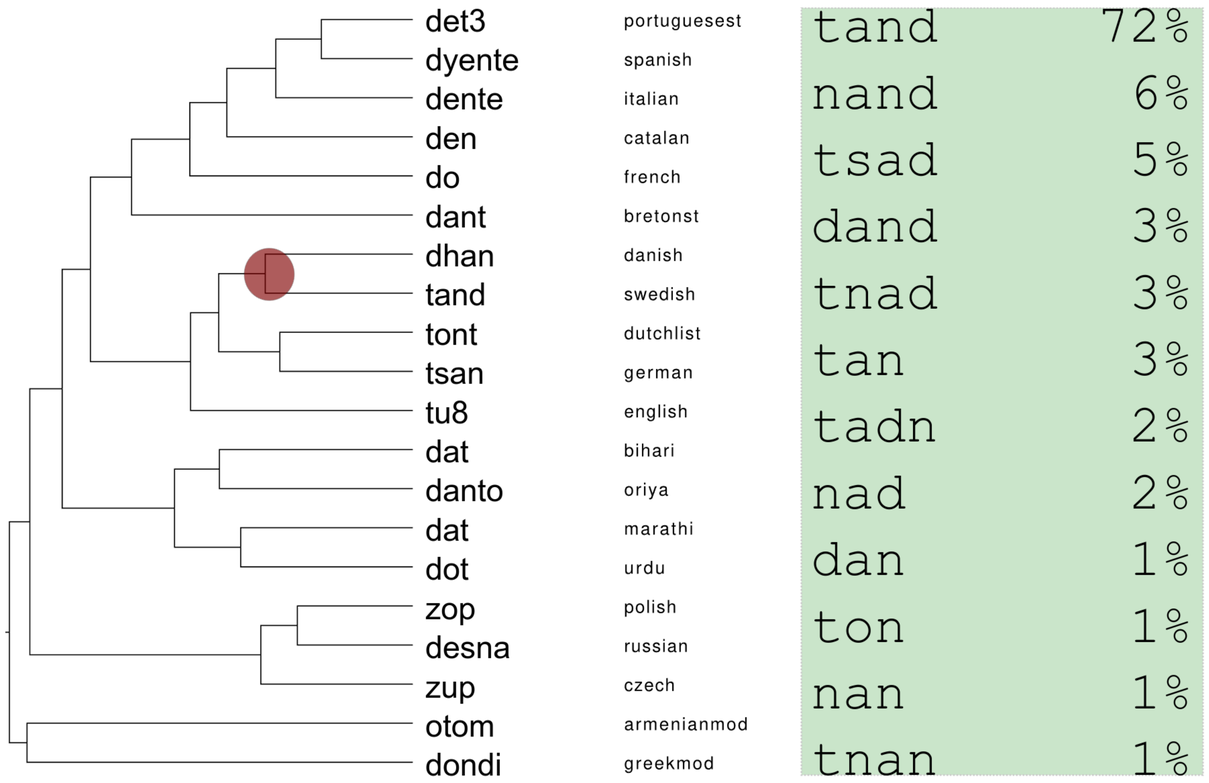
dent	67%
den	18%
dant	14%
dan	1%

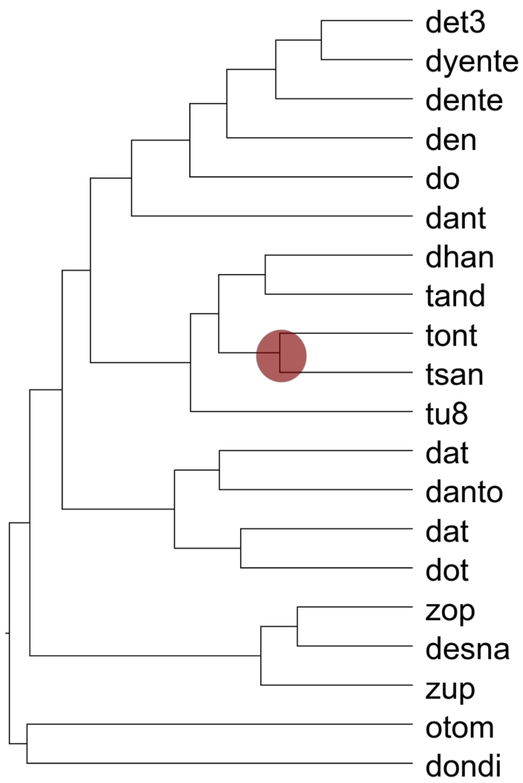


dant	71%
den	11%
dan	9%
dent	4%
dont	4%
dat	1%

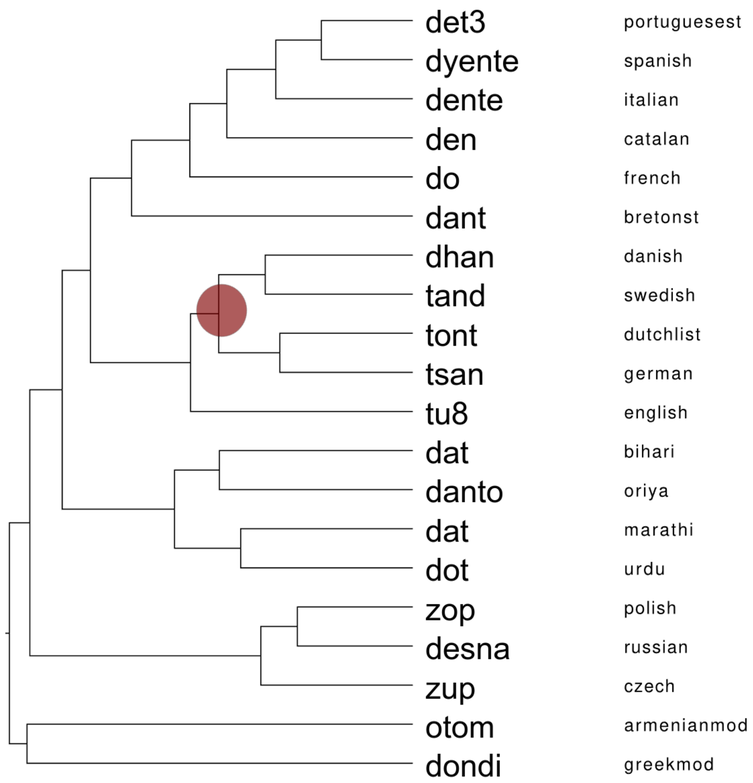


dant	89%
dan	4%
dont	3%
dat	2%
den	1%
tant	1%

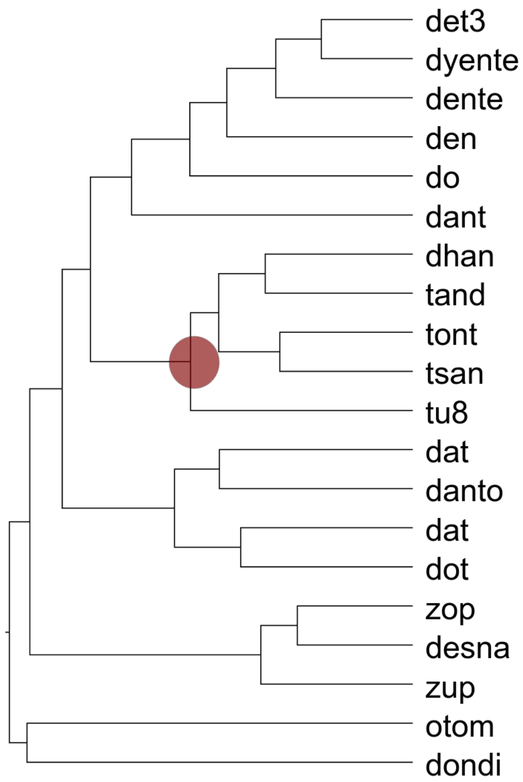

det3
dyente
dente
den
do
dant
dhan
tand
tont
tsan
tu8
dat
danto
dat
dot
zop
desna
zup
otom
dondi
portuguesest
spanish
italian
catalan
french
bretonst
danish
swedish
dutchlist
german
english
bihari
oriya
marathi
urdu
polish
russian
czech
armenianmod
greekmod
tand 72%
nand 6%
tsad 5%
dand 3%
tnad 3%
tan 3%
tadn 2%
nad 2%
dan 1%
ton 1%
nan 1%
tnan 1%



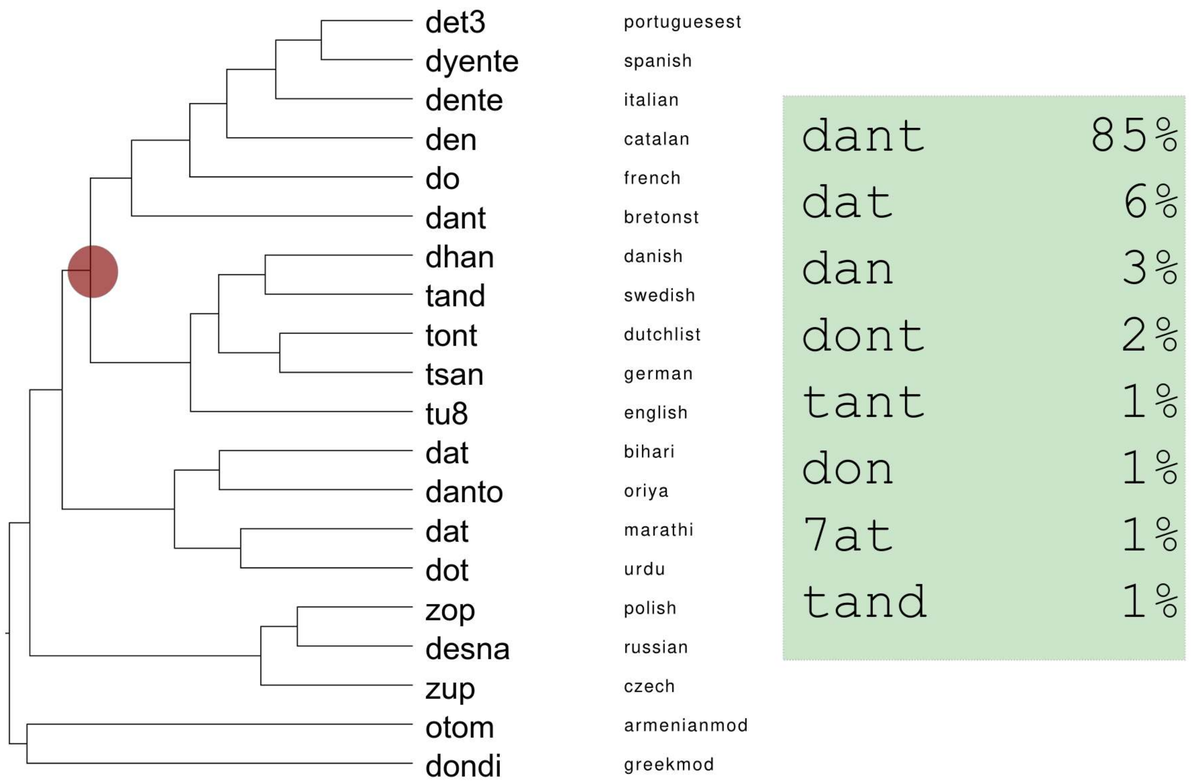
ton	50%
tont	16%
tan	8%
tsat	6%
tant	5%
tond	5%
tsan	3%
tson	2%
tando	1%
tsot	1%
tnat	1%
nont	1%
t3n	1%



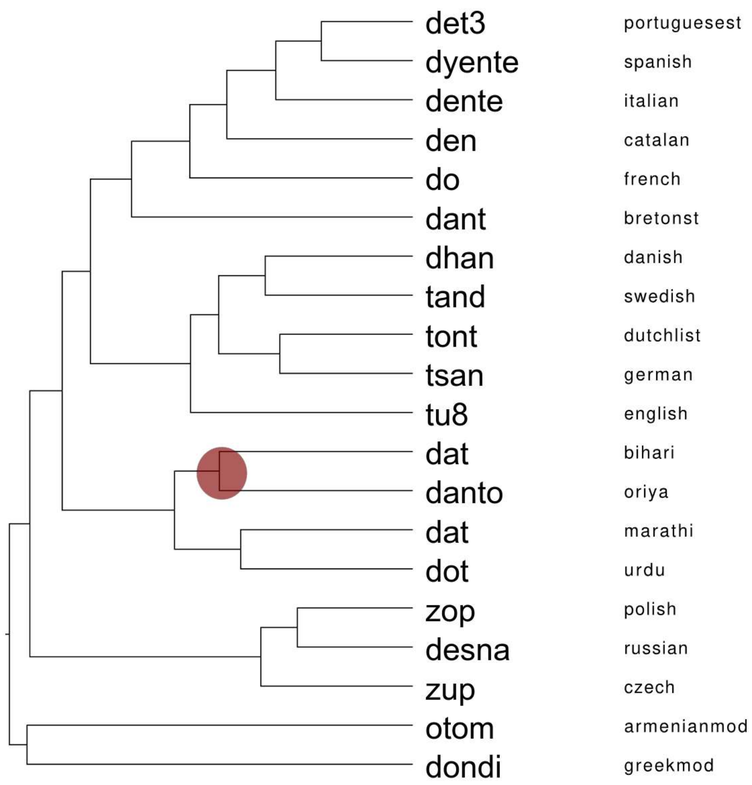
tand	53%
tan	14%
tant	10%
nant	4%
nand	3%
tsad	3%
dant	2%
ton	2%
tont	2%
tnat	2%
tsat	1%
tond	1%
tadn	1%
dand	1%
nadt	1%



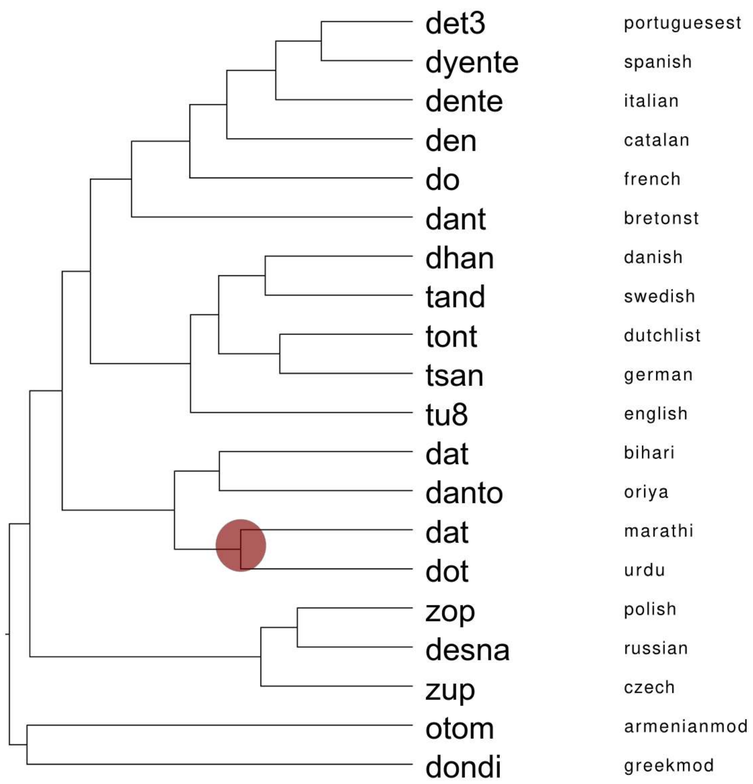
tand	47%
tan	23%
tant	12%
nant	5%
dant	3%
ton	3%
ten	1%
dan	1%
tont	1%
tadn	1%
nad	1%
tnat	1%
tadt	1%



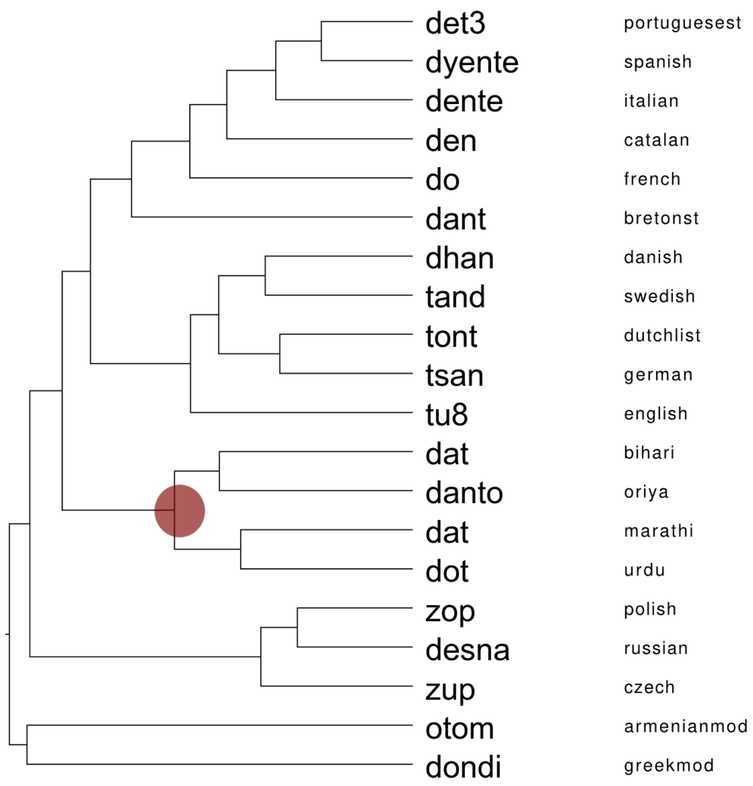
dant	85%
dat	6%
dan	3%
dont	2%
tant	1%
don	1%
7at	1%
tand	1%



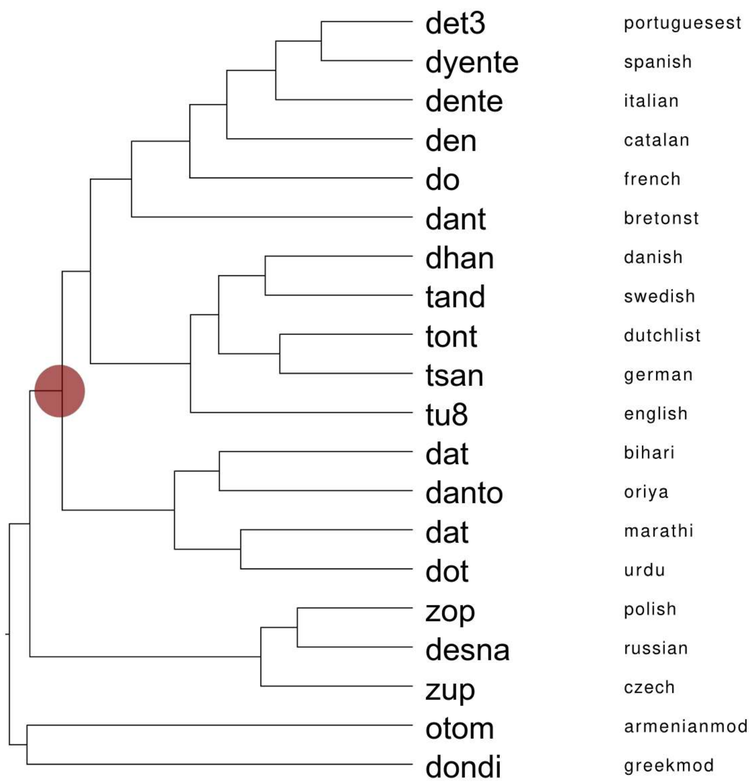
dat	88%
dant	8%
datt	2%
dats	1%
datr	1%


det3
dyente
dente
den
do
dant
dhan
tand
tont
tsan
tu8
dat
danto
dat
dot
zop
desna
zup
otom
dondi
portuguesest
spanish
italian
catalan
french
bretonst
danish
swedish
dutchlist
german
english
bihari
oriya
marathi
urdu
polish
russian
czech
armenianmod
greekmod

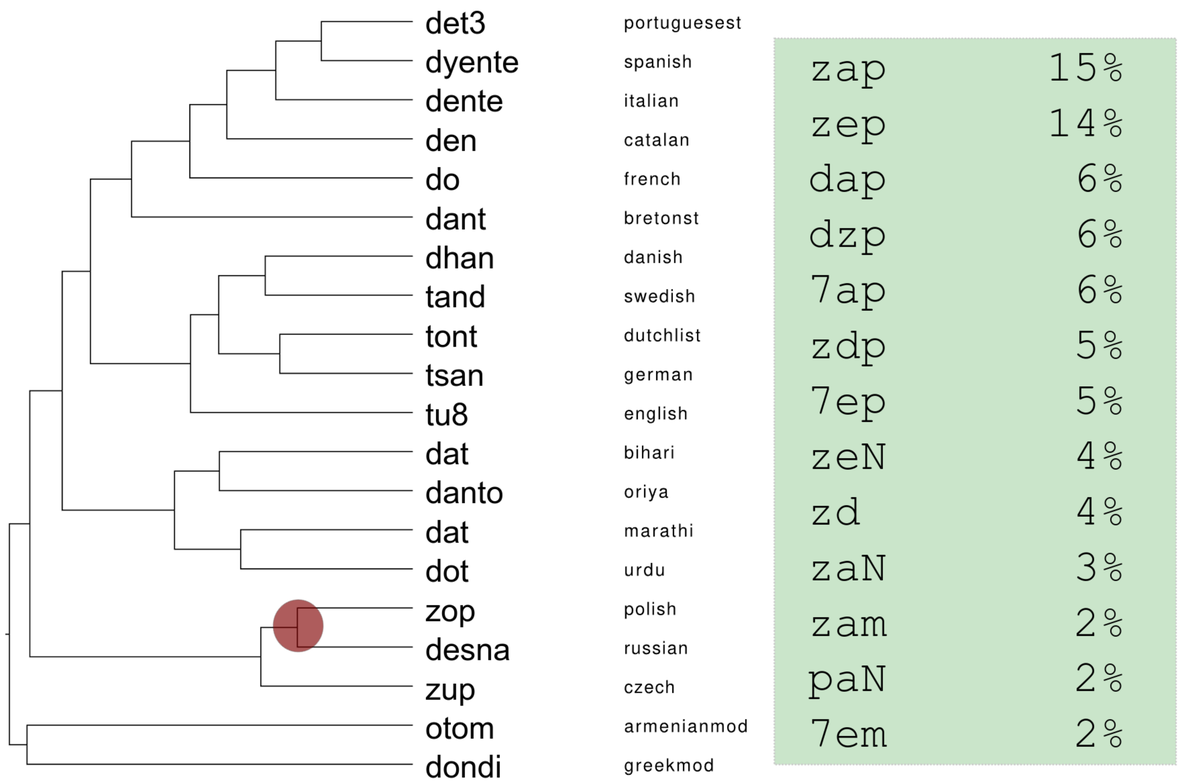
dat	94%
dot	6%

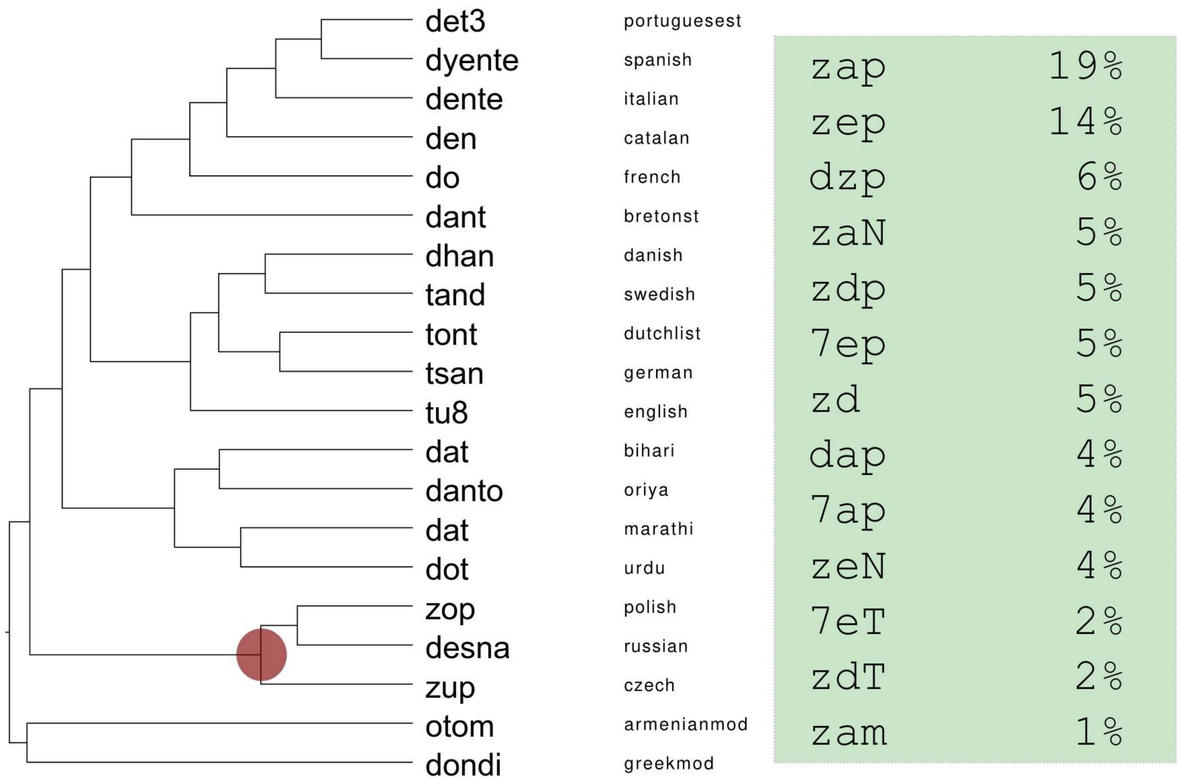


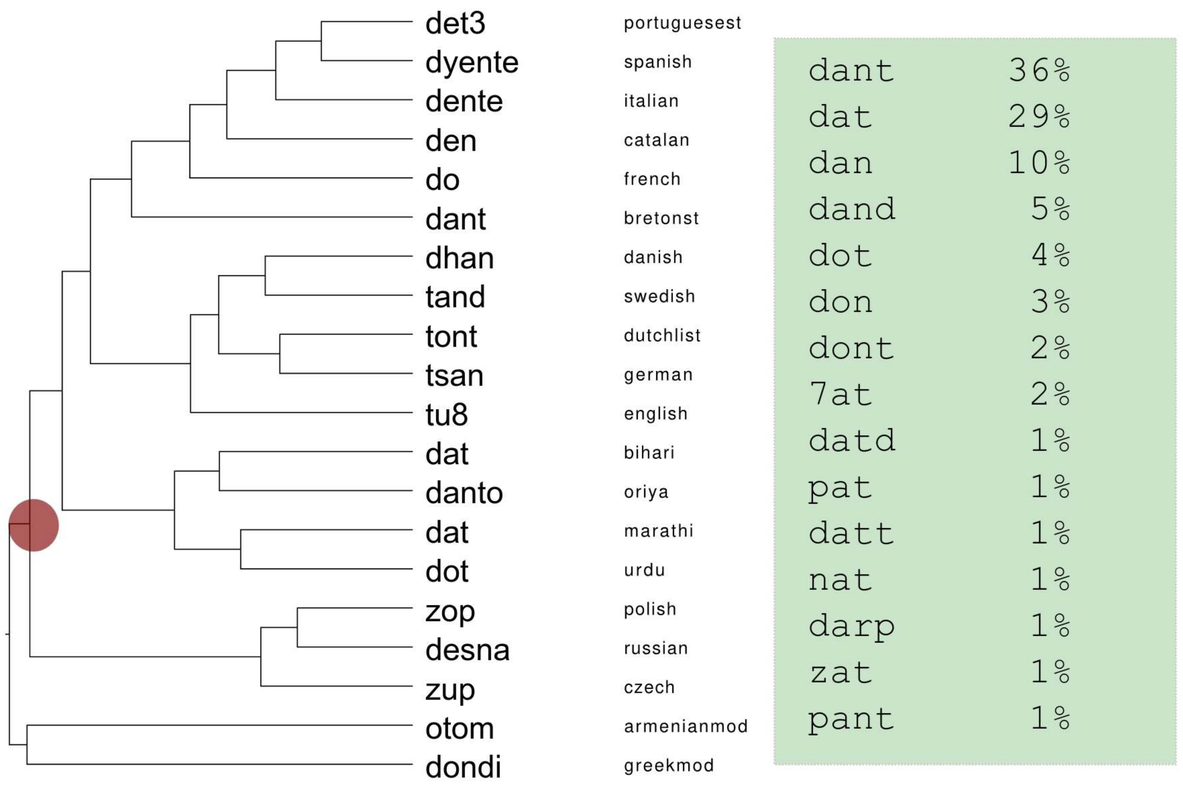
dat	89%
dant	8%
dats	1%
datt	1%
dot	1%

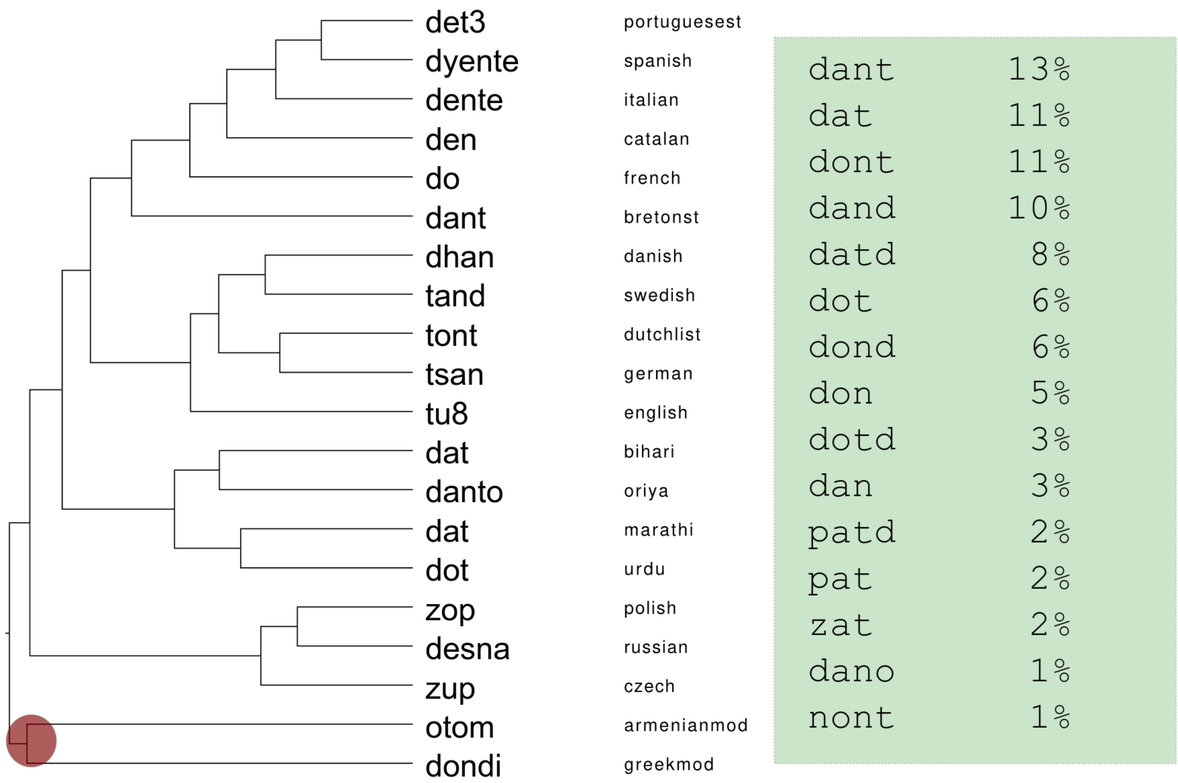


dant	58%
dat	34%
dan	3%
dot	2%
dont	2%
dar	1%

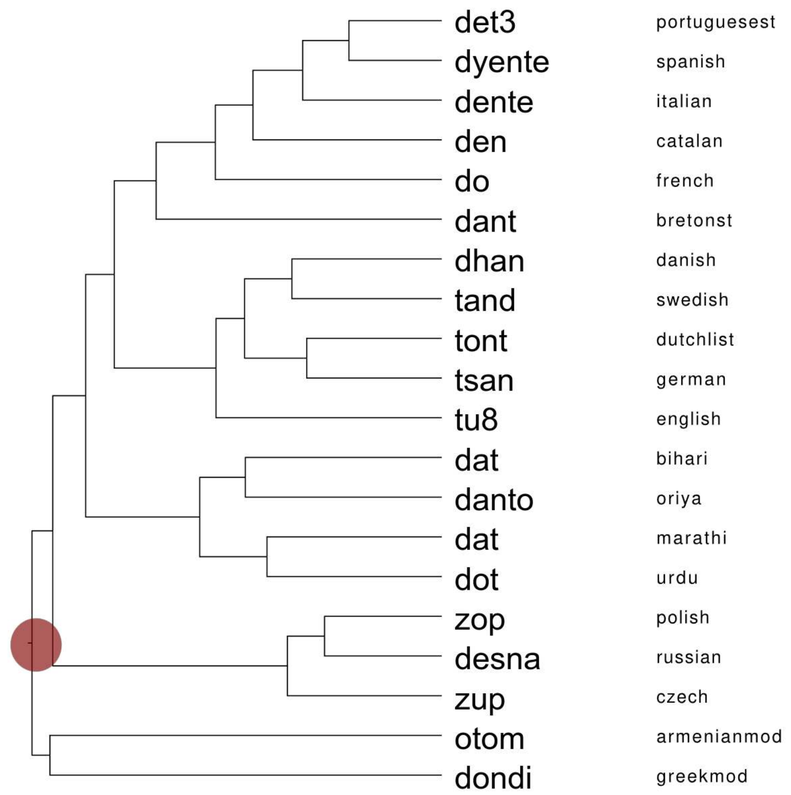

det3
dyente
dente
den
do
dant
dhan
tand
tont
tsan
tu8
dat
danto
dat
dot
zop
desna
zup
otom
dondi
portuguesest
spanish
italian
catalan
french
bretonst
danish
swedish
dutchlist
german
english
bihari
oriya
marathi
urdu
polish
russian
czech
armenianmod
greekmod
zap 15%
zep 14%
dap 6%
dzp 6%
7ap 6%
zdp 5%
7ep 5%
zeN 4%
zd 4%
zaN 3%
zam 2%
paN 2%
7em 2%


det3
dyente
dente
den
do
dant
dhan
tand
tont
tsan
tu8
dat
danto
dat
dot
zop
desna
zup
otom
dondi
portuguesest
spanish
italian
catalan
french
bretonst
danish
swedish
dutchlist
german
english
bihari
oriya
marathi
urdu
polish
russian
czech
armenianmod
greekmod
zap 19%
zep 14%
dzp 6%
zaN 5%
zdp 5%
7ep 5%
zd 5%
dap 4%
7ap 4%
zeN 4%
7eT 2%
zdT 2%
zam 1%


det3
dyente
dente
den
do
dant
dhan
tand
tont
tsan
tu8
dat
danto
dat
dot
zop
desna
zup
otom
dondi
portuguesest
spanish
italian
catalan
french
bretonst
danish
swedish
dutchlist
german
english
bihari
oriya
marathi
urdu
polish
russian
czech
armenianmod
greekmod
dant 36%
dat 29%
dan 10%
dand 5%
dot 4%
don 3%
dont 2%
7at 2%
datd 1%
pat 1%
datt 1%
nat 1%
darp 1%
zat 1%
pant 1%



dant	13%
dat	11%
dont	11%
dand	10%
datd	8%
dot	6%
dond	6%
don	5%
dotd	3%
dan	3%
patd	2%
pat	2%
zat	2%
dano	1%
nont	1%



dant	24%
dat	22%
dand	11%
dot	7%
dont	7%
don	6%
dan	4%
datd	3%
zat	2%
7at	2%
nont	1%
dots	1%
patd	1%
pat	1%
wan	1%

Conclusion

- deep learning has high potential for extracting relevant information from comparative databases
- can be combined with standard phylogenetic methods
- allows character-based phylogenetic for low-resource families
- questions for further research
    - separating phylogenetic signal from orthogonal factors (morphology, coding errors...)
    - fine-tuning DL training to focus specifically on phylogenetic information
    - shifting to continuous representations for phylogenetic inference and ancestral form reconstruction